Universiteit Leiden

Exploring Disjunctive Queries in Subgroup Discovery

Marit Paul (s4037774)

Supervisors:
Francesco Bariatti & Arno Knobbe

Bachelor Thesis– Data Science and Artificial Intelligence

Leiden Institute of Advanced Computer Science (LIACS)
liacs.leidenuniv.nl

July 1, 2026

Contents

Abstract

Subgroup Discovery (SD) identifies interpretable subpopulations that exhibit unusual target variable distributions. The standard representation uses conjunctions of conditions, which limits the expressiveness of discovered patterns. Many real-world patterns are fundamentally disjunctive, requiring unions of several profiles. This thesis investigates whether disjunctive queries improve subgroup quality and what trade-offs arise. A pairwise combination method is proposed that forms disjunctive subgroups as unions of two conjunctive components. The method is compared against a conjunctive baseline using Weighted Relative Accuracy (WRAcc), then evaluated on nine different datasets. The pairwise method consistently improves all top-10 subgroups on the Real Estate dataset. Multi-dataset evaluation shows generalization across domains. A systematic search depth analysis identifies depth 3 as the optimal setting, balancing quality gains against computational cost, which increases dramatically beyond this depth. These findings demonstrate that disjunctive queries are a valuable extension to SD, and that conjunctive and disjunctive representations serve complementary roles.

1 Introduction

What hidden patterns remain invisible simply because the tools used to find them are not expressive enough? This question lies at the heart of *exploratory data analysis*, where the goal is not only to confirm known hypotheses but to discover unexpected structures that may lead to new insights. Among the techniques developed for this purpose, Subgroup Discovery (SD) has emerged as a powerful approach for identifying interpretable subpopulations that exhibit statistically interesting behavior with respect to a target variable [7]. Unlike black-box predictive models that optimize global accuracy by fitting a single function to all instances, SD produces human-readable descriptions of specific groups (for example, "customers over 30 who live in the north") that show unusual distributions of the target variable compared to the global population. This combination of descriptive and predictive qualities has made SD valuable across diverse domains, such as biomedical research, for identifying patient subgroups with distinct gene expression profiles [4]; marketing, for detecting customer segments with unique purchasing behaviors [2]; and social science, for finding countries susceptible to political instability [11].

The standard representation for subgroup descriptions is a conjunction of conditions: conditions are connected using logical AND operators. For instance, a conjunctive subgroup description might take the form: Age $> 30 \wedge$ Region $=$ North This representation is appealing for several reasons. First, the resulting descriptions are easy to interpret because they describe a group by listing the conditions that all members must satisfy. Second, the search space remains computationally manageable because new conditions can only be added through conjunction, which strictly refines the subgroup and reduces its coverage. Third, this format is compatible with standard rule-learning algorithms, because conjunctive rules share the same logical structure as conditions in decision trees or association rules, thus existing implementations can be adapted with minimal effort. The search mechanism remains unchanged; only the evaluation function shifts from predictive accuracy to subgroup-specific measures. One such measure is the Weighted Relative Accuracy (WRAcc) measure [12], balancing coverage against the deviation of the target distribution of the subgroup from the global norm. Consequently, subgroup discovery can leverage the established search procedures from rule learning. The search for optimal conjunctive subgroups is typically performed using either beam search or exhaustive search algorithms that traverse the lattice of

possible conjunctions, pruning branches that cannot yield better subgroups according to the chosen quality measure [6].

However, purely conjunctive descriptions have an inherent limitation: they can only represent axis-aligned hyper-rectangles in the instance space, defined by intersections of simple conditions. Many real-world patterns are fundamentally disjunctive: they can be described as a union of several distinct profiles that all lead to the same outcome. Consider a marketing scenario where a product is popular among two distinct customer groups: young customers under 20 who live in the South, and older customers over 30 who live in the north. A conjunctive rule such as Age $> 30 \wedge$ Region $=$ North would capture only the older northern customers, missing the young southern group entirely. Conversely, a rule that tries to cover both groups with a single conjunction would have to include all customers between 20 and 30 regardless of region, many of whom may not be interested in the product. Similarly, in spatial data analysis, an L-shaped region of high target concentration cannot be represented as a single rectangle without including a large "missing corner" that dilutes the subgroup's quality.

These limitations suggest that disjunctive descriptions could reveal meaningful structures that purely conjunctive search systematically misses. A disjunctive description takes the form of a Disjunctive Normal Form (DNF) rule: $(A \wedge B) \vee (C \wedge D)$, which captures the union of multiple profiles. Conversely, Conjunctive Normal Form (CNF) rules such as $(A \vee B) \wedge (C \vee D)$ represent intersections of unions, capturing patterns where instances must satisfy at least one condition from each of several groups. This thesis focuses on DNF rules, as they align with the use case of identifying multiple distinct profiles that all lead to the same outcome, as illustrated in the marketing example above. CNF rules, while also representing a form of disjunction, correspond to a different logical structure and are not investigated in this work.

The customer example above illustrates a fundamental trade-off. Conjunctive rules are computationally tractable and easy to interpret, but they cannot capture patterns that consist of multiple disconnected profiles. Disjunctive rules, on the other hand, can represent such patterns exactly, but this expressiveness comes at a cost. This expressiveness-complexity trade-off captures the fundamental difficulty of disjunctive subgroup discovery. The inclusion of OR operators fundamentally expands the hypothesis space from axis-aligned rectangles to unions of rectangles. This expanded space has the potential to capture patterns that are invisible to conjunctive search. However, this increased expressiveness carries significant computational costs. The size of the hypothesis space grows combinatorially with the number of allowed disjunctions, and naive exploration of this space may become infeasible as dataset size grows. Furthermore, the introduction of OR operators raises important practical questions. How should the quality of these subgroups be measured fairly? Can the computational challenges be addressed through pruning strategies without sacrificing the discovery of high-quality patterns?

Despite the potential benefits of disjunctive representations and these important questions, disjunctive subgroup discovery remains significantly understudied. The existing literature contains isolated efforts that explore specific forms of disjunction, but systematic investigation of the general OR operator within standard subgroup discovery frameworks remains limited. Prior work such as SDIGA [2] explores disjunctive normal form rules in the context of fuzzy logic, introducing additional complexity beyond the core disjunction operator. Similarly, DISGROU [3] permits disjunction only within individual numerical attributes, not across different features. Consequently, several fundamental questions remain unanswered. Does adding an OR condition consistently produce higher-quality subgroups across diverse datasets? What trade-offs between expressive

power and computational complexity arise when disjunctions are introduced? Can the expanded search space be explored efficiently through appropriate pruning techniques without degrading the quality of discovered patterns?

This thesis addresses these questions through a systematic investigation of disjunctive subgroup discovery. The investigation is guided by the following primary research question: *Does the inclusion of disjunctive queries in subgroup discovery lead to the identification of higher-quality subgroups compared to traditional conjunctive approaches, and what are the broader implications for the search process?* This primary question is decomposed into three specific sub-questions that guide the experimental design and analysis:

1. How does the use of disjunctive descriptions affect the distribution of WRAcc scores compared to conjunctive-only approaches?

2. What trade-offs between subgroup quality and search complexity arise when introducing disjunctions?

3. How can the search space for disjunctive subgroup descriptions be structured and explored efficiently?

1.1 Thesis overview

The remainder of this thesis is organized as follows. Section 2 provides the necessary background on subgroup discovery, including formal definitions of both conjunctive and disjunctive queries, the Weighted Relative Accuracy (WRAcc) quality measure, and the analytical tools used throughout the experiments: ROC analysis, paired t-tests, and Cohen's d effect size. Section 3 reviews existing literature on classical subgroup discovery algorithms and prior disjunctive approaches, highlighting the gap that this thesis addresses. Section 4 describes the nine UCI datasets used in the experiments, the preprocessing steps applied, and the configuration of the SubDisc framework. Section 5 presents the methods developed in this thesis: a case study illustrating the limitations of conjunctive rules, the conjunctive baseline configuration, a proof-of-concept method for single OR additions, the core pairwise combination method for forming disjunctive subgroups as unions of two conjunctive components, statistical evaluation using paired t-tests and Cohen's d, a search depth experiment to determine the optimal depth setting, a multi-dataset evaluation across eight additional datasets, and a theoretical upper bound for pruning OR combinations to enable future scalability. Section 6 summarizes the findings, discusses the relation to existing work, acknowledges limitations, and outlines directions for future research. The complete source code for all experiments is provided in Appendix A.

2 Background

This section presents the foundational concepts required to understand the remainder of this thesis. Subgroup discovery is formally defined, covering both the standard conjunctive formulation and the disjunctive extension that this thesis investigates. The Weighted Relative Accuracy (WRAcc) quality measure is introduced and its formulation is derived. Additionally, the analytical tools used in the experimental evaluation are presented: ROC analysis, paired t-tests, and Cohen's d effect size.

2.1 Subgroup Discovery

Subgroup discovery is a data mining task that aims to identify subpopulations, called subgroups, that exhibit statistically interesting behavior with respect to a target variable of interest [7]. Rather than building a model that predicts the target for every instance, subgroup discovery focuses on finding specific subsets of the data where the target distribution deviates significantly from the global average. The goal is to find patterns that explain or characterize a particular property of interest.

One of the key strengths of subgroup discovery is the interpretability of its results. Unlike complex black-box models such as deep neural networks or ensemble methods, subgroup descriptions are expressed as simple logical rules that are generally more interpretable than black-box models, though their comprehensibility depends on the complexity of the rule and the expert's familiarity with the domain. For instance, a rule like "Age $>$ 50 $\wedge$ Blood Pressure $>$ 140" is immediately comprehensible to a medical professional. This interpretability is particularly important in domains where decisions must be transparent and explainable, such as clinical diagnosis or credit risk assessment.

Furthermore, subgroup descriptions are often actionable, making them valuable for decision support. When a subgroup with an unusually high proportion of positive instances is discovered, domain experts can investigate the underlying causes and potentially design targeted interventions. For example, if a marketing analyst discovers that customers with "Age $<$ 20 $\wedge$ Region = South" have an unusually high purchase rate for a product, a targeted advertising campaign can be launched specifically for that group.

Subgroup discovery occupies a unique position between descriptive and predictive data mining tasks [7]. Descriptive tasks, such as clustering or association rule mining, aim to summarize data or identify co-occurring patterns without focusing on a specific target variable. Predictive tasks, such as classification or regression, aim to construct models that accurately predict the target for unseen instances. Subgroup discovery combines elements of both approaches: it is *target-oriented* like predictive tasks, yet it produces interpretable descriptive rules rather than a global predictive model. Subgroup discovery can accommodate various target types, including binary, categorical, and numerical variables. This thesis focuses on the binary setting, where the target indicates the presence or absence of a property of interest (e.g., purchased a product or has a disease).

2.1.1 Formal Definition Conjunctive Queries

Let a dataset D consist of N instances, each described by a set of features $X_1, \ldots, X_m$ and a binary target variable $Y \in \{0, 1\}$. Without loss of generality, the positive class $Y = 1$ is taken as the

property of interest. For example, in a marketing application, $Y = 1$ might indicate that a customer purchased a product, while in a medical application it might indicate the presence of a disease.

A subgroup is described by a condition C, which is a conjunction of individual conditions on single features. Each individual condition takes the form $X_i \ \theta \ v$, where $\theta \in \{=\} \cup \{<, \leq, >, \geq\}$ and v is a value from the domain of X_i. For categorical features, only equality conditions are used (e.g., Region = North). For numerical features, only inequality conditions are used (e.g., Age > 30). A subgroup description can consist of a single condition or multiple conditions joined by an AND operator:

$$C = (X_{i_1} \ \theta \ v_1) \wedge (X_{i_2} \ \theta \ v_2) \wedge \cdots \wedge (X_{i_k} \ \theta \ v_k) \tag{1}$$

Here, each X_{i_j} denotes a distinct feature (no feature appears more than once in a conjunction, as repeating the same feature would be redundant). The set of instances satisfying C, a "subgroup", is denoted $S \subseteq D$. The size of the subgroup is $|S|$, and the number of positive instances within the subgroup is $\text{pos}_S = |\{i \in S : Y_i = 1\}|$.

2.1.2 Formal Definition Disjunctive Queries

While the previous definition covers conjunctive subgroup descriptions, this thesis also considers disjunctive queries. In logic, disjunctive formulas can take two canonical forms. The first is Disjunctive Normal Form (DNF), where a formula is a disjunction of conjunctive clauses:

$$C_{\text{DNF}} = (A_1 \wedge A_2 \wedge \cdots \wedge A_{k_1}) \vee (B_1 \wedge B_2 \wedge \cdots \wedge B_{k_2}) \vee \ldots \tag{2}$$

Here, $A_1, A_2, \ldots, B_1, B_2, \ldots$ are individual conditions of the form $X_i \ \theta \ v$, as defined in the conjunctive case. The subscripts $k_1, k_2, \ldots$ denote the number of conditions in each conjunctive block. A DNF rule captures instances that satisfy at least one of several distinct profiles, making it suitable for representing unions of subpopulations.

The second form is Conjunctive Normal Form (CNF), where a formula is a conjunction of disjunctive clauses:

$$C_{\text{CNF}} = (A_1 \vee A_2 \vee \cdots \vee A_{k_1}) \wedge (B_1 \vee B_2 \vee \cdots \vee B_{k_2}) \wedge \ldots \tag{3}$$

A CNF rule captures instances that must satisfy at least one condition from each of several groups, representing intersections of unions rather than unions of intersections. This thesis focuses exclusively on DNF rules, since they capture the type of pattern central to this work: multiple distinct profiles that all lead to the same outcome. CNF rules, by contrast, represent intersections of unions and are not considered here. For a DNF rule C_{DNF}, the set of instances satisfying it is the union of the instances satisfying each conjunctive block. Unlike purely conjunctive descriptions, the same feature may appear in multiple disjuncts, as each block describes an independent profile that leads to the target property.

2.2 Weighted Relative Accuracy

The Weighted Relative Accuracy (WRAcc) measure is a widely used quality function in subgroup discovery. Unlike simple accuracy, which only considers the proportion of correct predictions, WRAcc explicitly balances two objectives: the subgroup should be sufficiently large (coverage)

to be meaningful, yet its distribution on the target variable should deviate from the global norm (unusualness). This trade-off is essential because a very specific subgroup may have high unusualness but low coverage (and may be spurious), while a very general subgroup may have high coverage but low unusualness (and reveals nothing new). Weighted Relative Accuracy formalizes this trade-off in a single score [12].

2.2.1 Definition

For a subgroup S described by condition C and a binary target variable with positive class H, WRAcc is defined as [12]:

$$\text{WRAcc}(S) = p(C) \times \big(p(H|C) - p(H)\big) \tag{4}$$

where:

- $p(C) = \frac{|S|}{|D|}$ is the coverage, i.e., the proportion of instances in the subgroup;
- $p(H|C) = \frac{\text{pos}_S}{|S|}$ is the precision, i.e., the proportion of positive instances within the subgroup;
- $p(H) = \frac{\text{pos}_D}{|D|}$ is the global positive rate, i.e., the proportion of positive instances in the dataset.

The measure can be interpreted as the coverage multiplied by the gain in precision over the baseline. A subgroup that is no different from the global distribution ($p(H|C) = p(H)$) receives a score of zero, regardless of its coverage. A subgroup that performs worse than the baseline ($p(H|C) < p(H)$) receives a negative score, indicating a subgroup depleted of positives compared to the global distribution. A subgroup that performs better than the baseline ($p(H|C) > p(H)$) receives a positive score, indicating an enriched subgroup.

WRAcc balances coverage and gain: a subgroup with perfect precision ($p(H|C) = 1$) but very low coverage yields a low score because $p(C)$ is small; conversely, a subgroup with high coverage but negligible gain ($p(H|C) \approx p(H)$) also yields a low score. The score is maximized when both factors are substantial.

For a perfectly balanced dataset with $p(H) = 0.5$, WRAcc ranges between -0.25 and 0.25. The maximum of 0.25 occurs when a subgroup captures all positive instances ($p(H|C) = 1$), which forces $p(C) = 0.5$ since positives make up exactly half the data; any larger subgroup would necessarily include negatives, reducing precision and thus lowering the product. The minimum of -0.25 occurs symmetrically when a subgroup captures all negative instances. For imbalanced datasets, the WRAcc range is smaller due to a trade-off: if the positive class is rare ($p(H)$ low), the potential gain $p(H|C) - p(H)$ can be large, but the maximum coverage $p(C)$ is limited by the small number of positives; conversely, if the positive class is common ($p(H)$ high), coverage can be large but the potential gain is limited. The maximum absolute WRAcc is therefore achieved at $p(H) = 0.5$. Scores close to the relevant extreme (positive or negative) indicate strong associations between the subgroup condition and the target variable.

2.2.2 Simplified Formulation

By expanding the terms, the definition of WRAcc can be rearranged as follows:

$$\text{WRAcc}(S) = \frac{|S|}{|D|} \times \left(\frac{\text{pos}_S}{|S|} - \frac{\text{pos}_D}{|D|} \right) = \frac{\text{pos}_S}{|D|} - \frac{|S| \times \text{pos}_D}{|D|^2}$$

Note that $\frac{\text{pos}_D}{|D|}$ is precisely $p(H)$, the global positive rate. Substituting this gives:

$$\text{WRAcc}(S) = \frac{\text{pos}_S}{|D|} - \frac{|S| \times p(H)}{|D|}$$

Factoring out $\frac{1}{|D|}$ yields the compact form:

$$\text{WRAcc}(S) = \frac{1}{|D|} \times \big(\text{pos}_S - |S| \times p(H) \big) \tag{5}$$

This formulation expresses WRAcc directly in terms of two fundamental quantities: the number of positive instances in the subgroup (pos_S) and the total size of the subgroup ($|S|$). The dataset size $|D|$ and the global positive rate $p(H)$ are constants for a given dataset. This form reveals how WRAcc depends on pos_S and $|S|$ independently, making it easier to analyze how changes in these quantities affect the score. In particular, the WRAcc of a subgroup increases when pos_S is large relative to $|S| \times p(H)$.

2.2.3 Interpretation and Properties

WRAcc evaluates a subgroup by comparing the observed number of positive instances within the subgroup to the number that would be expected under the global distribution. A positive WRAcc value indicates that the subgroup contains more positive instances than expected, while a negative value indicates fewer positive instances than expected. Scores close to zero suggest that the subgroup behaves similarly to the overall dataset and therefore does not reveal particularly informative structure.

An important property of WRAcc is that it naturally penalizes overly small or overly general subgroups. Because the score is weighted by coverage, subgroups containing only a few instances cannot obtain very high scores solely through extreme precision. Conversely, very large subgroups are unlikely to score highly unless their target distribution differs substantially from the global baseline. This balance prevents the measure from favoring highly specific patterns that may be caused by noise, while also discouraging trivial patterns that provide little additional insight.

As a result, WRAcc is well suited for subgroup discovery tasks in which the objective is to identify statistically meaningful and interpretable patterns. By simultaneously accounting for subgroup size and deviation from the baseline distribution, the measure emphasizes subgroups that are both reliable and practically relevant for further analysis, making it a popular choice in comparative studies of subgroup discovery algorithms [6].

2.3 Receiver Operating Characteristic Analysis

Beyond measuring subgroup quality with WRAcc, this thesis also employs ROC analysis to evaluate the collective performance of multiple subgroups. Receiver Operating Characteristic (ROC) analysis is a standard tool for evaluating binary classifiers. In the context of subgroup discovery, a subgroup can be interpreted as a binary classifier that predicts the positive class for all instances satisfying

its conditions. This perspective allows subgroup quality to be assessed in terms of two fundamental quantities: the true positive rate (TPR) and the false positive rate (FPR).

For a subgroup S described by condition C, with pos_S positive instances in the subgroup, pos_D total positives in the dataset, $|S|$ the subgroup size, and $|D|$ the total number of instances, TPR and FPR are defined as:

$$\text{TPR}(S) = \frac{\text{pos}_S}{\text{pos}_D}, \qquad \text{FPR}(S) = \frac{|S| - \text{pos}_S}{|D| - \text{pos}_D} \tag{6}$$

TPR measures the proportion of actual positives correctly covered by the subgroup (also known as recall or sensitivity). FPR measures the proportion of negatives incorrectly included. A perfect subgroup would achieve TPR = 1 and FPR = 0, while random guessing corresponds to the diagonal line where TPR = FPR.

When multiple subgroups are evaluated, they can be plotted in the ROC space with FPR on the horizontal axis and TPR on the vertical axis. The convex hull is the boundary that encloses all points, connecting only the outermost ones. It is constructed by iteratively removing points that lie below the line connecting two others. The area under this convex hull (AUC) quantifies the overall trade-off between TPR and FPR achieved by a set of subgroups. Higher AUC values indicate better performance across all operating points. AUC complements measures like WRAcc because it captures collective performance across multiple subgroups rather than evaluating each subgroup in isolation. Furthermore, if two subgroups have similar WRAcc but different positions in ROC space, they capture different profiles of positive instances: one may cover many positives at the cost of including many negatives, while another may cover fewer positives but with near-perfect precision.

2.4 Statistical Significance Testing

To determine whether observed differences in subgroup quality are statistically meaningful rather than attributable to random variation, paired t-tests can be used. The paired design is appropriate when each observation in one sample is naturally paired with an observation in the other sample, for instance when comparing a disjunctive subgroup with the conjunctive subgroups from which it was formed.

Given k paired observations with differences d_i, the mean difference is $\bar{d} = \frac{1}{k}\sum_{i=1}^{k} d_i$ and the standard deviation of the differences is s_d. The t-statistic is computed as:

$$t = \frac{\bar{d}}{s_d/\sqrt{k}} \tag{7}$$

Under the null hypothesis that the true mean difference is zero, this t-statistic follows a t-distribution with $k - 1$ degrees of freedom. The degrees of freedom represent the number of independent pieces of information available for estimating variability. The p-value is the probability of observing a t-statistic as extreme as the one calculated, under the assumption that the null hypothesis is true. A small p-value indicates that such an extreme result is unlikely to occur by random chance alone, providing evidence against the null hypothesis.

2.5 Effect Size Measurement

While the p-value indicates whether an effect exists, it does not reveal the magnitude of that effect. Effect sizes address this limitation by quantifying the strength of the observed difference in a standardized way, independent of the original measurement scale. Among the effect size measures available, Cohen's d is chosen for this thesis because it is well-established in the literature and its interpretation is intuitive [1]. For the paired comparisons in this thesis, the appropriate formulation is the paired version of Cohen's d [5]:

$$d = \frac{\bar{d}}{s_d} \tag{8}$$

Here, $\bar{d}$ is the mean of the paired differences (i.e., the average improvement), and s_d is the standard deviation of those differences. Dividing the mean improvement by its standard deviation expresses the effect size in units of variability. A larger d means the observed improvement is large compared to the natural fluctuations in the data, indicating a more substantial and reliable effect. Following Cohen's conventions, values of 0.2, 0.5, and 0.8 are interpreted as small, medium, and large effects respectively [1].

3 Related Work

This section situates the contribution of this thesis within the existing literature on subgroup discovery. The related work is organized into two main subsections. First, classical conjunctive subgroup discovery algorithms are examined, including their search strategies, quality measures, and known limitations. Second, prior work that has explored disjunctive or extended representations is surveyed, with particular attention to the forms of disjunction supported and the limitations of existing approaches.

3.1 Classical Subgroup Discovery Algorithms

The EXPLORA system [9] was among the first to formalize subgroup discovery as a distinct data mining task, separate from classification (which predicts for all instances) or association rule mining (which has no target variable). EXPLORA employs a multi-strategy approach that combines different search heuristics depending on the characteristics of the data being analyzed. While powerful for its time, EXPLORA is primarily designed for data stored in a single table, where all instances share the same set of attributes.

The MIDOS algorithm [14] extends subgroup discovery to multi-relational databases, enabling the discovery of subgroups that involve conditions spanning multiple tables in a relational database. To make this search feasible, MIDOS introduced several innovations: declarative bias based on foreign keys to focus the search on meaningful table connections, adapted single-table evaluation functions to the multi-relational setting, and employed optimistic estimate pruning (cutting off branches that cannot possibly beat the current best subgroup) and minimal support pruning (ignoring subgroups that are too small to be interesting) to avoid exploring unpromising search paths. Despite their innovations, neither EXPLORA nor MIDOS support disjunctive OR operators and focus exclusively on conjunctive descriptions.

3.2 Disjunctive Approaches

While conjunctive representations dominate the subgroup discovery literature, several lines of research have explored richer rule forms that move beyond simple AND combinations. These approaches vary in the type of disjunction they support, ranging from general disjunctive normal form rules to disjunctions confined to individual numerical attributes. This section reviews two representative methods: SDIGA, which employs fuzzy logic to learn rules in disjunctive normal form, and DISGROU, which focuses on discontinuous intervals within single numerical attributes. Both methods demonstrate the potential of disjunctive representations but suffer from limitations that this thesis aims to address.

3.2.1 SDIGA: Disjunctive Normal Form with Fuzzy Logic

SDIGA (Subgroup Discovery Iterative Genetic Algorithm) [2] represents an early and influential effort to move beyond conjunctions. SDIGA discovers rules in disjunctive normal form (DNF) using fuzzy logic, where conditions such as “age is young” are represented with membership degrees rather than sharp thresholds. The algorithm employs a genetic search with specialized crossover and mutation operators tailored to the DNF representation. The iterative nature of SDIGA allows

the algorithm to refine discovered rules over multiple generations, gradually improving both quality and interpretability. Experiments on marketing data demonstrated that DNF rules could capture multiple distinct customer profiles within a single subgroup. The study reported that SDIGA achieved strong performance on coverage and support compared to other subgroup discovery algorithms.

However, SDIGA's use of fuzzy logic introduces several complications that make it difficult to isolate the effect of disjunction from the additional complexity of the fuzzy framework. First, fuzzy membership functions require specifying parameters such as the shape and position of each fuzzy set. These parameters are typically determined by domain experts or through additional optimization, adding a layer of complexity beyond the core disjunction operator. Second, evaluating fuzzy rules requires specialized quality measures that account for partial membership rather than crisp set membership. This complicates direct comparison with conjunctive rules, where an instance either satisfies a condition or does not. Consequently, any improvement in subgroup quality from SDIGA could potentially be attributed to the fuzziness of the representation rather than the disjunctive structure itself, making it unclear whether OR operators alone are beneficial. Furthermore, the genetic search may not scale efficiently to large datasets, as genetic algorithms typically require many generations to converge and are sensitive to parameter settings such as population size, crossover rate, and mutation rate.

3.2.2 DISGROU: Disjunction within Numerical Attributes

DISGROU [3] takes a different approach, focusing specifically on disjunction within numerical attributes. Rather than allowing OR across different features (as in SDIGA's DNF rules), DISGROU permits a single numerical attribute to take a union of disjoint intervals, such as $30 \leq \text{age} \leq 35 \vee 40 \leq \text{age} \leq 50$. This can be viewed as a restricted form of Conjunctive Normal Form (CNF), where the rule consists of a single conjunctive clause containing only one attribute, with that attribute allowing a disjunction of intervals. More generally, a full CNF rule would be $(A \vee B) \wedge (C \vee D)$, requiring the union of intervals across multiple attributes; DISGROU restricts this to just one attribute: $(X \in I_1 \vee X \in I_2) \wedge \top$. This allows subgroups to exclude irrelevant middle regions that would otherwise be forced into a continuous interval to maintain contiguity. For example, if instances with age between 36 and 39 do not exhibit the target behavior, a continuous interval from 30 to 50 would incorrectly include these instances, diluting subgroup quality. A discontinuous interval can exclude this middle region, potentially improving quality while still covering the relevant younger and older age ranges.

The DISGROU algorithm works directly on numerical data without requiring prior discretization, avoiding information loss that can occur when continuous values are binned into discrete categories. The search procedure is specialized for identifying promising interval combinations, using a support threshold that only retains subgroups covering at least a minimum number of instances (e.g., 25). This filters out very small subgroups whose unusual target distribution might be due to chance rather than a meaningful pattern. Experimental results on several datasets showed that DISGROU frequently outperformed methods restricted to continuous intervals, demonstrating that allowing discontinuous intervals can improve subgroup quality in certain contexts. However, DISGROU's disjunction capability is confined to individual attributes; the algorithm does not support disjunctions across different features. For example, DISGROU can represent $(\text{age} \leq 30 \vee \text{age} \geq 50)$ but cannot represent $(\text{age} \leq 30) \vee (\text{income} > 50000)$. This limitation restricts

the algorithm's ability to capture patterns that involve alternative profiles defined by different combinations of features.

In summary, while SDIGA and DISGROU demonstrate the potential of disjunctive representations, both suffer from limitations that prevent a clean assessment of the OR operator's standalone effect: SDIGA couples disjunction with fuzzy logic, and DISGROU restricts disjunction to single attributes. This thesis addresses these gaps by investigating the general OR operator within a standard crisp subgroup discovery framework, allowing disjunctions across arbitrary combinations of features without additional complexity.

4 Data

This section describes the experimental setup used throughout this thesis. First, the datasets selected for evaluating disjunctive subgroup discovery are presented, including the preprocessing steps and their key characteristics. Second, the configuration of SubDisc, the subgroup discovery framework used to generate the conjunctive baseline, is detailed.

4.1 Datasets

The evaluation of disjunctive subgroup discovery requires a diverse collection of datasets to ensure that observed effects are not artifacts of specific data characteristics. Rather than focusing on a single domain, the selection spans multiple application areas to test whether the disjunctive method performs consistently well across different types of data. All datasets are drawn from the UCI Machine Learning Repository [8], a well-established collection of benchmark data for empirical analysis, enabling result verification and providing a basis for future comparative evaluations.

4.1.1 Data Selection

The UCI Machine Learning Repository provides curated, publicly accessible datasets. For this study, a total of nine datasets were selected, providing sufficient diversity to assess generalizability across different data characteristics while keeping the computational experiment manageable. The evaluation proceeds in two stages. First, the Real Estate dataset is used for initial testing. With 414 instances, six numerical features, intuitive attributes, and no missing values, this dataset allows rapid experimentation and straightforward interpretation of discovered subgroups before scaling to multiple datasets. Second, the remaining eight datasets are added to assess whether the findings generalize beyond this specific case. Together, the nine datasets span a wide range of characteristics, as reflected in the following four selection criteria.

The datasets span multiple domains, including real estate valuation, education, acoustics, food science, civil engineering, ecology, solar physics, automotive engineering, and marine biology. Second, the selection covers a broad range of instance counts, from 398 to 4,898, testing whether disjunctive methods scale across dataset sizes. Third, feature dimensionality varies from 5 to 30, encompassing both low-dimensional problems where patterns are easily interpretable and high-dimensional problems where positive instances may be scattered across many attributes. Fourth, the datasets include both purely numerical data and mixed categorical-numerical data, allowing evaluation of whether the method generalizes across different data types.

4.1.2 Data Preprocessing

All datasets undergo identical preprocessing to ensure experimental consistency. Datasets with ID columns have these columns removed, as they serve only as instance identifiers and contain no predictive information. For the Auto MPG dataset, the column containing the vehicle names is similarly removed.

Categorical features are treated as nominal types, meaning they are compared using equality rather than inequality, which would be meaningless for unordered categories. This avoids converting categories to arbitrary numerical codes that could imply false ordering relationships.

For datasets with temporal information (Forest Fires), month and day names are converted to numerical values (1-12 for months, 1-7 for days) so that they can be used as features. For the Wine Quality dataset, the color column (red or white) is excluded, as it indicates which of the two original datasets the instance came from rather than being a predictive feature.

For all datasets with numerical targets, the target is converted to a binary class by splitting at the median value. This yields a balanced split that avoids the complications of extreme class imbalance. When missing values are encountered, instances with missing values are removed.

4.1.3 Dataset Characteristics

Table 1 summarizes the key characteristics of the datasets used in this study.

Table 1: Summary of UCI datasets used in the experiment. For each dataset, the table reports the UCI identifier, the number of instances, the number of features (excluding the target), and the positive class definition. The positive class is defined as above the median for continuous targets.

Dataset	**UCI ID**	**# Instances**	**# Features**	**Positive Class**
Real Estate	477	414	6	Price above median
Student Performance	320	649	30	Final grade above median
Airfoil Noise	291	1,503	5	Noise above median
Wine Quality	186	4,898	11	Quality above median
Concrete Strength	165	1,030	8	Strength above median
Forest Fires	162	517	12	Burned area above median
Solar Flare	89	1,389	10	Common flares above median
Auto MPG	9	398	7	MPG above median
Abalone	1	4,177	8	Rings above median

4.1.4 Dataset Description

Real Estate. The Real Estate dataset (UCI ID 477) contains 414 instances of property transactions in Sindian District, New Taipei City, Taiwan. Each instance is described by six numerical features: transaction date, house age (years), distance to the nearest Mass Rapid Transit station (meters), number of convenience stores within walking distance, latitude, and longitude. The target variable is house price per unit area (10,000 New Taiwan Dollars per Ping). For subgroup discovery, the binary target indicates whether the price exceeds the median value.

Student Performance. The Student Performance dataset (UCI ID 320) records 649 instances of secondary education students from two Portuguese schools, covering both mathematics and Portuguese language courses. Thirty features capture demographic, social, and school-related attributes such as age, sex, parental education and occupation, study time, past failures, alcohol consumption, and absences. The target is the final grade (G3) for the respective course. With its high proportion of nominal attributes, this dataset helps determine whether the disjunctive method is broadly applicable across different data representations.

Airfoil Self-Noise. The Airfoil Self-Noise dataset (UCI ID 291) contains 1,503 instances from NASA aerodynamic and acoustic tests of NACA 0012 airfoils in an anechoic wind tunnel. Five features are provided: frequency (Hz), angle of attack (degrees), chord length (meters), free-stream velocity (meters per second), and suction side displacement thickness (meters). The target is scaled sound pressure level (decibels). With 1,503 instances and only five features, this dataset represents a unique regime where the instance count is high relative to the number of features.

Wine Quality. The Wine Quality dataset (UCI ID 186) includes 4,898 instances of red and white variants of Portuguese Vinho Verde wine. Eleven physicochemical features measure properties such as acidity, sugar content, sulfur dioxide levels, pH, and alcohol percentage. The target is sensory quality score (0–10), binarized at the median. The classes are ordered but imbalanced; for example, there are many more normal wines than excellent or poor ones. The mixture of correlated chemical measurements presents a feature interaction structure where disjunctive combinations may capture distinct quality profiles.

Concrete Compressive Strength. The Concrete Compressive Strength dataset (UCI ID 165) contains 1,030 instances with eight input variables measuring concrete composition: cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, fine aggregate, and age (days). The target is compressive strength measured in megapascals. All features are numerical with no missing values.

Forest Fires. The Forest Fires dataset (UCI ID 162) records 517 instances of forest fires in the Montesinho Natural Park, Portugal. Twelve features describe each fire: spatial coordinates, temporal information (month, day of week), Fire Weather Index components (FFMC, DMC, DC, ISI), and meteorological conditions (temperature, relative humidity, wind speed, rainfall). The target is the burned area in hectares, binarized at the median. This dataset introduces temporal and meteorological features, allowing us to see whether disjunctive patterns can capture seasonal or weather-dependent risk profiles.

Solar Flare. The Solar Flare dataset (UCI ID 89) contains 1,389 instances representing active regions on the sun. The ten features capture aspects of solar activity: region appearance (Zurich class, spot size, spot distribution), activity status (activity, evolution, previous flare activity), complexity (historically-complex, became complex on this pass), and size measurements (area, area of largest spot). The target is the number of common (C-class) flares produced in the following 24 hours, binarized at the median. This dataset represents the solar physics domain and tests whether disjunctive descriptions can handle categorical features with domain-specific codes.

Auto MPG. The Auto MPG dataset (UCI ID 9) comprises 398 instances of automobiles. Seven features describe each vehicle: displacement, the number of cylinders, horsepower, weight, acceleration, model year, and origin. The target variable is fuel consumption measured in miles per gallon, binarized at the median, and instances with missing horsepower values are removed. This dataset includes a categorical feature (origin) and correlated numerical predictors, providing a different type of data structure.

Abalone. The Abalone dataset (UCI ID 1) has 4,177 instances of abalone. Features include sex (categorical: male, female, infant), various shell measurements (length, diameter, height), and weight measurements (whole, shucked, viscera, shell). The target is the number of rings, where adding 1.5 gives the age in years, binarized at the median to distinguish older abalone. This dataset combines a categorical feature with multiple correlated continuous measurements of size and weight, allowing us to see whether disjunctive descriptions can capture age-related patterns that involve different combinations of physical measurements.

4.2 Subdisc Configuration

SubDisc is an open-source Java framework for subgroup discovery that provides a flexible and configurable implementation of subgroup discovery algorithms [10]. Unlike many specialized implementations tailored to specific problem settings, SubDisc is designed to be generic. It can handle multiple data types (binary, nominal, and numeric) for both input attributes and target variables, and supports various subgroup discovery settings. The framework has been used in numerous scientific publications across domains such as bioinformatics, fraud detection, and Bayesian networks [10].

4.2.1 Key Configurable Parameters

SubDisc takes as input a dataset and a target variable, and outputs a ranked list of subgroups according to a user-specified quality measure. The behavior of the algorithm can be controlled through several key parameters:

- **Search depth**: the maximum number of conditions allowed in a subgroup description. This parameter controls the complexity of the discovered subgroups and directly affects the size of the search space.
- **Minimum coverage**: the minimum number of instances a subgroup must cover to be considered. While WRAcc already penalizes very small subgroups through its coverage weighting, this parameter serves as a pre-filter to discard them early, reducing computation time by avoiding evaluation of subgroups that are likely spurious or lack practical utility.
- **Numeric strategy**: determines how numerical attributes are handled during search. This thesis uses a strategy that finds the optimal split point for a numerical condition by evaluating all possible thresholds. While more computationally expensive than evaluating thresholds at fixed percentile intervals, this approach ensures that no potentially better split point is missed.
- **Numeric operator setting**: determines which comparison operators are allowed for numerical attributes. This thesis uses the default setting, which considers only $\leq$ and $\geq$ conditions (equality is not allowed for numerical features).
- **Quality measure**: specifies which quality measure to use for ranking subgroups. This thesis uses WRAcc (Weighted Relative Accuracy).
- **Minimum quality threshold**: a threshold below which subgroups are discarded. This helps filter out low-quality results.

- **Search strategy**: determines how the search space is traversed. This thesis uses the default strategy, beam search, which is a heuristic that maintains a fixed-size set (the beam) of the most promising candidate subgroups at each level of refinement.

4.2.2 Python Integration: pySubDisc

The Python wrapper pySubDisc provides integration between SubDisc and the Python environment [13]. SubDisc is implemented in Java, making it well-suited for the computationally intensive search tasks at the core of subgroup discovery, while pySubDisc handles data loading and result processing in Python. pySubDisc provides a Python interface to the Java backend, allowing users to load data from pandas DataFrames, configure search parameters, run the algorithm, and retrieve results as DataFrames. This enables the entire workflow to be written in Python without direct Java interaction.

5 Approach

This section presents the methods developed to investigate whether disjunctive queries improve subgroup quality. A case study first illustrates that disjunctive descriptions can outperform conjunctive ones using an L-shaped pattern from the Real Estate dataset. A conjunctive baseline is then established using SubDisc with fixed parameters. A proof-of-concept method tests whether single OR additions are beneficial, followed by the core pairwise combination method that forms disjunctive subgroups as unions of two conjunctive components. Statistical evaluation using paired t-tests and Cohen's d determines significance. A search depth experiment identifies an optimal depth setting, after which the method is evaluated on eight additional datasets. Finally, a theoretical upper bound on OR combination quality is derived to support future pruning strategies.

5.1 Case Study

The weakness of conjunctive subgroup descriptions is not merely theoretical: it manifests clearly in real data. To build intuition for why disjunctive descriptions can outperform conjunctive ones, this subsection examines a concrete example from the Real Estate dataset before proceeding to the systematic experimental evaluation. We consider an L-shaped pattern that no single conjunctive rule can capture without loss of quality, so that a disjunctive subgroup achieves substantially higher quality than any conjunctive alternative. This case study demonstrates exactly when and why OR combinations beat AND-only rules.

The example uses two features from the Real Estate dataset: house age and distance to the nearest Mass Rapid Transit (MRT) station. The target variable indicates whether the house price per unit area exceeds the median value.

During exploratory analysis, a particularly instructive pattern emerged that clearly illustrates the benefits of disjunctive descriptions. This pattern consists of two conjunctive components that we can connect with a logical OR:

- **Component A**: house age ≤ 35.8 AND distance to MRT ≤ 737.9161
- **Component B**: house age ≤ 11.8 AND distance to MRT ≤ 1406.43

Component A describes houses within close proximity to a station (distance ≤ 737.9 meters) that are relatively young (age ≤ 35.8 years). Component B describes even newer houses (age ≤ 11.8 years) within a slightly larger distance (≤ 1406.4 meters). These two descriptions cover qualitatively different profiles of high-priced properties: one emphasizing very close transit access with somewhat older houses, the other emphasizing very new houses with slightly more distant transit access. A purely conjunctive rule cannot capture both profiles simultaneously without including instances that satisfy neither description, i.e., houses that are slightly older and also slightly further.

Figure 1 visualises the spatial aspect of this pattern using the two features that form its basis: house age and distance to MRT station. Component A corresponds to the green rectangle, and component B corresponds to the orange rectangle.

Table 2 reports the WRAcc components for each subgroup. Component A achieves WRAcc 0.169 with precision 0.807 and coverage 0.551. Component B achieves WRAcc 0.103 with higher precision (0.938) but lower coverage (0.234). Their disjunctive combination achieves WRAcc 0.173 with coverage 0.568 and precision 0.804, outperforming both individual components.

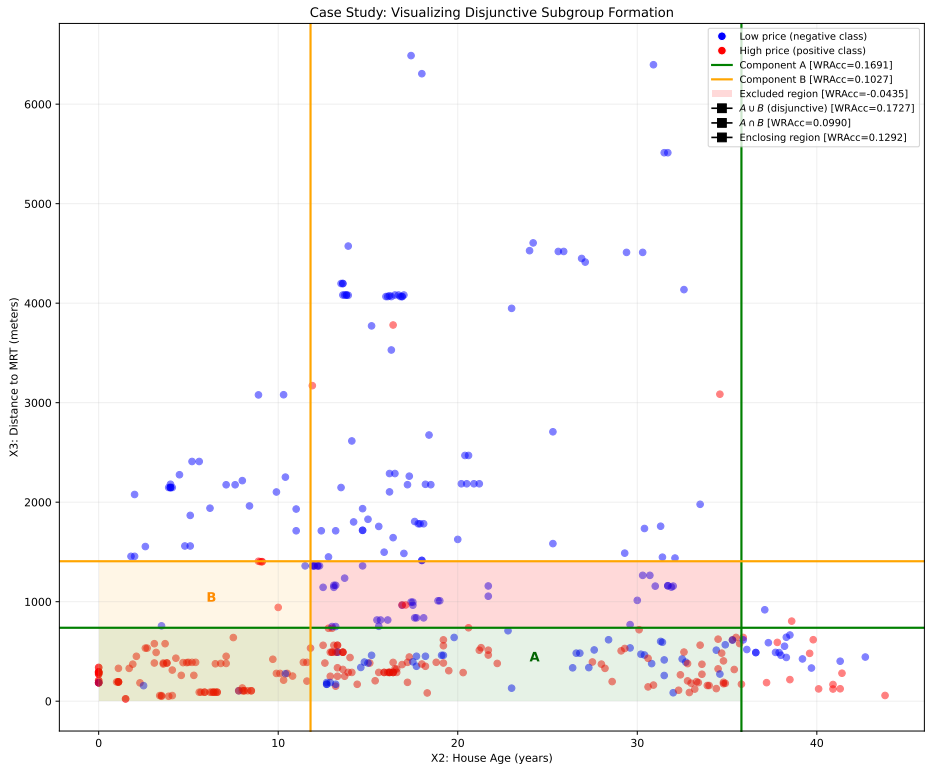

Figure 1: Decision boundaries of the two conjunctive components (A: green, B: orange) forming a disjunctive subgroup on the Real Estate dataset. Red points: high-price houses (positive class); blue points: low-price houses. The OR combination excludes the red "missing corner", achieving WRAcc 0.173 and outperforming the best conjunctive rule (0.129).

Crucially, the excluded region has negative WRAcc (-0.043), meaning its precision (0.050) is well below the global baseline of 0.500. Including this region would severely dilute subgroup quality: the enclosing conjunctive rule covering the entire bounding box (age ≤ 35.8, distance ≤ 1406.4) achieves only 0.129 WRAcc with precision 0.695, which is substantially lower than the disjunctive combination. This demonstrates exactly why conjunctive rules fail: they cannot exclude low-quality regions without also losing high-quality regions that are not contiguous.

The disjunctive combination succeeds precisely because it excludes this low-quality region. A single conjunctive rule that attempted to do the same would have to produce the enclosing conjunctive rule: age ≤ 35.8 AND distance ≤ 1406.4. As the table shows, this yields WRAcc 0.129, which is lower than the disjunctive combination. Alternatively, it could produce the intersection of the two components, which achieves very high precision (0.956) but low coverage (0.217), resulting in WRAcc 0.099: lower than both the disjunctive combination and the enclosing conjunctive rule. In both cases, the result is sub-optimal compared to an L-shaped rule. This is why the best approximation that a classic conjunction-only method can produce is to produce two independent rules instead of a single one.

Thus, the L-shaped pattern provides clear evidence that disjunctive descriptions can capture meaningful structures that purely conjunctive search systematically misses.

Table 2: Quantitative comparison of the L-shaped pattern components. $p(B)$: coverage, $p(H|B)$: precision, $p(H) = 0.5$: global positive rate. The disjunctive combination achieves WRAcc 0.173, outperforming both components (0.169, 0.103) and the enclosing conjunctive rule (0.129). The excluded region has negative WRAcc (-0.043), demonstrating that disjunctive descriptions can exclude low-quality regions.

Subgroup	**p(B)**	**p(H|B)**	**p(H)**	**WRAcc**
Component A (distance ≤ 737.9, age ≤ 35.8)	0.551	0.807	0.500	0.169
Component B (age ≤ 11.8, distance ≤ 1406.4)	0.234	0.938	0.500	0.103
Disjunctive combination (A $\cup$ B)	0.568	0.804	0.500	0.173
Enclosing conjunctive rule	0.664	0.695	0.500	0.129
Excluded region	0.097	0.050	0.500	-0.043
Intersection of components (A $\cap$ B)	0.217	0.956	0.500	0.099

5.2 Conjunctive Baseline

The case study above demonstrated that a disjunctive combination of two conjunctive subgroups can, at least in some cases, outperform either component alone. To systematically evaluate such disjunctive combinations, a set of conjunctive subgroups must first be discovered to serve as both the baseline for comparison and the building blocks for disjunctive construction. We extract those with SubDisc, configured with the following parameters, which remain fixed across all experiments.

The minimum coverage is set to 25 instances, which is roughly 6% of the smaller datasets and well under 1% of the largest. At this low threshold, the search explores the subgroup space broadly, excluding only the most spurious patterns. While setting the threshold even lower would be computationally infeasible, the chosen value captures virtually all meaningful subgroups without sacrificing speed for the sake of faster experiments. A higher threshold would raise concerns about missed results; here, the exploration can be considered exhaustive within the bounds of computational feasibility.

The numeric strategy evaluates all possible split points for numerical attributes to find the optimal threshold, which guarantees that no better split point is missed while still being fast enough given the modest size of the datasets. The quality measure is set to WRAcc, with a minimum threshold of 0.01 to filter out very low-quality subgroups. The search strategy is beam search, a heuristic that maintains a fixed-size set of the most promising candidates at each refinement level. For methods requiring a top-k parameter, k is fixed to 10. This value provides a sufficiently large set for meaningful comparison without overwhelming the analysis with low-quality patterns, and is consistent with common practice in subgroup discovery evaluation where top-k sets typically range from 10 to 100 subgroups.

Running SubDisc with these parameters on a dataset produces a list of subgroups ranked by WRAcc, each described by a conjunction of conditions, along with its coverage and WRAcc score. The resulting list serves two purposes. First, it provides the conjunctive baseline against which the disjunctive results are compared. Second, it supplies the building blocks for the disjunctive generation strategies.

5.3 Proof of Concept: Single Additional Condition

The simplest way to create a disjunctive subgroup is to extend an existing conjunctive subgroup by adding a single condition via the OR operator. This strategy serves as a lightweight proof-of-concept: can the simple approach of adding a single condition via an OR operator already improve the quality of an existing conjunctive subgroup? This approach is intentionally simple, serving as an initial probe into whether disjunctive extensions are worthy of more exhaustive investigation. If even this minimal disjunctive extension improves subgroup quality, it suggests that more complex disjunctive strategies may yield further gains.

The pseudocode for this method is provided in Algorithm 1. First, SubDisc is run to obtain the top-10 conjunctive subgroups. Second, a pool of candidate single conditions is generated. A single condition is the simplest possible building block for a subgroup description, taking the form of a single comparison on one feature, such as "Age > 30". For numerical features, thresholds are evaluated at regular percentile intervals (every 10%), and two conditions are created per threshold: $X_i > v$ and $X_i \leq v$. For categorical features, an equality condition is created for each unique value.

Third, for each of the top-10 conjunctive subgroups, every candidate condition in the pool is tested as an OR addition. This means each candidate condition is combined with the subgroup using the logical OR operator, creating a disjunctive rule that includes instances satisfying either the original subgroup condition or the candidate condition. The quality of this resulting disjunctive subgroup is then measured. The candidate condition that produces the highest WRAcc for each original conjunctive subgroup is retained.

Then, the algorithm outputs the augmented subgroups. Each output consists of the original conjunctive condition, the best candidate condition found during the search, and the disjunctive rule formed by combining them with OR. The original WRAcc, the new WRAcc after OR addition, and the improvement are also recorded.

The time complexity of this approach is $O(k \cdot m \cdot n)$, where k is the number of top subgroups (10), m is the number of candidate conditions, and n is the number of instances. To understand why, consider a concrete example. Suppose a dataset has $n = 500$ instances and 10 numerical features. For each numerical feature, thresholds are evaluated at 10% percentile intervals, producing 9 split points (10%, 20%, ..., 90%). Each split point generates two candidate conditions ($X_i > v$ and $X_i \leq v$), resulting in $2 \times 9 = 18$ conditions per numerical feature. With 10 numerical features, the total number of candidate conditions is $m = 10 \times 18 = 180$. The algorithm evaluates each of the top $k = 10$ subgroups against all $m = 180$ candidate conditions. For each evaluation, computing the union requires checking all $n = 500$ instances to determine which instances satisfy either the original subgroup condition or the candidate condition. Thus, the total number of operations is approximately $10 \times 180 \times 500 = 900,000$ instance checks. This is computationally trivial for modern hardware, making this approach a practical first step.

5.3.1 Experimental Evaluation

The proof-of-concept method was evaluated on the Real Estate dataset at depth 3. The results show that six out of ten subgroups improve after adding a single OR condition, with an average WRAcc improvement of approximately 0.008 among the improved subgroups. The average WRAcc of the original conjunctive subgroups is 0.168, while the average WRAcc of the OR-augmented subgroups rises to 0.173, a relative increase of 3.0%.

Algorithm 1 Proof-of-Concept: Single OR Addition to Conjunctive Subgroups

Require: Dataset D, top-k conjunctive subgroups $\mathcal{S} = \{S_1, \ldots, S_k\}$
Ensure: List of augmented subgroups $\mathcal{R}$ with WRAcc and improvement recorded for each

$\mathcal{C} \leftarrow \emptyset$
for each numerical feature X **do**
  for each percentile $p \in \{10, 20, \ldots, 90\}$ **do**
    $v \leftarrow p$-th percentile of X
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{(X > v), (X \leq v)\}$ ▷ Two directional conditions per threshold
  end for
end for
for each categorical feature X **do**
  for each unique value $v \in \text{domain}(X)$ **do**
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{(X = v)\}$ ▷ One equality condition per category
  end for
end for
$\mathcal{R} \leftarrow \emptyset$
for each subgroup $S \in \mathcal{S}$ **do**
  $(S_{best}, q_{best}) \leftarrow (S, \text{WRAcc}(S))$
  for each condition $c \in \mathcal{C}$ **do**
    $S_{new} \leftarrow S \cup \{i \in D : i \text{ satisfies } c\}$ ▷ Union: instances in S OR satisfying c
    $q_{new} \leftarrow \text{WRAcc}(S_{new})$
    if $q_{new} > q_{best}$ **then**
      $(S_{best}, q_{best}) \leftarrow (S_{new}, q_{new})$
    end if
  end for
  $\mathcal{R} \leftarrow \mathcal{R} \cup \{(S, S_{best}, q_{best})\}$
end for
return $\mathcal{R}$

Figure 2 plots the coverage and WRAcc values for the top-10 conjunctive subgroups and their best OR-augmented versions. The most notable effect is the increase in coverage: as expected, adding an OR condition expands the subgroup. What is more interesting, however, is that this coverage increase does not come at the cost of WRAcc; instead, WRAcc also improves in six out of ten cases. The improvements in WRAcc among the six successful cases range from approximately 0.005 to 0.010. This suggests that these subgroups had potential for expansion without substantially sacrificing precision, and the added instances were sufficiently enriched in positives to compensate for the increased coverage. The fact that a majority of subgroups improve with a single OR addition, and none decrease in quality, confirms that OR-expansion is a potentially interesting operation and justifies pursuing more comprehensive disjunctive strategies.

5.4 Pairwise Combination

While the proof-of-concept method restricts attention to adding a single condition using OR, the core approach of this thesis presented in this section, considers combinations of complete conjunctive

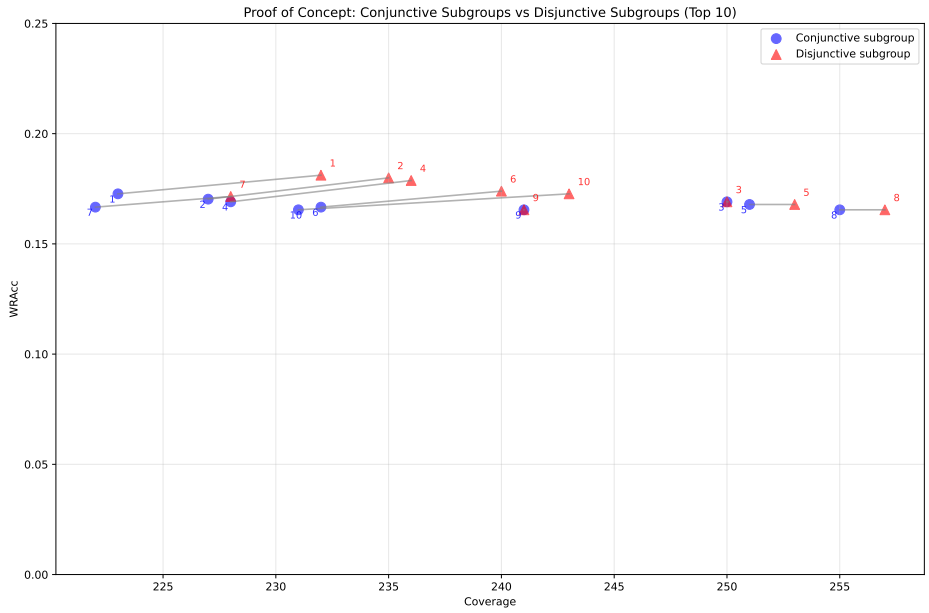

Figure 2: Coverage vs WRAcc for the top-10 conjunctive subgroups and their best OR-augmented versions on the Real Estate dataset (depth 3). Blue circles represent original conjunctive subgroups (average WRAcc = 0.168); red triangles represent the same subgroups after adding the best single condition via the OR operator (average WRAcc = 0.173). Six of ten subgroups show improvement, with gains ranging from 0.005 to 0.010.

subgroups. Rather than adding an atomic condition, this approach combines two fully-formed conjunctive subgroups, each potentially containing several conditions, into a single disjunctive description. The central question is whether the union of two such subgroups can achieve higher quality than either alone, similarly to what happened in the case study of Section 5.1.

The pseudocode for this method is provided in Algorithm 2. First, SubDisc is run to obtain the complete set of conjunctive subgroups (not just the top 10). This (large) set includes all discovered subgroups that meet the minimum coverage and quality thresholds. Second, a pool of candidate OR combinations is generated by considering all unique pairs of subgroups from this complete set. For each pair (S_i, S_j), the disjunctive subgroup is defined as the union $S_i \cup S_j$ (i.e., instances satisfying S_i OR S_j). The WRAcc of this new disjunctive subgroup is then calculated.

Third, a pair is only retained as a candidate if it satisfies a *strict improvement* condition: $\text{WRAcc}(S_i \cup S_j) > \max(\text{WRAcc}(S_i), \text{WRAcc}(S_j))$. In other words, the OR combination must outperform both of its components. A pair that merely matches the quality of its best component is discarded, as it adds no new value beyond simply presenting the better component alone, while potentially making interpretation more complex due to the additional disjunction. This condition ensures that only non-redundant, genuinely improving disjunctions are kept for further analysis.

Finally, the algorithm returns the best disjunctive candidates. For each retained pair, the output includes the two original conjunctive conditions, their individual WRAcc scores, the disjunctive rule formed by combining the conditions with OR, the WRAcc of the combined rule, and the improvement over the better of the two components. The candidates are then sorted by their WRAcc, and the top 10 are retained for comparison against the conjunctive baseline.

The time complexity of this method is $O(s^2 \cdot n)$, where s is the number of conjunctive subgroups and n is the number of instances. Suppose SubDisc discovers $s = 1000$ subgroups on a dataset with $n = 500$ instances. The number of unique pairs is $\frac{1000 \times 999}{2} = 499,500$. For each pair, computing the quality of the union requires evaluating the instances in $S_i \cup S_j$; in the worst case, this union may include nearly all n instances. Thus, the total number of instance checks is $499,500 \times 500 = 249,750,000$ operations. While potentially feasible for a single dataset, this becomes prohibitive for datasets with many attributes, where the number of subgroups can grow substantially.

Algorithm 2 Pairwise OR-Combination Method

Require: Dataset D, conjunctive subgroup set $\mathcal{S} = \{S_1, \ldots, S_m\}$
Ensure: Top-k disjunctive subgroups by WRAcc

$\mathcal{R} \leftarrow \emptyset$
for each pair $(S_i, S_j) \in \mathcal{S} \times \mathcal{S}$ where $i < j$ **do** ▷ Consider all unique pairs
  $S_{new} \leftarrow S_i \cup S_j$ ▷ Union: instances satisfying S_i OR S_j
  $q_{new} \leftarrow \text{WRAcc}(S_{new})$
  $q_{best} \leftarrow \max(\text{WRAcc}(S_i), \text{WRAcc}(S_j))$
  if $q_{new} > q_{best}$ **then** ▷ Only keep if union outperforms both components
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{(S_{new}, S_i, S_j, q_{new})\}$
  end if
end for
Select first k entries from $\mathcal{R}$ by WRAcc
return $\mathcal{R}$

5.4.1 Experimental Evaluation

Figure 3 plots the top-10 disjunctive subgroups alongside their two conjunctive components. Each disjunctive subgroup (red triangle) is connected to its two conjunctive components (blue points labeled A and B). In all ten cases, the disjunctive combination achieves higher WRAcc than both of its components. Taking the union of two subgroups always increases coverage, but this comes at a risk: the added instances may contain few positives, which would reduce precision and potentially lower WRAcc. The consistent improvement across all ten pairs therefore indicates that the additional instances brought in by each union are sufficiently enriched in positives to compensate for the increased coverage.

The magnitude of improvement is modest but consistent across the top-10 candidates, ranging from approximately 0.019 to 0.033. Notably, the best component in a disjunctive pair is not always the highest-ranked subgroup overall. This indicates that the pairwise combination method identifies complementary relationships that are not apparent from individual subgroup qualities alone. Two moderately good subgroups that cover different parts of the positive space can together outperform a single excellent subgroup that covers only one cluster. The value of a subgroup thus depends not only on its individual quality but also on how it complements other subgroups in the set.

Figure 4 shows the ROC space for both approaches, with each point representing either a conjunctive component (circles and squares) or a disjunctive combination (triangles), colored by their WRAcc values. The disjunctive subgroups occupy a specific region of the ROC space, the

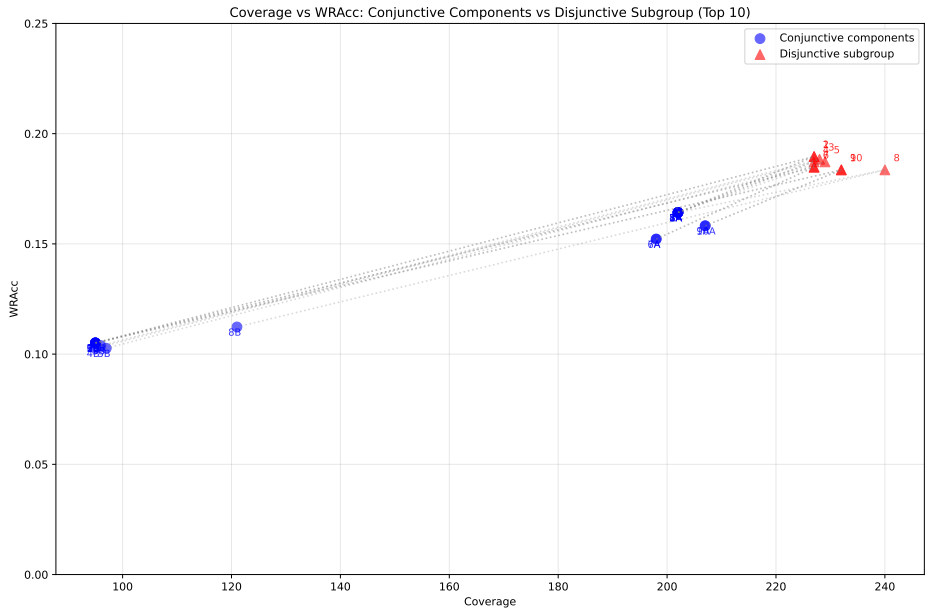

Figure 3: Coverage vs WRAcc for the top-10 disjunctive subgroups and their two conjunctive components on the Real Estate dataset (depth 3). Blue points labeled A and B are the two conjunctive components; red triangles are their OR combination. Gray dotted lines connect each component to its corresponding disjunctive subgroup. The numbers next to each point indicate the rank of the disjunctive subgroup (1 through 10), with the same number used for its two components and their OR combination. Note that the blue components are not necessarily the top-10 conjunctive subgroups overall; they are the specific components that form each disjunctive combination.

upper left quadrant, where they achieve high TPR (approximately 0.90-0.95) while maintaining moderate FPR (approximately 0.17-0.21), representing a favorable trade-off. This is exactly what we would expect if the union of two subgroups successfully covers more positives while adding relatively few negatives. The convex hull of the disjunctive points (AUC = 0.886) exceeds that of the conjunctive components alone (AUC = 0.856). This 0.030 increase in AUC indicates that the disjunctive method's subgroups achieve higher TPR for given FPR levels compared to the conjunctive components alone.

Examining which points lie on each hull reveals the complementary nature of the two approaches. The conjunctive hull is formed by a combination of high-precision components (very low FPR) and moderate-TPR components. The disjunctive hull improves upon this by adding combinations that achieve substantially higher TPR while maintaining acceptable FPR. The combined hull, which achieves the highest AUC (0.914), includes both the high-precision conjunctive components and the high-TPR disjunctive combinations. This demonstrates that disjunctive subgroups do not render conjunctive ones obsolete. Rather, the optimal set of subgroups contains both types: conjunctive rules capture extremely specific niches with very high precision (few false positives), while disjunctive combinations cover broader sets of positives with higher recall (capturing more true positives). This is an important insight: the two representation types serve different purposes, and a mixture of both yields the best overall performance.

In summary, the pairwise combination method consistently produces disjunctive subgroups

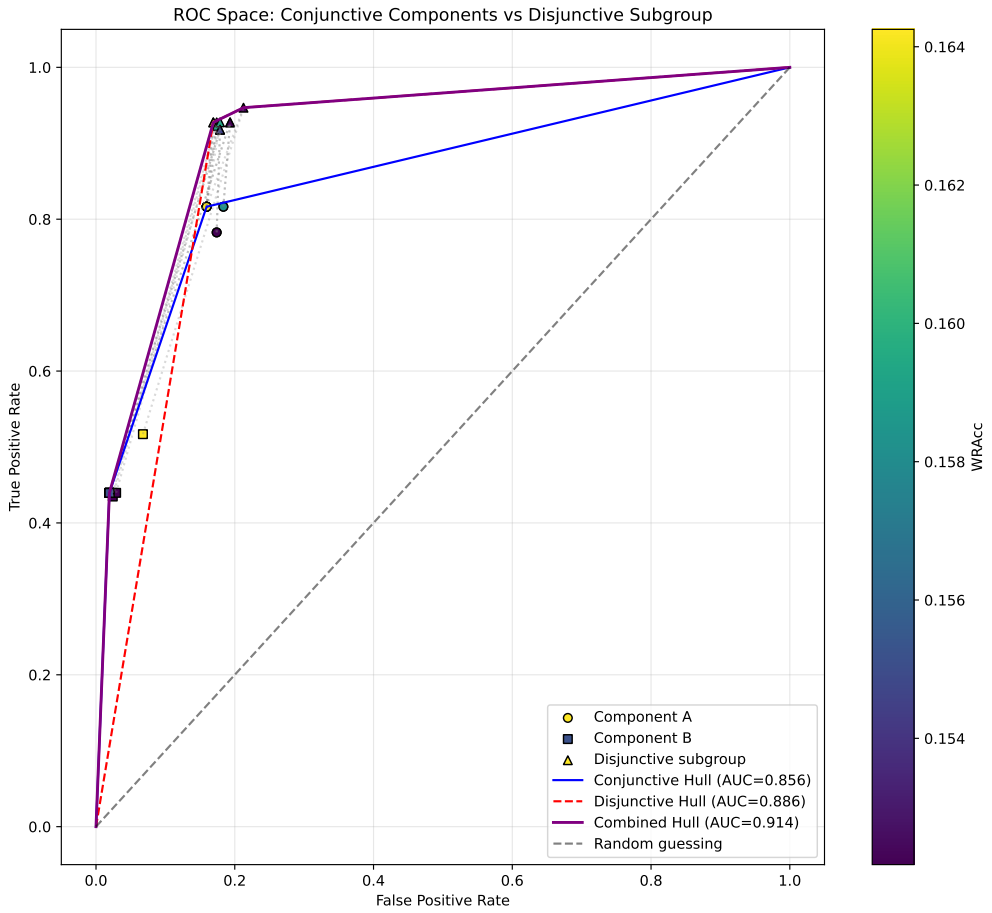

Figure 4: ROC space comparison of conjunctive components (circles, squares) and disjunctive combinations (triangles) on the Real Estate dataset (depth 3). The disjunctive hull (AUC = 0.886) outperforms the conjunctive hull (AUC = 0.856); the combined hull (AUC = 0.914) shows that mixing both types yields the best performance. Color indicates WRAcc. Gray dotted line: random guessing.

that outperform their conjunctive components in terms of WRAcc. While the improvements are modest in magnitude, they are systematic across all top-10 candidates. More importantly, the ROC analysis reveals that disjunctive and conjunctive subgroups serve complementary roles: conjunctive rules excel at capturing extremely specific, high-precision niches, while disjunctive combinations achieve higher TPR by covering broader sets of positives. The optimal set of subgroups therefore contains both types, suggesting that subgroup discovery systems should support both conjunctive and disjunctive descriptions rather than forcing a choice between them.

5.4.2 Statistical Evaluation

To determine whether the improvements observed with the pairwise combination method are statistically meaningful rather than attributable to random variation, a paired t-test was performed on the top-30 disjunctive subgroups and their better conjunctive components. The paired design is appropriate because each disjunctive subgroup $A \vee B$ is naturally paired with its better component

$\max(\text{WRAcc}(A), \text{WRAcc}(B))$. The null hypothesis is that the true mean difference between the WRAcc of the disjunction and the WRAcc of the better component is zero, indicating that combining subgroups with OR provides no systematic benefit. The alternative hypothesis is a one-sided test that the disjunctive scores are higher on average.

For statistical testing, the top-30 subgroups were compared ($k = 30$). This sample size brings the sampling distribution of the mean difference closer to a normal curve. A significance threshold of $\alpha = 0.05$ is used. The results are summarized in Table 3. The mean WRAcc of the better conjunctive components is 0.1605, while the mean WRAcc of the disjunctive combinations is 0.1837, an improvement of 0.0232. The paired t-test yields a t-statistic of 22.91 with a p-value below 0.0001, well below $\alpha = 0.05$. Therefore, the null hypothesis is rejected: the improvement is statistically significant.

Table 3: Paired t-test comparing disjunctive subgroups against their better conjunctive components on the Real Estate dataset at search depth 3. The top-30 subgroups were evaluated for each condition. The mean improvement of 0.0232 is statistically significant ($p < 0.0001$) with a large effect size (Cohen's $d = 4.18$).

Statistic	**Value**
Number of subgroup pairs	30
Mean WRAcc of better conjunctive components	0.1605
Mean WRAcc of disjunctive OR combinations	0.1837
Mean improvement	0.0232
Standard deviation of improvement	0.0055
t-statistic	22.91
p-value	< 0.0001
Significant at $\alpha = 0.05$	Yes
Cohen's d	4.18
Effect size interpretation	Large

Cohen's d was calculated as the mean improvement divided by the standard deviation of the differences, yielding $d = 4.18$. Following Cohen's conventions ($d = 0.2$ small, $d = 0.5$ medium, $d = 0.8$ large), this represents a large effect size, confirming that the improvement is not only statistically significant but also practically meaningful. The absolute improvement in WRAcc is only 0.023, which might seem small; however, this reflects that the magnitude of the gain is limited by the WRAcc scale.

In summary, the statistical evaluation provides strong evidence that the pairwise combination method produces disjunctive subgroups that systematically outperform their conjunctive components. The large effect size ($d = 4.18$) indicates that this improvement is not a marginal effect but a substantial and reliable difference. These results directly address the first research sub-question: disjunctive descriptions consistently yield higher WRAcc scores than conjunctive-only approaches on the Real Estate dataset.

5.5 Search Depth Experiment

The pairwise combination method described earlier becomes practical only when the number of conjunctive subgroups is manageable. That number, however, depends critically on the dataset

and the SubDisc parameters, in particular the search depth parameter. This parameter determines the maximum number of conditions allowed in a single conjunctive rule: depth 1 permits only single conditions (e.g., Age > 30), depth 2 permits conjunctions of two conditions (e.g., Age $> 30 \wedge$ Region = North), and deeper searches allow increasingly specific descriptions. The search depth thus controls a fundamental trade-off: deeper searches may discover more complex and potentially higher-quality subgroups, but they also dramatically expand the search space, increasing the number of candidate pairs for the OR-combination phase.

To characterize this trade-off, we vary the search depth from 1 to 10 on the Real Estate dataset. Depth 10 is chosen as the upper bound because the search space grows exponentially with depth, and depth 10 represents a practical limit beyond which the number of possible conjunctions becomes too large for exhaustive pairwise evaluation within a reasonable time. For each depth d, SubDisc is run with its standard configuration.

The following quantities are recorded for each depth. First, the runtime of the pairwise OR-combination phase is measured. This quantifies the computational cost of evaluating all candidate pairs at each depth. Second, the number of conjunctive subgroups discovered is recorded along with the number of disjunctive OR-combinations that improve upon the better of their two components. These two quantities together indicate how the size of the search space affects the discovery of beneficial disjunctive patterns. Third, the quality of the top-10 disjunctive OR-combinations is recorded, specifically the average WRAcc of these ten subgroups. Average quality is used because it reflects typical performance across multiple subgroups better than a single best value. This will identify whether deeper searches yield better subgroups and, if so, at what depth further increases no longer produce meaningful improvement. Together, these analyses inform the choice of search depth for the subsequent multi-dataset evaluation.

5.5.1 Experimental Evaluation

Figure 5 presents the results. The top-left panel shows the number of conjunctive subgroups discovered at each depth. The count increases gradually at shallow depths, then rises steeply between depths 3 and 4, after which growth slows considerably and eventually plateaus. This pattern reflects the combinatorial nature of the search space: at moderate depths, each new condition can be combined with many existing subgroups, producing a rapid expansion. Once the search reaches depths where most combinations have been explored, or where further conditions repeat the same attributes (the dataset has only six features), the number of new subgroups diminishes. The plateau after depth 5 indicates that additional depth yields few new subgroups, as the algorithm has already discovered most meaningful conjunctive patterns.

The top-right panel displays the number of disjunctive OR-combinations that improve upon the better of their two components. This count follows a similar pattern: a relatively shallow increase at low depths, followed by a dramatic jump between depths 3 and 4, then continued growth at a slightly reduced pace through depth 9, before a modest decline at depth 10. The steep rise coincides with the rapid increase in conjunctive subgroups, as each new conjunctive subgroup creates many new candidate pairs. The continued growth after the conjunctive plateau suggests that even when few new conjunctive subgroups are added, the combinations among existing subgroups continue to yield new improving disjunctions. The decline at the deepest depths occurs because very specific subgroups (with many conditions) tend to have low coverage, and unions of such subgroups rarely outperform the better component alone.

The bottom-left panel plots the runtime per depth. This runtime includes both the SubDisc search and the pairwise OR-combination phase; however, the SubDisc contribution is negligible compared to the combination phase. Total runtime remains modest at shallow depths, then increases sharply between depths 3 and 4, continues growing through depth 7, and then declines. The increase from depth 3 to 7 reflects the growing number of conjunctive subgroups and therefore candidate pairs, confirming that the computational bottleneck is the pairwise evaluation of conjunctive subgroups. The decrease after depth 7 coincides with the plateau in conjunctive subgroups. Interestingly, runtime decreases after depth 7, which may be explained by the fact that deeper subgroups are smaller and therefore faster to evaluate (computing the union of two small subgroups requires fewer instance checks than the union of two large ones).

The bottom-right panel shows the average WRAcc of the top-10 conjunctive subgroups versus the top-10 disjunctive OR-combinations. Both lines rise from depth 1 to a peak at depth 3, where the disjunctive subgroups achieve their largest advantage over the conjunctive baseline. Beyond depth 3, conjunctive quality remains stable, while disjunctive quality gradually declines, with the gap narrowing at greater depths. The decline in disjunctive quality is notable: while more disjunctive combinations exist at deeper depths (as seen in the top-right panel), they are not of higher quality. This suggests that the additional candidate pairs discovered at greater depths are largely low-quality combinations. The convergence of the two lines at deeper depths indicates that when subgroups become too specific, even their OR-combinations fail to capture meaningful patterns.

Several conclusions emerge from these observations. First, the steep increase in conjunctive subgroups between depths 3 and 4, followed by a plateau, reveals the effective search capacity of the Real Estate dataset. With only six features, the search space is fundamentally bounded; beyond depth 5, refinements cannot introduce meaningful new information. This is a practical limitation worth noting: for datasets with few features, increasing the search depth beyond the number of features yields diminishing returns, as any additional condition must repeat an already-used attribute.

Second, the peak in the count of improving disjunctive pairs occurs at a greater depth than the peak in conjunctive subgroups, but the quality of those disjunctions peaks much earlier. This decoupling between quantity and quality is informative: while deeper searches generate many more candidate disjunctions, the best ones are found at moderate depths. The additional candidates at greater depths are largely low-quality combinations that do not improve the top-10 ranking.

Lastly, the quality trends indicate that depth 3 is optimal. It achieves the highest average quality for disjunctive subgroups and maximizes the advantage over conjunctive descriptions, while keeping computational cost manageable at just a few seconds. Deeper depths offer no quality improvement, in fact, disjunctive quality declines, and come with dramatically higher runtime, increasing from seconds at depth 3 to minutes at depth 4 and beyond. Based on these results, depth 3 was selected as the setting for the experiments on the Real Estate dataset (proof-of-concept, pairwise combination, and statistical evaluation), balancing quality gains against computational cost. This trade-off analysis directly informs the multi-dataset evaluation, where depths 1 through 5 are tested. The range up to depth 5 captures the region where most meaningful growth occurs while avoiding the extreme computational costs and quality decline observed at greater depths, allowing verification of whether the patterns observed on Real Estate generalize across other domains.

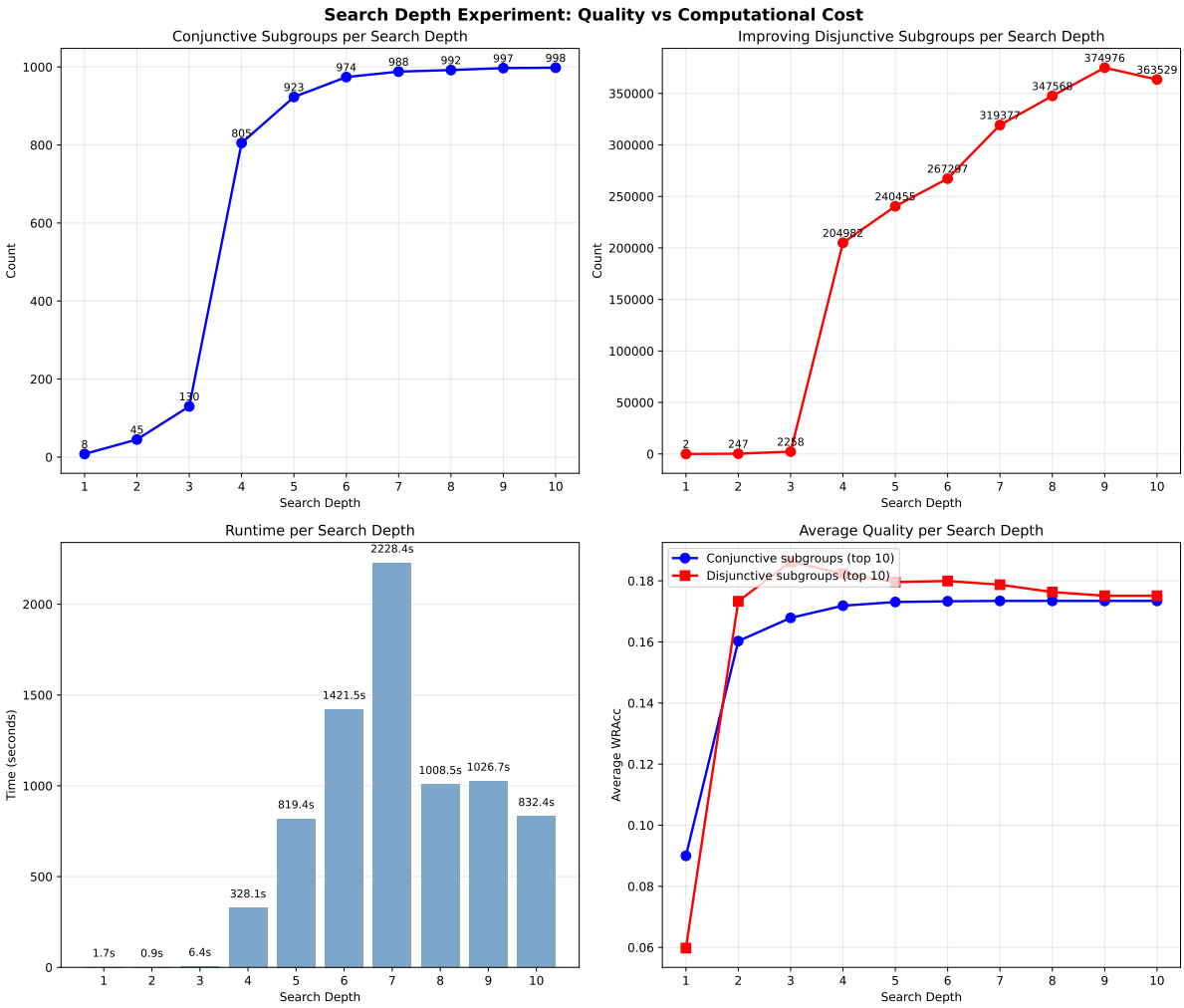

Figure 5: Search depth experiment results on the Real Estate dataset, depths 1 to 10. Each panel shows a different metric: number of conjunctive subgroups discovered (top-left), number of improving disjunctive pairs (top-right), total runtime in seconds (bottom-left), and average WRAcc of the top-10 subgroups (bottom-right).

5.6 Multi-Dataset Evaluation

The search depth experiment presented in Section 5.5 shows the trade-off between subgroup quality and computational cost on a single dataset, identifying depth 3 as a reasonable default for subsequent experiments. However, the method's performance on the Real Estate dataset may not generalize to other domains. To determine whether disjunctive subgroup discovery provides consistent benefits across different data characteristics, the evaluation is extended to the eight additional datasets described in Section 4.

For each dataset, the pairwise OR-combination method is applied following the same protocol. Based on the search depth experiment results, depths are tested from 1 to 5 rather than the full 1 to 10 range. For each depth, SubDisc is run, the complete set of conjunctive subgroups is collected, all unique pairs are evaluated, and the top-10 disjunctive OR-combinations that improve upon their

best component are retained. The same metrics as in the search depth experiment are recorded for each dataset-depth combination.

5.6.1 Experimental Evaluation

Figure 6 shows the average WRAcc improvement across all datasets at each depth, with error bars indicating the standard deviation. Depth 1 shows a small positive improvement of 0.0059. From depth 2 onward, the average improvement increases to 0.0184 at depth 2, remains stable at 0.0183 at depth 3 and 0.0181 at depth 4, then declines to 0.0157 at depth 5. The key insight is that depths 2 through 4 perform nearly identically on average, all achieving improvements around 0.018. The decline at depth 5 suggests that deeper searches may begin to overfit, producing highly specific subgroups whose unions rarely improve upon the better component. The error bars are substantial, indicating that while the average is positive, some datasets benefit much more than others. This leads to the natural next question: which datasets drive these improvements, and which do not?

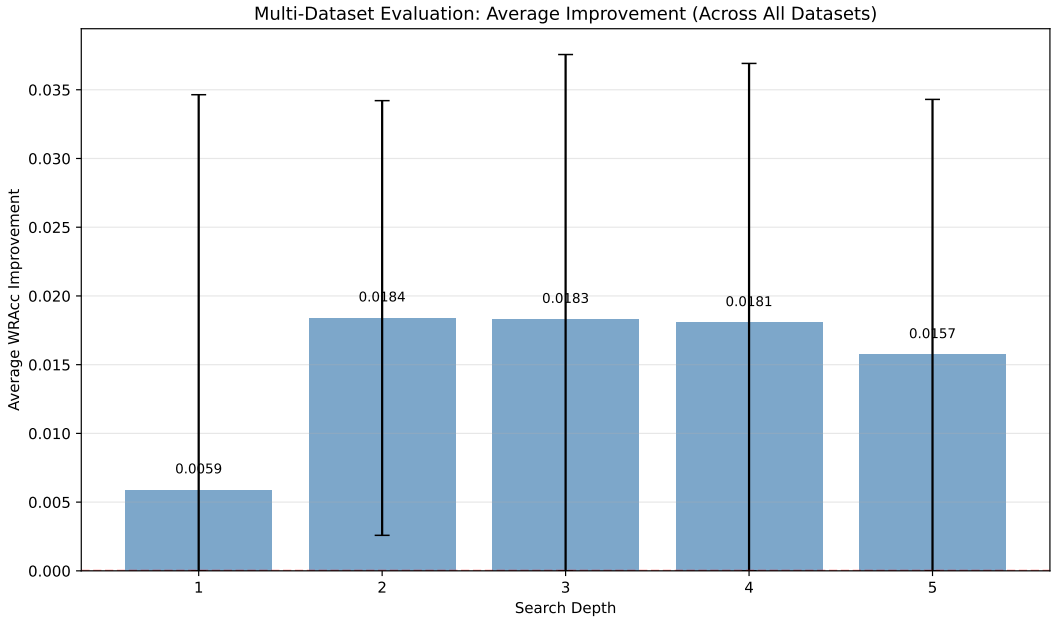

Figure 6: Average WRAcc improvement of disjunctive subgroups over their better conjunctive component across nine UCI datasets, aggregated per search depth (1 to 5). Error bars indicate standard deviation. Depths 2 through 4 achieve nearly identical average improvements (0.0184, 0.0183, and 0.0181 respectively), while depth 5 shows a decline to 0.0157. The substantial error bars indicate that some datasets benefit much more from disjunctive descriptions than others.

Figure 7 reveals why the error bars in the summary plot are so large. The heatmap shows the average improvement for each dataset individually. Concrete Strength stands out with improvements exceeding 0.06 at depths 3 through 5. This dataset contains eight correlated numerical features measuring concrete composition and age. The strong correlations between these features create many opportunities for complementary subgroups. The OR combination of such subgroups can cover different formulations of high-strength concrete, achieving higher overall quality than either subgroup alone.

Airfoil Noise also shows substantial improvements, peaking at 0.036 at depth 3. Real Estate, Student Performance, Forest Fires, Solar Flare, and Auto MPG show moderate improvements ranging from 0.013 to 0.020 at their optimal depths. Auto MPG's best improvement occurs at depth 1, but this is unreliable given the small number of subgroups at that depth; its performance at depths 2 through 4 is more representative, showing improvements around 0.007 to 0.009.

In contrast, Wine Quality and Abalone show near-zero or slightly negative improvements throughout. This is not a failure of the method but rather reflects the underlying structure of these datasets. Wine Quality has ordered but imbalanced classes: the quality score ranges from 0 to 10, but there are many more wines with average ratings than with excellent or poor ratings. This imbalance may make it difficult to find complementary subgroups that substantially improve upon individual components. Abalone predicts age from ring count, which most likely has a nearly linear relationship with features like shell measurements and weight. If the relationship is approximately linear, then the best subgroups are already captured by conjunctive rules, and OR combinations offer little additional benefit.

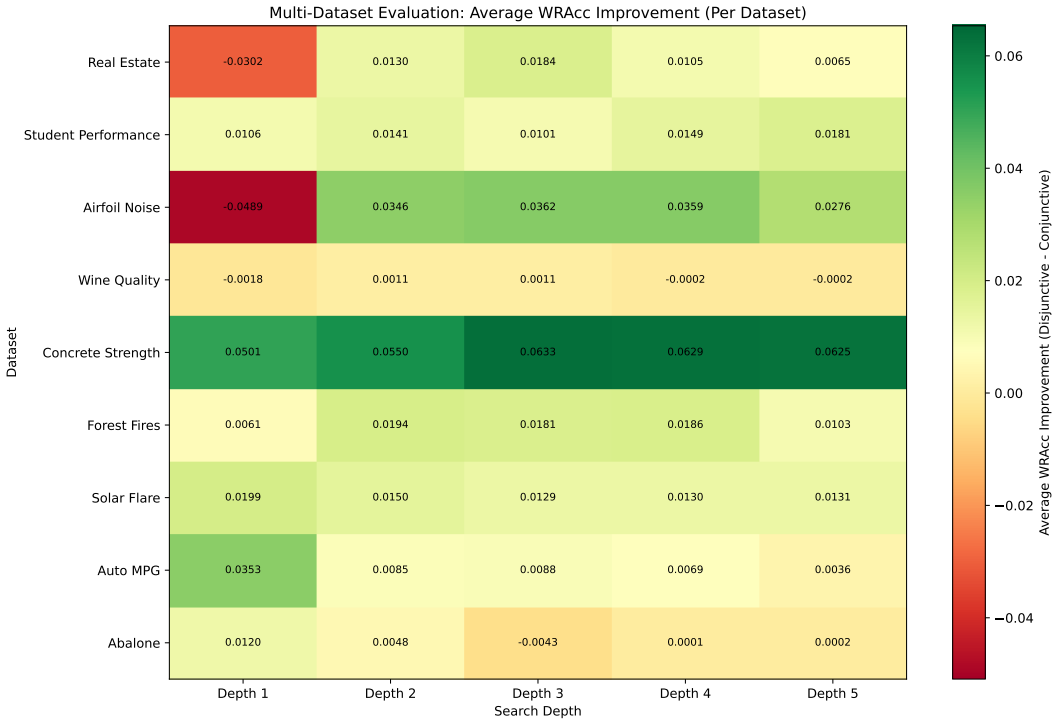

Figure 7: Average WRAcc improvement per dataset across depths 1 to 5. Concrete Strength shows the largest gains (over 0.06 at depths 3-5). Airfoil Noise shows substantial improvement, peaking at 0.036 at depth 3. Real Estate, Student Performance, Forest Fires, Solar Flare, and Auto MPG show moderate improvements ranging from 0.013 to 0.020 at their optimal depths. Auto MPG's best improvement occurs at depth 1, but its performance at depths 2 through 4 is more representative due to the small number of subgroups at depth 1. Wine Quality and Abalone show near-zero or slightly negative improvements throughout.

Figure 8 examines the computational cost of increasing search depth. Normalizing runtime to depth 2 provides a fair way to compare how different datasets scale with depth, regardless of their

absolute runtime differences due to varying instance counts and feature dimensionality. The plot shows relative runtime normalized to depth 2. A logarithmic scale is used because the runtime differences span several orders of magnitude. All datasets converge at depth 2 by construction, but their trajectories diverge substantially from that point onward. At depth 1, runtimes span several orders of magnitude below the baseline. From depth 2 onward, runtimes increase consistently for nearly all datasets.

The growth rate varies considerably across datasets. Auto MPG shows the steepest increase, with its runtime at depth 3 already exceeding 1000 times the depth-2 baseline, and continuing to grow through depth 5. Concrete Strength and Real Estate also show large increases, reaching approximately 1000 times the depth-2 baseline at depth 5. These datasets all contain multiple correlated numerical features (Concrete Strength has eight measurements of composition and age; Real Estate has latitude, longitude, house age, and distance to MRT; Auto MPG has displacement, weight, and acceleration). At deeper search depths, the algorithm generates many conjunctive subgroups involving different combinations of these correlated features, leading to a combinatorial explosion in the number of candidate pairs. Solar Flare exhibits a similarly large jump between depths 3 and 4. This dataset has ten categorical features with small domains (e.g., spot distribution has 4 possible values, evolution has 3 possible values). As search depth increases, the number of possible value combinations grows rapidly, producing many conjunctive subgroups and hence many candidate pairs.

In contrast, Wine Quality and Abalone show comparatively modest growth, remaining below 10 times the depth-2 baseline even at depth 5. This is consistent with their relatively flat improvement profiles in the heatmap: fewer high-quality subgroups are discovered, so the candidate pair space remains small. Forest Fires shows the lowest growth of all datasets, staying near the baseline even at depth 5, which aligns with its moderate improvement profile and suggests that the dataset's structure yields relatively few conjunctive subgroups. Student Performance and Airfoil Noise show intermediate growth rates, neither as extreme as Auto MPG nor as modest as Wine Quality. Airfoil Noise has only five features, which limits the number of possible subgroup combinations, while Student Performance contains many categorical features whose values are spread sparsely, limiting the number of high-coverage subgroups.

The absolute runtime values from Table 4 provide context for interpreting the relative increases. The average trend line (black dashed) shows that depth 3 is already roughly 500 times slower than depth 2 on average, and depths 4 and 5 are around 1000 times slower. Translating these relative increases into absolute terms: at depth 2, runtimes range from fractions of a second to several minutes; a 500-fold increase means that datasets taking seconds at depth 2 can take tens of minutes at depth 3, and those taking minutes can extend to hours. Given that the quality improvements plateau between depths 2 and 4, the enormous computational overhead incurred beyond depth 3 is difficult to justify. The fact that quality actually declines at depth 5 while cost continues to rise confirms that depth 3 represents the point of diminishing returns: beyond this depth, additional computational investment yields no quality improvement and, in fact, leads to worse results. Depth 3 therefore remains the recommended default: it achieves quality nearly identical to depth 2, keeps computational cost manageable, and avoids the extreme runtimes that make depths 4 and 5 impractical for larger datasets.

In summary, the multi-dataset evaluation confirms that the pairwise combination method generalizes across diverse datasets, producing consistent positive improvements. Concrete Strength benefits most from disjunctive descriptions, with improvements exceeding 0.06, likely due to

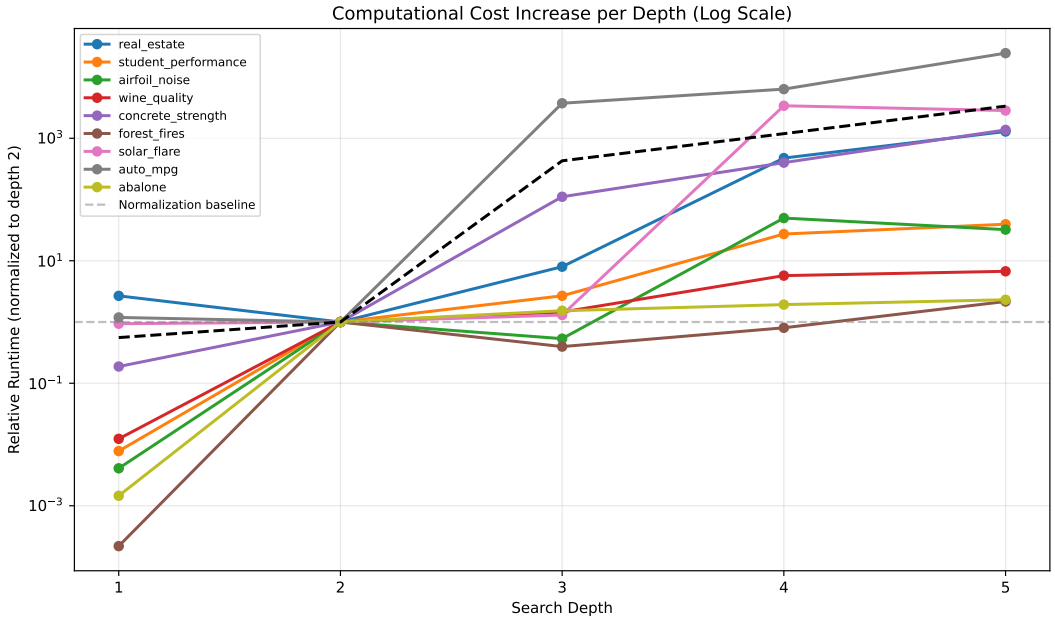

Figure 8: Relative computational cost per depth, normalized to depth 2 (log scale). Runtimes increase consistently from depth 2 onward, but growth rates vary substantially. Real Estate, Concrete Strength, Solar Flare, and Auto MPG exceed 1000 times the depth-2 baseline at depth 5, while Wine Quality and Abalone remain below 10 times. The black dashed line shows the average across all datasets. Depth 3 is roughly 500 times slower than depth 2 on average; depths 4 and 5 exceed 1000 times slower.

its many correlated numerical features that create opportunities for complementary subgroups. Real Estate, Student Performance, Airfoil Noise, Forest Fires, Solar Flare, and Auto MPG show moderate improvements ranging from 0.013 to 0.020, demonstrating that the method is broadly effective across different domains and data types. Wine Quality and Abalone show near-zero improvement, indicating that disjunctive descriptions are not universally beneficial; datasets with imbalanced ordered classes or approximately linear relationships leave little room for improvement over conjunctive rules. The computational cost analysis reveals that depths 2 through 4 achieve nearly identical average improvements, but runtime increases dramatically with depth. This confirms that depth 3 represents the point of diminishing returns for practical applications.

5.7 Theoretical Foundation for Pruning OR Combinations

The multi-dataset evaluation revealed that computational cost grows dramatically with search depth, with depth 3 already 500 times slower than depth 2 on average and depths 4 and 5 exceeding 1000 times slower. While the pairwise combination method performs well on modestly sized datasets, scaling to larger datasets or deeper search depths requires more efficient exploration of the candidate pair space. This subsection derives a theoretical upper bound on the quality of OR combinations based on the linear formulation of WRAcc. This bound could form the basis of a pruning strategy to reduce the number of candidate pairs that need to be evaluated, opening the door to more scalable implementations of disjunctive subgroup discovery.

Table 4: Absolute runtimes at depth 2 for each dataset. Discovery time: SubDisc subgroup search; Combination time: pairwise OR evaluation; Total: combined runtime in seconds.

Dataset	**Discovery (s)**	**Combination (s)**	**Total (s)**
Real Estate	0.13	0.34	0.47
Student Performance	7.50	85.91	93.42
Airfoil Noise	0.17	15.92	16.09
Wine Quality	49.06	84.78	133.84
Concrete Strength	0.29	1.23	1.52
Forest Fires	28.76	65.41	94.17
Solar Flare	0.10	0.14	0.24
Auto MPG	0.04	0.07	0.11
Abalone	307.08	0.47	307.56

5.7.1 Derivation of the Upper Bound

The bound follows from the simplified formulation of WRAcc introduced in Section 2. Recall that:

$$\text{WRAcc}(S) = \frac{1}{|D|} \times \big(\text{pos}_S - |S| \times p(H)\big) \tag{9}$$

where pos_S is the number of positive instances in subgroup S, $|S|$ is the subgroup size, $|D|$ is the dataset size, and $p(H)$ is the global positive rate. This formulation expresses WRAcc as a linear function of two quantities: the number of positive instances in the subgroup and the total subgroup size. Both $\frac{1}{|D|}$ and $p(H)$ are constants for a given dataset. This linearity enables algebraic manipulation of WRAcc expressions for combined subgroups.

For two subgroups A and B, the number of positive instances in their union equals the sum of positives in each subgroup minus the positives in their intersection:

$$\text{pos}_{A\cup B} = \text{pos}_A + \text{pos}_B - \text{pos}_{A\cap B} \tag{10}$$

Similarly, the size of the union is:

$$|A \cup B| = |A| + |B| - |A \cap B| \tag{11}$$

Substituting these into the linear formulation of WRAcc yields an expression for the quality of the OR combination:

$$\text{WRAcc}(A \cup B) = \frac{1}{|D|}\left[(\text{pos}_A + \text{pos}_B - \text{pos}_{A\cap B}) - (|A| + |B| - |A \cap B|) \cdot p(H)\right] \tag{12}$$

Rearranging terms separates the contributions of A and B from the contribution of their intersection:

$$\text{WRAcc}(A \cup B) = \frac{1}{|D|}\left[(\text{pos}_A - |A| \cdot p(H)) + (\text{pos}_B - |B| \cdot p(H)) - (\text{pos}_{A\cap B} - |A \cap B| \cdot p(H))\right] \tag{13}$$

The first two terms are precisely WRAcc(A) and WRAcc(B). Substituting yields:

$$\text{WRAcc}(A \cup B) = \text{WRAcc}(A) + \text{WRAcc}(B) - \frac{1}{|D|}\left(\text{pos}_{A\cap B} - |A \cap B| \cdot p(H)\right) \tag{14}$$

The subtracted term involves the intersection of A and B. The quantity $\text{pos}_{A\cap B} - |A \cap B| \cdot p(H)$ is the difference between the observed number of positive instances in the intersection and the number expected under the global distribution. It measures whether the intersection has a higher positive rate than the global average. If this quantity is positive (the intersection is enriched), then we are subtracting a positive number from $\text{WRAcc}(A) + \text{WRAcc}(B)$:

$$\text{WRAcc}(A \cup B) = \text{WRAcc}(A) + \text{WRAcc}(B) - \frac{1}{|D|}\Big(\underbrace{\text{pos}_{A\cap B} - |A \cap B| \cdot p(H)}_{\geq 0}\Big) \tag{15}$$

Consequently, the actual value of $\text{WRAcc}(A \cup B)$ is less than or equal to the sum:

$$\text{WRAcc}(A \cup B) \leq \text{WRAcc}(A) + \text{WRAcc}(B) \tag{16}$$

If the quantity is negative (the intersection is depleted), then a negative number is subtracted, which is equivalent to adding a positive number. In that case, the inequality reverses, and the OR combination could exceed the sum.

In practice, for pairs of subgroups that are both enriched (positive WRAcc), the intersection could theoretically be depleted. But if the intersection has a lower positive rate than the global average, including it in the union would only dilute the quality of the OR combination. In that case, one could simply omit the overlapping instances by taking the disjoint union instead. Since the OR operator does not force inclusion of the intersection, we only need to consider pairs where $\text{pos}_{A\cap B} \geq p(H) \cdot |A \cap B|$ for the purpose of finding improving disjunctive subgroups. Pairs with depleted intersections would not yield better OR combinations than simply taking the disjoint union, so they can be safely ignored.

Based on the reasoning above, the positive rate of the intersection is assumed to be at least the global positive rate. Under this assumption, the inequality $\text{WRAcc}(A \cup B) \leq \text{WRAcc}(A) + \text{WRAcc}(B)$ holds, meaning that the quality of an OR combination can never exceed the sum of the qualities of its parts. This inequality forms the foundation of the pruning strategy, and the following section describes how it is applied during the search.

5.7.2 Proposed Pruning Approach

The bound $\text{WRAcc}(A \cup B) \leq \text{WRAcc}(A) + \text{WRAcc}(B)$ could be used to prune the search space without sacrificing quality. A hypothetical implementation would proceed as follows. The conjunctive subgroups would first be sorted in descending order by their WRAcc scores: $\text{WRAcc}(S_1) \geq \text{WRAcc}(S_2) \geq \cdots \geq \text{WRAcc}(S_m)$. The algorithm would maintain the best quality found among all evaluated OR combinations, initially set to the best single conjunctive subgroup quality.

The search would then proceed with an outer loop over each subgroup S_i and an inner loop over subsequent subgroups S_j (with $j > i$). Before evaluating any pair, two pruning checks could be applied. The first pruning check occurs in the outer loop. For a subgroup S_i with quality q_i, if $2 \cdot q_i$ is less than or equal to the current best quality, then for any subgroup S_j with quality $q_j \leq q_i$, the sum $q_i + q_j$ is at most $2 \cdot q_i$ and therefore cannot exceed the current best. Since the list is sorted

in descending order, all remaining subgroups have quality less than or equal to q_i. Hence, no pair involving S_i or any subsequent subgroup could improve upon the current best. The outer loop would break, terminating the search.

The second pruning check occurs in the inner loop. For a fixed S_i with quality q_i, the inner loop would iterate over S_j in descending order of quality. As long as $q_i + q_j$ exceeds the current best, the pair (S_i, S_j) remains a candidate for evaluation. Once a subgroup S_j is encountered where $q_i + q_j$ is less than the current best, all subsequent subgroups have equal or lower quality, so the sum would also be at most the current best. The inner loop would break at this point, skipping all remaining pairs with the same S_i. Only when a pair (S_i, S_j) passes both pruning checks would the actual union be evaluated. The disjunctive subgroup would be constructed as $S_i \cup S_j$, and its WRAcc computed. If the resulting WRAcc exceeds the current best, the best is updated. The improvement condition $\text{WRAcc}(S_i \cup S_j) > \max(\text{WRAcc}(S_i), \text{WRAcc}(S_j))$ would then determine whether the pair should be retained in the final result set.

The pruning strategy requires that the conjunctive subgroups be sorted by their WRAcc values in descending order. This sorting step has complexity $O(m \log m)$, where m is the number of subgroups. After sorting, the loops apply the pruning checks. Without pruning, evaluating all pairs would require checking each of the $O(m^2)$ pairs, each evaluation scanning all n instances to compute the union, resulting in $O(m^2 \cdot n)$ operations. With pruning, many pairs would be skipped based on the sum of their WRAcc values, requiring only one addition and one comparison per skipped pair.

The actual number of pairs evaluated would depend on the distribution of WRAcc values. For example, if most subgroups have very low WRAcc values, the sum of two such values would rarely exceed the current best, so many pairs would be pruned early. Conversely, if many subgroups have high WRAcc values close to the best found so far, more pairs would satisfy the condition and require full evaluation. As the search progresses and better disjunctive subgroups are discovered, the current best increases, making the pruning condition $q_i + q_j >$ current best stricter. Consequently, fewer pairs would satisfy the condition and fewer pairs would require full evaluation. While this pruning approach has not been implemented in the current thesis, it provides a clear direction for future work to scale the pairwise combination method to larger datasets or deeper search depths.

6 Discussion and Conclusion

This thesis investigated three central questions: whether disjunctive queries improve subgroup quality, what trade-offs between quality and complexity arise, and how the search space can be explored efficiently. The experimental results provide clear answers to each of these questions, while also revealing important limitations and directions for future research.

6.1 Summary of Findings

The case study confirmed that disjunctive patterns exist in real data: an L-shaped region on the Real Estate dataset could not be captured by any single conjunctive rule without including a low-quality region. By excluding this region, a disjunctive combination achieved substantially higher WRAcc.

The proof-of-concept experiment showed that OR-expansion is a safe direction for further research: adding a single condition improved six out of ten subgroups, with an average gain of 0.005.

The pairwise combination method consistently improved all top-10 subgroups on the Real Estate dataset, yielding an average WRAcc gain of 0.023. This improvement represents a large effect size, with Cohen's d = 4.18. ROC analysis revealed that disjunctive and conjunctive subgroups serve complementary roles: conjunctive rules excel at high-precision niches, while disjunctive combinations achieve higher true positive rates. The combined convex hull achieved the highest AUC, suggesting that subgroup discovery systems should support both representation types.

The search depth experiment identified depth 3 as optimal: disjunctive quality peaked at this depth, while runtime increased dramatically beyond it. Deeper searches generated more candidate disjunctions, but not of higher quality.

The multi-dataset evaluation showed that the pairwise combination method generalizes across diverse domains, though the magnitude of improvement varied substantially by dataset. Concrete Strength showed the largest gains, likely due to its correlated numerical features. Real Estate, Student Performance, Airfoil Noise, Forest Fires, Solar Flare, and Auto MPG showed moderate improvements. Wine Quality and Abalone showed near-zero improvement, indicating that disjunctive descriptions are not universally beneficial; datasets with imbalanced ordered classes or approximately linear relationships leave little room for improvement over conjunctive rules.

Finally, the computational cost analysis revealed a substantial trade-off: depth 3 was roughly 500 times slower than depth 2 on average, while depths 4 and 5 exceeded 1000 times slower. Given that quality did not improve beyond depth 3, this depth represents the point of diminishing returns.

6.2 Relation to Existing Work

The Introduction identified a gap in the literature: while SDIGA and DISGROU explored disjunctive representations, neither provided a clean assessment of the general OR operator. This thesis fills that gap by investigating the OR operator within a standard crisp subgroup discovery framework, allowing disjunctions across arbitrary feature combinations. Unlike SDIGA, which couples disjunction with fuzzy logic, this work isolates the effect of the OR operator by operating with crisp set membership. Unlike DISGROU, which restricts disjunction to individual numerical attributes, this work allows

disjunctions across arbitrary combinations of features, enabling patterns like the L-shaped region that require different features in each disjunct.

The experimental results confirm that the general OR operator is a valuable addition to subgroup discovery. The consistent improvements across diverse datasets, particularly those with correlated numerical features, suggest that disjunctive descriptions are broadly useful rather than limited to specific niche cases. The complementary roles of conjunctive and disjunctive subgroups observed in the ROC analysis demonstrate that neither representation type dominates the other; instead, each excels in different regions of the performance space. Conjunctive rules achieve very high precision with low false positive rates, while disjunctive combinations can capture substantially more true positives.

6.3 Limitations

Several limitations of this work should be acknowledged. First, the evaluation was restricted to binary classification tasks. While this is a common setting for subgroup discovery, the findings may not generalize to multi-class or numerical target variables. Second, to keep computation feasible, the experiments were limited to modestly sized UCI datasets. The conclusions might differ on truly large-scale datasets, where the combinatorial growth of candidate pairs could become prohibitive. Third, the analysis focused on the top-10 and top-30 subgroups; whether disjunctive descriptions offer benefits beyond these rankings remains unknown. Fourth, the pairwise combination method considers only unions of two conjunctive subgroups. Allowing unions of three or more components might yield additional improvements, but also would dramatically expand the search space. Finally, the pruning strategy derived in Section 5.4 was not fully implemented in this thesis. While the theoretical upper bound is proven, its practical effectiveness in reducing runtime without sacrificing quality remains untested.

6.4 Future Work

The limitations outlined above suggest several promising directions for future research. Implementing the pruning strategy is the most immediate next step. The upper bound $\text{WRAcc}(A \cup B) \leq \text{WRAcc}(A) + \text{WRAcc}(B)$ can be used to skip pairs that cannot improve upon the current best, potentially reducing the number of evaluated pairs from $O(s^2)$ to $O(s \log s)$ or better, depending on the distribution of WRAcc values. A pruning experiment should be conducted to quantify the actual savings and verify that no high-quality disjunctive combinations are lost.

Beyond pruning, several algorithmic improvements could be explored. The current pairwise combination method considers all unique pairs of conjunctive subgroups, but many of these pairs are unlikely to yield improving disjunctions. A more sophisticated candidate generation strategy could focus on subgroups that cover complementary regions of the instance space. For instance, one could cluster subgroups based on which instances they cover; pairs from different clusters are more likely to cover different parts of the positive space and thus yield beneficial unions. Another direction is to avoid the two-step process entirely: instead of first generating conjunctive subgroups and then pairing them, genetic algorithms or beam search could be adapted to directly search the space of disjunctive rules. This would allow the search to explore combinations of three or more components and to refine disjunctions incrementally, rather than being limited to unions of pre-discovered conjunctive rules.

The multi-dataset results also raise interesting questions for future investigation. Why does Concrete Strength benefit so much more from disjunctive descriptions than other datasets? Is it solely due to the correlation structure of its features, or are there other factors at play? A systematic analysis of dataset characteristics that predict the effectiveness of disjunctive descriptions would help practitioners decide when to apply the method. Similarly, the near-zero improvements on Wine Quality and Abalone deserve further scrutiny. Could different quality measures, alternative target binarization strategies, or deeper search depths reveal benefits that were not captured by the current experimental protocol?

Finally, the observation that disjunctive and conjunctive subgroups serve complementary roles points toward a broader research direction: rather than treating disjunctive queries as a replacement for conjunctive ones, future subgroup discovery systems could integrate both representation types. The ROC analysis showed that conjunctive rules excel at high-precision niches while disjunctive combinations achieve higher true positive rates, and the combined convex hull outperformed either type alone. This suggests that an optimal subgroup discovery system would output a mixed set of subgroups, automatically selecting the most appropriate representation for each pattern based on quality and interpretability criteria.

6.5 Concluding Remarks

This thesis has demonstrated that disjunctive queries are a valuable addition to subgroup discovery. The inclusion of OR operators expands the hypothesis space from axis-aligned hyper-rectangles to unions of rectangles, enabling the discovery of patterns that purely conjunctive search systematically misses. The L-shaped region in the Real Estate dataset provides a concrete example of such a pattern. The pairwise combination method consistently produces disjunctive subgroups that outperform their conjunctive components, with statistically significant improvements and large effect sizes. The multi-dataset evaluation confirms that these findings generalize across diverse domains, though with variation depending on dataset characteristics.

The computational cost of exhaustive pairwise evaluation is substantial, but the trade-off can be managed by restricting search depth to moderate values. Depth 3 strikes a balance between quality and cost: it achieves quality nearly identical to depth 2, keeps computational cost manageable, and avoids the extreme runtimes that make depths 4 and 5 impractical for larger datasets or those with many correlated features. The theoretical upper bound derived in this thesis opens the door to more efficient search strategies, though implementing this pruning strategy remains future work.

Returning to the opening question of this thesis: what hidden patterns remain invisible simply because the tools used to find them are not expressive enough? The evidence presented here suggests that many such patterns exist, and that disjunctive queries can bring them to light. By supporting disjunctive queries across arbitrary features, subgroup discovery becomes more expressive without sacrificing interpretability: the resulting rules are still simple logical combinations of conditions, just with an OR connecting them rather than only AND. This expressiveness comes with computational costs due to the expanded hypothesis space. However, these costs can be managed in practice by restricting search depth to moderate values (e.g., depth 3) and by setting a minimum coverage threshold to discard very small subgroups early. Future work on pruning strategies based on the theoretical upper bound could reduce costs further. Subgroup discovery systems should support both conjunctive and disjunctive descriptions, recognizing that the two representation types serve complementary roles in capturing the rich structure of real-world data.

References

[1] Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, Hillsdale, NJ, 2nd edition.

[2] MJ del Jesus, P. Gonzalez, F. Herrera, and M. Mesonero. Evolutionary fuzzy rule induction process for subgroup discovery: A case study in marketing. *IEEE transactions on fuzzy systems*, 15(4):578–592, 2007.

[3] Reynald Eugenie and Erick Stattner. Disgrou: an algorithm for discontinuous subgroup discovery. *PeerJ. Computer science*, 7:e512–, 2021.

[4] Dragan Gamberger, Nada Lavrač, Filip Železný, and Jakub Tolar. Induction of comprehensible models for gene expression datasets by subgroup discovery methodology. *Journal of biomedical informatics*, 37(4):269–284, 2004.

[5] Jean-Christophe Goulet-Pelletier and Denis Cousineau. A review of effect sizes and their confidence intervals, part i: The cohen's d family. *The Quantitative Methods for Psychology*, 14(4):242–265, 2018.

[6] Sumyea Helal. Subgroup discovery algorithms a survey and empirical evaluation. *Journal of computer science and technology*, 31(3):561–576, 2016.

[7] Franciso Herrera, Cristóbal José Carmona, Pedro González, and María José del Jesus. An overview on subgroup discovery: foundations and applications. *Knowledge and information systems*, 29(3):495–525, 2011.

[8] Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. The UCI machine learning repository.

[9] Willi Klösgen. Explora: A multipattern and multistrategy discovery assistant. In *Advances in Knowledge Discovery and Data Mining*, 1996.

[10] Arno Knobbe, Marvin Meeng, Wouter Duivesteijn, Matthijs van Leeuwen, Rob Konijn, Michael Mampaey, Siegfried Nijssen, Iyad Batal, and Jeremie Gobeil. Subdisc: Subgroup discovery, 2021.

[11] Daniel Lambach and Dragan Gamberger. Temporal analysis of political instability through descriptive subgroup discovery. *Conflict management and peace science*, 25(1):19–32, 2008.

[12] Nada Lavrač, Peter Flach, Blaz Zupan, Peter A Flach, J van Leeuwen, and Saso Dzeroski. Rule evaluation measures: A unifying view. In *Inductive Logic Programming*, volume 1634 of *Lecture Notes in Computer Science*, pages 174–185. Springer Berlin / Heidelberg, Germany, 1999.

[13] Willem Jan Palenstijn and Arno Knobbe. pySubDisc: Python wrapper for SubDisc, 2025.

[14] Stefan Wrobel, Jan Komorowski, and Jan Zytkow. An algorithm for multi-relational discovery of subgroups. In *Lecture notes in computer science*, pages 78–87, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

A Code Availability

The complete source code for all experiments in this thesis is available on GitHub:
https://github.com/MaritPaul/bachelor-thesis-disjunctive-subgroup-discovery