**Annex D. Comparison of main multi-label classification methods**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **Strength** | **Weeknesses** | **Handling imbalaced data** | **Handling label correlation** | **Best for** |
| **Bayesian Networks** | - Probabilistic model; - Handles label dependencies; - Uncertainty handling; | - Computationally expensive; - Requires domain knowledge; | - Handles uncertainty well but can struggle with **imbalanced labels** in large datasets without proper priors; | - **Captures label dependencies** effectively, ideal for correlated labels; | - Complex, correlated labels; - Uncertain data; |
| **Neural Networks** | - Scalable; - Flexible; - Good for complex data ; - Automatic feature extraction; | - Requires large data; - Computa-tionally expensive; - Black-box model; | - Can perform poorly on **imbalanced data** unless combined with techniques like **class weighting, sampling**, or **loss function adjustments;** | - Can handle **label correlation** well if **multi-output neural networks** or **embedding layers** are used, but computationally expensive; | - Large datasets; - Complex relationships; - High-dimensional data; |
| **One-vs-Rest** | - Easy to implement; - Works well for independent labels; | - Ignores label dependencies; - Not ideal for correlated labels; | - Struggles with highly **imbalanced labels** as classifiers for rare labels may not be trained properly; | - **Ignores label correlations**, handles independent labels better; | - Independent labels; - Simpler tasks; |
| **One-vs-One** | - Treats each pair of labels as a classification problem; - Can be more accurate than OvR for some problems; | - Requires a large number of classifiers; - Computatio-nally expensive; | - Can perform better with **imbalanced labels** than OvR because it focuses on **pairwise comparisons**, but still sensitive to imbalance; | - **Ignores label correlations** and treats each label pair independently; | - Tasks with relatively independent labels; - Can work well with smaller label sets; |
| **Classifier Chains** | - Captures label dependencies; - Flexible with base classifiers; | - Sensitive to the order of classifiers; - Not scalable with many labels; | - Handles **imbalanced data better** as each classifier can focus on specific label correlations, but performance can degrade with highly imbalanced data; | - **Captures label dependencies** effectively, works well when labels are correlated; | - Moderate correlation between labels; - Sequential dependencies; |
| **Binary Relevance** | - Simple; - Efficient for independent labels; | - Ignores label dependencies; - Poor performance with correlated labels; | - Struggles with **imbalanced labels**, as each label is treated independently, potentially ignoring the imbalance of rare labels; | - **Ignores label correlations**, works well for independent labels; | - Independent labels; - Smaller datasets; |
| **k-NN for Multi-Label** | - Simple; - Interpretable; - Good for small data; | - Computa-ionally expensive for large datasets; - Struggles with high-dimensionality; | - Can be sensitive to **imbalanced data** due to the nearest neighbors often favoring the majority class unless specific adjustments are made; | - Can handle **label correlation** in a **local context** (based on neighbors), but not explicitly designed for label dependencies; | - Small datasets; - Low-dimensional spaces; |
| **Label Powerset** | - Treats multi-label problem as multi-class by considering every unique label combination as a class; - Can capture label dependencies; | - Can result in a large number of classes; - Doesn’t scale well with many labels or combinations; | - Can struggle with **imbalanced data** as the number of classes may be unbalanced, causing rare combinations to have very few samples; | - **Captures label dependencies** by treating unique label combinations as separate classes, but can be computationally inefficient with many labels; | - Tasks with a smaller number of possible label combinations; - Where label dependencies are crucial; |
| **Label Ranking** | - Suitable for problems where the order or ranking of labels matters; - Models the rank of labels rather than just the presence/absence; | - Assumes a ranking relationship, which may not always hold; - Not ideal for cases where label dependencies are weak; | - Can handle **imbalanced labels** better than binary classification as it ranks labels based on their importance, but rare labels might still be underrepresented in training; | - **Can capture label dependencies** to some extent in ranking but may not be explicitly designed for correlated labels in the traditional sense; | - When labels have an inherent ranking order ; |