Machine Learning - Exercise 1: Classification

Group 31 - TU Wien

Bianca Apostolescu, Lu Chen, Matthias Glinzner

1 Datasets

The first dataset that we chose is called *"Crimes in Los Angeles from 2010"*. It is a relatively large dataset, with approximately 1.7 million entries and both numerical and categorical features, summing up to 26 total features. Moreover, this dataset has 9.3 million missing values (approximately *21%* of the dataset is missing), making it a challenge to be analysed. The dataset represents a transcription from original crime reports being a government-issued dataset, thus having accurate data and allowing for interesting preprocessing opportunities and classification tasks.

The second dataset that was chosen is called *"College US News Dataset"*. Compared to the first one, this dataset is relatively small, having only 1302 entries and 34 features. There are two categorical features, the rest being numerical and having a various number of missing values that sum up to 7830 (approximately *17.7%* of the dataset is missing). This dataset is a smaller version of another dataset that binaries one of the categorical features in order to be used as a target for binary classification.

The third dataset obtained from Kaggle is *Loan Application Dataset*, encompassing applications filed between 2012 and 2018. The dataset contains over 100,000 records, including pivotal variables such as applicant demographics, loan amounts requested, employment history, and loan repayment statuses.

The fourth and final data set was obtained from Kaggle as well. It contains 285 samples of *Breast Cancer Diagnostics* with 30 features each.

2 Classifiers

2.1 Crimes in Los Angeles from 2010 - KNN, Random Forest, and MLP

The characteristics of this dataset are the really high dimensionality, the extreme non-linearity, and the class imbalance. Given that it is a large dataset with uncorrelated features, the classification task proved to be difficult.

- **High Dimensionality:** The tree-based and neural networks algorithms are known to be suitable for high dimensional data, therefore the chosen algorithms are Random Forest and MLP.
- **Class Imbalance:** This is a problem that when left untreated can greatly impact the final results. However, the tree-based methods are suitable for this due to their result aggregation.
- **Non-Linearity:** After analysing the data and performing the preprocessing techniques, we noticed that the features have low correlation scores. Algorithms that can tackle this problem are Radnom Forest, MLP, and to some extent the KNN method when fine-tuning the neighbors.

2.2 Colleges US news - KNN, Random Forest, and MLP

Considering the relatively small dataset, with straight-forward features and only one categorical feature, the *State* column of the dataset, multiple classifiers were tested in order to which performed better on the pre-processed dataset. Although the dataset is small, it has a lot of outliers that can greatly influence the final results. Thus, some of the chosen classifiers perform better in the presence of outliers than others, such as the tree-based algorithms. On the other side, KNN is known to be easily influenced by the presence of outliers.

Moreover, the Random Forest and MLP classifiers can be used with mixed data types (both categorical and numerical data), compared to KNN which only accepts numerical data.

- **Outliers:** The tree-based or ensemble algorithms, such as Random Forest and Adaboost are known to handle outliers better than less complex methods, like KNN, especially when the number of neighbors used for the KNN algorithm is low. Their approach is to build multiple decision trees and aggregate the results to get to best possible predictions. Moreover, an SVM algorithm was tested as it is known to have good results when dealing with outliers, although it is mostly used when working with images due to their linearity, not text data.
- **Linearity:** After analysing the data, the conclusion was that the features are correlated, thus the chosen models needed to be suitable for this kind of data. Therefore, the SVM and the MLP algorithms were chosen.

- **High Dimensionality:** Although the number of entries in the dataset is relatively low, there are 32 features used for the training set, thus being a high-dimensional dataset. Classifiers known to perform better when dealing with high-dimensional data are tree-based models and neural networks, such as Random Forest and MLP.

2.3 Breast Cancer - K-NN, Random Forest and MLP

As with the second data set, this one is also relatively small, consisting of only 285 entries. It also only has one categorical feature and no missing values.

The data set is of high dimensionality, making Random Forest and MLP robust choices; To account for outliers and also to make the results comparable between data sets, K-NN was chosen as the third classifier.

2.4 Loan Application - LightGBM, XGBoost, and MLP

This project faced the intricate task of analyzing a dataset with high dimensionality, potential class imbalance, and multifaceted feature interactions. To navigate these challenges, we strategically selected LightGBM, XGBoost, and MLP classifiers.

- **High Dimensionality**: LightGBM and XGBoost, are known for their proficiency in handling high-dimensional data. Their sophisticated tree-building algorithms are particularly adept at processing extensive features efficiently, minimizing performance trade-offs.

- **Class Imbalance**: Another hurdle was the potential class imbalance, a frequent issue in loan datasets where some loan grades are less prevalent. LightGBM emerged as a suitable solution, thanks to its inherent ability to handle categorical features effectively and its enhanced focus on minority classes during training.

- **Complex Interactions and Non-Linearity**: Lastly, given the diverse range of features, we anticipated intricate interactions and nonlinear relationships within the data. MLP, with its deep learning capabilities, was an ideal choice for modeling these complex relationships. Its layered structure and nonlinear processing power enable it to capture and interpret the nuanced inter-dependencies and patterns that traditional linear models might overlook.

3 Data Exploration and Preprocessing

3.1 Crimes in Los Angeles from 2010

As mentioned in Section 1, this is a very large dataset with 26 features and 21% missing values that needed to be dealt with in order to perform the classification task. Therefore, multiple pre-processing methods needed to be implemented in order to prepare the dataset:

- **Feature and Data Reduction:** After an initial correlation analysis was performed, uncorrelated features were removed, such as 'Adress' and the redundant ID codes for other features. Also, the dataset contains information from 2010 until 2018. However, the 2018 year has very little data, thus it was completely removed from the dataset. The final number of features was reduced to half the original size.

- **Feature Mapping:** Some features like 'Victim_Descent' and 'Victim_Sex' were represented by certain codes that were hard to understand and analyse. Therefore, we decided to map them and assign easy-to-understand values to them.

- **Categorical Encoding:** Considering that the dataset consists of only categorical features, except for the 'Victim_Age' column, which is numerical, and the date-time features ('Date_Reported' and 'Date_Occurred'), the rest of the features needed to be transformed into a numeric format, suitable for the classification models. The method that was used is *Label Encoding* due to the great variance of the string-type features.

- **Missing Values:** Only four features out of the remaining 13 features did not have any missing values: 'Time_Occurred', 'Area_Name', 'Reporting_District', 'Date_Reported', and 'Date_Occurred'. For the rest, we analysed every feature separately and reported them to the extracted year from the 'Date_Occurred' column. Then, we replaced the missing values with the median value for that year. This method worked for all features, except the 'Location_', where the missing values could not be replaced by the median value due to their irregular pattern. Thus, we decided to drop this column. However, the downside of this method is that we ended up with highly skewed distributions for most features that needed to be treated in another way.

- **Feature Grouping:** Features such as 'Crime_Code_Description' and 'Premise_Description' had more than 200 unique values which can increase the difficulty of classification. We decided to group these values in order of their frequency in the dataset. We kept the first 10 or 15 labeled values (depending on the number of unique values in the feature) as they were, and the rest were assigned to the same bucket. This process reduced the skewness of the distributions, although it was not completely removed. Also, the values from the 'Victim_Age' column were grouped into buckets and then label encoded.

- **Date-Time Features:** After pre-processing the dataset, we performed again a correlation analysis and noticed that the date-time features proved to be redundant, so we removed them.

- **Choosing the Target Feature:** The final data consisted of 10 features, all of them numerical due to the label encoding. Considering this dataset was not originally a classification-oriented dataset, thus not having a pre-defined target column, we needed to choose an appropriate feature that we wanted to classify. The chosen one was 'Victim_Age'.

3.2 College US News Dataset

Compared to the first dataset, this is a smaller one with mostly numerical features (only one categorical feature out of 33 - there is also the target feature that is categorical). The pre-processing task was less complicated and the steps are described below:

- **Feature Reduction:** An initial analysis of the dataset revealed that none of the features were redundant, therefore allowing us to keep all of them.

- **Missing Values:** There were only three features out of 34 that did not have missing values. To fill in the missing values, we replaced them with the median value for each column.

- **Outliers:** After analysing the distributions, we noticed their skewness due to the outliers. There was no particular pattern of outliers, some of the values being too little, while others to great. Therefore, we removed them using the *Clipping Method*. Moreover, considering the variance of the data, some features containing a wider range of values than others, we created a function that clips the outliers based on their IQR.

- **Feature Importance:** A tree-based method (Random Forest) was used to determine if there are any unimportant features. The results proved that every column brings a different contribution to the dataset, so no features were removed.

- **Feature Encoding:** There were two categorical features ('State' and 'binaryClass') which were transformed using *Label Encoding*.

- **Feature Scaling:** The last step is the scaling of the data. This was performed only on the training set and particularly for the MLP and KNN algorithms which are known to be sensitive to unscaled distributions. Two methods were used when classifying the data: a Standard Scaler and a MinMax Scaler.

3.3 Breast Cancer

For preprocessing the data set, two different scaling methods were used: Standardization and Normalization. Because the data set contains no missing values and only one categorical value (the one we're testing for), no further data modification is necessary.

To detect outliers, four different methods were employed: Isolation Forest, Minimum Covariance Determinant, Local Outlier Factor and One-Class SVM.

The difference in quality these methods make will be discussed in a later section.

3.4 Loan Dataset

An initial assessment revealed no missing values in both the training and test sets. The following steps outline our approach to preparing the dataset:

- **Feature Reduction**: We scrutinized the dataset for redundant or repetitive features. A notable example includes the near-perfect correlation observed between 'loan_amnt' and 'installment' and other features not highly correlated with loans.

- **Categorical Encoding**: The dataset also comprises several categorical features, primarily in string format. We adeptly transformed these into a more analysis-friendly format, utilizing methods like dummy variables and direct encoding, to better align with the analytical requirements of our models.

- **Date-Time Features**: Further enriching our dataset, we extracted valuable insights from date-time columns, such as 'issue_d_month' and 'earliest_cr_line_year', leading to the creation of new features like 'loan_age' and 'credit_history_length'.

- **Feature Scaling**: Lastly, the features were scaled, particularly for use with the MLP (Multi-Layer Perceptron) model.

Table 1: Cross-validation scores for different models.

	LGBM	LGBM_Xscaled	XGB	XGB_Xscaled	MLP	MLP_Xscaled
cv1	0.989468	0.989484	0.988880	0.988880	0.304006	0.822001
cv2	0.981581	0.982886	0.975452	0.975452	0.322019	0.838477
cv3	0.994494	0.995097	0.989428	0.989428	0.401062	0.851210
cv4	0.990330	0.989654	0.983176	0.983176	0.327703	0.833560
cv5	0.992934	0.991714	0.988970	0.988970	0.352843	0.823676

According to the Table, MLP algorithms are sensitive to the scale and distribution of input features and requires that the data be normalized or standardized. This ensures that features with larger scales do not unduly influence the model, leading to a more balanced and effective learning process. Consequently, for MLP models, we implemented feature scaling to standardize the dataset, transforming each feature to have a mean of zero and a standard deviation of one.

3.5 The Impact of Scaling

The pre-processing strategy for the dataset was tailored to the specific requirements of the machine learning models being used. While tree-based models such as XGBoost and LightGBM are scale-invariant and do not require feature scaling, the application of Multilayer Perceptron (MLP) algorithms necessitated a different approach.

4 Results

4.1 Crimes in Los Angeles from 2010

Considering the high dimensionality of the dataset and the class imbalance, the most representative performance metric for the results is the weighted F1-Score. It takes into consideration both precision and recall and does not value greater one than the other.

Three classifiers were implemented and fine-tuned on different train-test splits, varying from 10% to 25% data for the test set. Figure 1 shows the confusion matrix for the best result among them - a 42% weighted F1-Score for the Random Forest classifier, with a 10% test set split and 30 estimators.

The labels from the confusion matrix represent the encoded age buckets and have the following meaning:

- **0:** 0-17 age bucket
- **1:** 18-24 age bucket
- **2:** 25-39 age bucket
- **3:** 40-59 age bucket
- **4:** 60+ age bucket

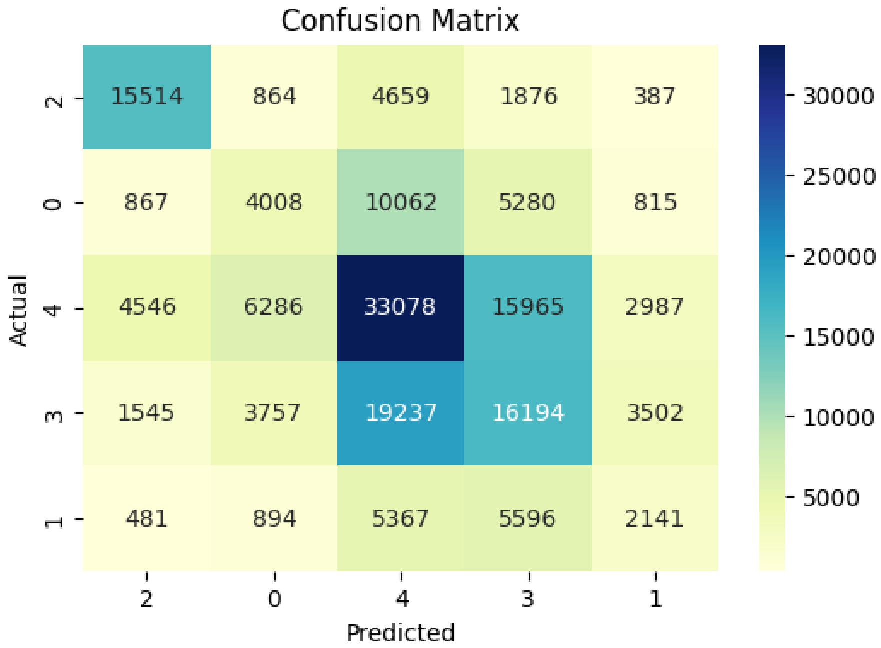


Figure 1: Confusion Matrix for Random Forest

Comparing the best results for each model, KNN, Random Forest, and MLP, we find that the overall best result is given by the Random Forest model, with a weighted F1-Score of 42%. Table 2 presents all the results.

Table 2: Performance metrics for Crimes in Los Angeles from 2010

Model	Weighted_F1	Weighted_Precision	Weighted_Recall	Accuracy
KNN	0.367	0.372	0.406	0.414
Random Forest	0.42	0.42	0.43	0.427
MLP	0.367	0.372	0.406	0.406

4.2 College US News Dataset

For this dataset, the weighted F1-Score was also used as the main performance metric.

Three classifiers were implemented and fine-tuned on different train-test splits, varying from 10% to 25% data for the test set. Figure 2 shows the confusion matrix for the best result among them - a 75.14% weighted F1-Score for the Random Forest classifier, with a 25% test set split and 200 estimators. Also, the labels 'P' and 'N' stand for *Positive* and *Negative*.

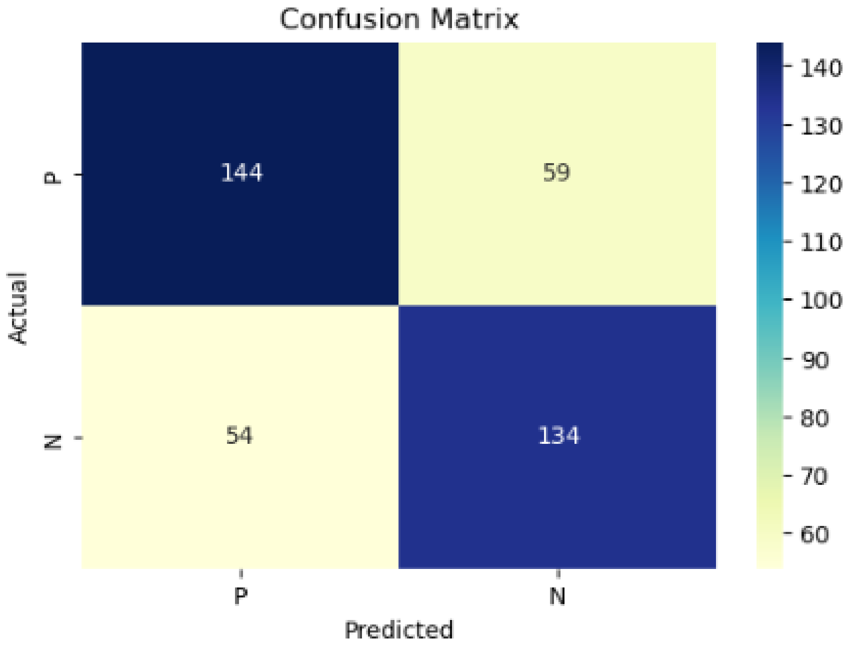


Figure 2: Confusion Matrix for Random Forest

Comparing the best results for each model, KNN, Random Forest, and MLP, we find that the overall best result is given by the Random Forest model, with a weighted F1-Score of 75.14%. Table 3 presents all the results.

Also, looking at the predicted results with and without scaling the data, we notice that the MLP is the most affected by this process.

Table 3: Weighted F1-Score for College US News Dataset

Model	No Scaler	Standard Scaler	MinMax Scaler
KNN	0.7355	0.7346	0.7213
Random Forest	0.75	0.7514	0.7482
MLP	0.7193	0.7048	0.7329

4.3 Breast Cancer

K-Nearest Neighbours

The first classification method we used is *K-Nearest Neighbours*. This method's performance is solid, especially after applying the right combination of preprocessing methods; its cross-validation accuracy sits at 0.958.

When compared to the default settings, this classifier's performance for the chosen dataset can be improved by switching the weight function to *distance*. Further, experimenting with the number of neighbours led to the value of 17 as optimal.

Random Forest

The second method, *Random Forest*, performs worst of the three with a cross-validation accuracy of only 0.947. However, it is most unaffected by switching scaling methods. In fact, it performs almost optimally without any preprocessing at all.

We found this method to work best with default parameters.

Multilayer Perceptron

The last method is *Multilayer Perceptron*. It outperforms the other two methods in all configurations; the cross-validation accuracy is 0.979.

In terms of parameter-tweaking, setting the solver to *lbfgs* yields the best results since the data set is relatively small. Experimenting with different settings shows further that results can be improved even more by increasing the tolerance for optimization to 0.01 and switching the activation function to *tanh*.

Performance

To measure performance we used accuracy scores and confusion matrices for the holdout method as well as accuracy for cross-validation. When used without any preprocessing or parameter-tweaking, the *Random Forest* method clearly outperforms the other two. This picture changes once optimizations take place: MLP then yields the most promising results.

The tables show further that *Random Forest*, unlike the other two methods, profits most from normalization, together with outlier detection via *Minimum Covariance Determinant.*

K-NN's results are relatively homogenous within the same scaler class; using standardization, *Local Outlier Factor* performs slightly better.

Unfortunately *MLP* doesn't converge fast enough when using normalization. For standardization though, *Isolation Forest* is the best choice.

Not surprisingly, each method yields slightly better results when validated via the holdout method; the exception being *K-NN.*

The confusion matrices show that save for *K-NN* (normalized), false negatives are 0. And even the combination mentioned above has a recall value of 0.97. In terms of false positives it is also *K-NN* that performs worst.

	Accuracy score (holdout)	Accuracy score (cross-validation)
K-NN	0.912	0.905
Random Forest	0.965	0.947
MLP	0.912	0.912

Table 4: Breast Cancer baseline performances (no preprocessing, default parameters).

35	0
5	17

(a) K-NN

35	0
2	20

(b) Random Forest

35	0
5	17

(c) MLP

Table 5: Breast Cancer confusion matrices for baseline performance.

4.4 Loan Dataset

Considering the imbalanced nature of the dataset, we have specifically used the weighted F1-score as our primary evaluation metric. The weighted F1-score takes into account the class imbalance and provides a more accurate

	Accuracy score (holdout)	Accuracy score (cross-validation)
Isolation Forest	0.9473684211	0.95789
Min. Covariance Det.	0.9473684211	0.95789
Local Outlier Factor	0.9649122807	0.95789
One-Class SVM	0.9473684211	0.95789

Table 6: Breast Cancer K-NN performance (Standardization, improved parameters).

35	0
4	18

(a) Isolation Forest

35	0
3	19

(b) Min. Covariance Det.

35	0
2	20

(c) Local Outlier Factor

35	0
3	19

(d) One-Class SVM

Table 7: Breast Cancer confusion matrices K-NN performance (Standardization, improved parameters).

representation of the model's overall performance. To further evaluate the model's performance on the minority class, we have also reported the F1 score for each minority class. This metric helps us gauge the model's capability to accurately predict instances from these specific classes.

In addition to class-specific metrics, we have utilized the ROC AUC curve score and confusion matrix plot. This allows us to aggregate the performance measures across all classes, taking into account the varying degrees of class imbalance across loan grades.

	Accuracy score (holdout)	Accuracy score (cross-validation)
Isolation Forest	0.8947368421	0.90877
Min. Covariance Det.	0.8947368421	0.90877
Local Outlier Factor	0.8947368421	0.90877
One-Class SVM	0.8947368421	0.90877

Table 8: Breast Cancer K-NN performance (Normalization, improved parameters).

34	1
5	17

(a) Isolation Forest

34	1
5	17

(b) Min. Covariance Det.

34	1
5	17

(c) Local Outlier Factor

34	1
5	17

(d) One-Class SVM

Table 9: Breast Cancer confusion matrices K-NN performance (Normalization, improved parameters).

	Accuracy score (holdout)	Accuracy score (cross-validation)
Isolation Forest	0.9649122807	0.92982
Min. Covariance Det.	0.9473684211	0.93333
Local Outlier Factor	0.9649122807	0.9193
One-Class SVM	0.9649122807	0.92982

Table 10: Breast Cancer Random Forest performance (Standardization, improved parameters).

35	0
2	20

(a) Isolation Forest

35	0
2	20

(b) Min. Covariance Det.

35	0
2	20

(c) Local Outlier Factor

35	0
2	20

(d) One-Class SVM

Table 11: Breast Cancer confusion matrices Random Forest performance (Standardization, improved parameters).

	Accuracy score (holdout)	Accuracy score (cross-validation)
Isolation Forest	0.9649122807	0.94737
Min. Covariance Det.	0.9824561404	0.93684
Local Outlier Factor	0.9649122807	0.94035
One-Class SVM	0.9473684211	0.94386

Table 12: Breast Cancer Random Forest performance (Normalization, improved parameters).

35	0
1	21

(a) Isolation Forest

35	0
1	21

(b) Min. Covariance Det.

35	0
3	19

(c) Local Outlier Factor

35	0
3	19

(d) One-Class SVM

Table 13: Breast Cancer confusion matrices Random Forest performance (Normalization, improved parameters).

	Accuracy score (holdout)	Accuracy score (cross-validation)
Isolation Forest	1	0.97895
Min. Covariance Det.	0.9824561404	0.97895
Local Outlier Factor	0.9824561404	0.97895
One-Class SVM	0.9824561404	0.97895

Table 14: Breast Cancer MLP performance (Standardization, improved parameters).

35	0
0	22

(a) Isolation Forest

35	0
1	21

(b) Min. Covariance Det.

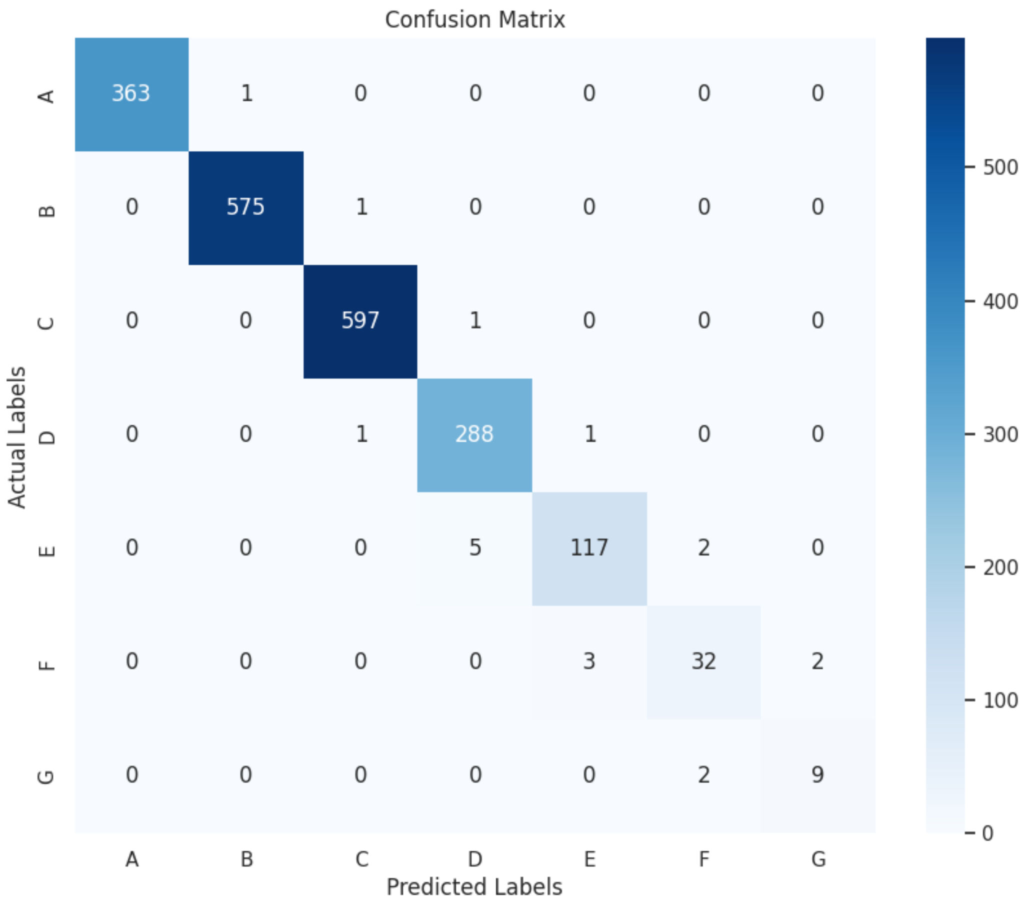
35	0
1	21

(c) Local Outlier Factor

35	0
1	21

(d) One-Class SVM

Table 15: Breast Cancer confusion matrices MLP performance (Standardization, improved parameters)



	A	B	C	D	E	F	G
A	363	1	0	0	0	0	0
B	0	575	1	0	0	0	0
C	0	0	597	1	0	0	0
D	0	0	1	288	1	0	0
E	0	0	0	5	117	2	0
F	0	0	0	0	3	32	2
G	0	0	0	0	0	2	9

Figure 3: Confusion Matrix analysis on LightGBMClassifier

Comparing among the LightGBM (LGBM), the multilayer perceptron (MLP), XGBoost (XGB), and deep learning, we find that LightGBM (LGBM) provides the most accurate performance for the classification, which has a weighted f1 score 95% and the ROC AUC score near to 1, suggesting its better ability to classify positive and negative instances. Also, it achieves a relatively high F1 score for Minority Class D, E, and F, making it more reliable for predicting this specific minority class.

However, with a significant class imbalance among the 7 classes, the high ROC AUC score may not reflect the model's performance accurately, considering the macro f1 score is not as high as the ROC AUC score. The majority classes dominate the ROC AUC calculation, while the model's performance on minority classes is relatively poor, leading to lower F1 scores. By examining the weighted F1 scores for classes D, E, and F, these three classes are particularly difficult for the model to predict accurately, and the model is

Table 16: Performance metrics of the models.

Model	ROC_AUC	Macro_f1	Weighted_f1	f1_Grade_D	f1_Grade_E	f1_Grade_F
LightGBM	0.997977	0.95	0.99	0.98	0.94	0.89
XGBoost	0.990776	0.95	0.98	0.98	0.96	0.85
MLP	0.973679	0.95	0.92	0.90	0.77	0.62

struggling to capture all positive instances or is incorrectly predicting samples as positive for these classes.

While we employed multiple techniques, including feature selection using feature importance and Stratified K-Fold Cross-Validation, but the impact of these methods on the overall results didn't always meet our initial expectations. There can be various reasons contributing to the less-than-ideal outcome. Firstly, the performance of a model heavily relies on the quality and representativeness of the dataset itself. Furthermore, the choice of hyperparameters and model configuration also plays a crucial role in achieving optimal results. It is possible that the selected hyperparameters were not well-tuned or that alternative algorithms could have been more suitable for our specific task.

Table 17: Performance optimization of LightGBM models.

Model	Weighted_f1	f1_G_D	f1_G_E	f1_G_F
LightGBM	0.99	0.99	0.96	0.88
LightGBM_Finetune	0.99	0.98	0.94	0.89
LightGBM_FeatureSelect	0.99	0.99	0.96	0.87
LightGBM_Finetune_FeatureSelect	0.98	0.98	0.94	0.86

In conclusion, parameter adjustment can in fact help improve the accuracy of the model, but this is limited. The most important thing is to improve it through data cleaning, feature selection, feature fusion, model fusion, and other means. So techniques such as class weighting, oversampling/undersampling, or even model stacking could be applied to improve the performance of the model on those challenging classes.

5 Comparison

Not surprisingly, given the differences of the four data sets, different classifiers performed best on each of them. While *Random Forest* yielded the best results for the "Crimes in Los Angeles from 2010" and the "College US

News" sets, "Breast Cancer" was best classified by *MLP* and "Loans" by *LGBM*.

The reason *Random Forest* performs well with the first two sets is their high dimensionality and the class imbalance/outlier presence respectively.

Contrary, because of the non-linearity of the "Breast Cancer" set, *MLP* turned out to be most precise.

And finally, among the methods chosen, *LGBM*, because of it being able to deal well with class imbalance, performs favorably for the "Loans" set.

Table 18: Best classifier performance per data set.

	Weighted F1-Score
"Crimes in LA" - *Random Forest*	0.42
"College US News" - *Random Forest*	0.7514
"Breast Cancer" - *MLP*	0.98
"Loans" - *LGBM*	0.98