Chris McKie

CSCI 164 Artificial Intelligence Programming

Final Project

GitHub Link: https://github.com/chrismckie/CSCI164.git

1. Dataset Exploration and Preprocessing

For this project, I am looking at two datasets: One containing data for defaulting on loans, and the other that labels credit card transactions as fraudulent or genuine.

The first is a loan default prediction dataset, and it's data comes from the Lending Club Bank. It's a huge dataset with more than 2 million rows and over a hundred features with lots missing information, so cleaning this dataset will be necessary. Some of the useful features I would want to use for the project include the loan amount, the interest rate, borrower's annual income, and debt-to-income ratio. There are a lot of features that aren't needed, like borrower's employment title, so many of these will be removed to shrink the dataset significantly. The data types of each feature are mostly numerical, although there are others such as dates. The target variable, will be the loan status, which will classify the loan as either "Fully Paid" or "Defaulted," in order to predict if a borrower will default on a loan.

The second dataset is a credit card fraud detection dataset, which comes from anonymous European credit card data from Kaggle. While this is still a large dataset, it's much smaller than the first one, having only about 280,000 records, with 30 features. Some features include the time (in seconds) from the first transaction on the card to the current one, and the amount of the transaction. Unfortunately, most of the labels are

marked as V1, V2, …, V28, as there was an Principle Component Analysis (PCA) transformation done on the data to keep them anonymous. Despite the anonymous data, the dataset still has a high usability score on Kaggle. The target variable is the Class feature, which classifies each dataset as either being Fraud or Not Fraud. All features have a numerical data type, except for the Class feature, which has a binary data type.

There were several preprocessing steps done with the loan default dataset. It contained hundreds of features, and many of them were not needed for this project. I removed them by creating a new array called keep_cols, which contained all the features I wanted to keep, then reassigned the loan dataframe to keep_cols, allowing which eliminated hundreds of unneeded columns. From there, I found that many of the rows had missing data, so I used the SimpleImputer from scikit-learn to fill the missing values with the feature's median.

I also used one-hot encoding for the loan default dataset. Several of the features were categorical, and the categories themselves were unordered, so one-hot encoding made the most sense for these features. The last preprocessing step I did with this dataset was standardization. I used the StandardScaler from scikit-learn to fit and transform the data.

Preprocessing for the credit card fraud dataset was much more straightforward, partly because the dataset had already been preprocessed. I did find some rows with missing values, but it was a very small amount compared to the size of the dataset, so I used the dropna() function to remove those rows. The dataset was already numerical,

so there was no need for encoding, so all I did after that was use the StandardScaler to fit and transform the data.

Each dataset was chosen under the theme of detecting financial risk. The first dataset looks at the risk that comes from a borrower defaulting on a loan, while the second looks at the risk that comes from credit card fraud. Both of these topics are real-world issues that AI could help mitigate for banks and other financial institutions.

Each of these datasets have also been used in other AI reports. For example, the paper “Credit Card Fraud Detection Using Machine Learning” by Meera AlEmad used the same credit card fraud dataset. This paper looked at K-Nearest Neighbors, Naive Bayes, Support Vector Machines, and Logistic Regression. The results predicted credit card fraud with a high amount of accuracy, ranging from from 97.76% to 99.94%, with the Support Vector Machine providing the most accurate results.

Another report called “LendingClub Loan Default and Profitability Prediction” by Peiqian Li and Gao Han, from Stanford University, used the loan default dataset to predict credit risk and loan profitability. The paper used Logistic Regression and Random Forest for their models. The result of the paper was a loan selection strategy with a return rate that is higher than the S&P 500’s annualized return for the past 90 years.

2. Model Development with scikit-learn

Both datasets were trained on multiple machine learning models. Both were trained using default parameters to establish benchmarks that will be compared after hyperparameter tuning.

The models used for each dataset was Logistic Regression, K-Nearest Neighbors, and Random Forest Classifier. Logistic Regression makes sense as a model for both datasets, because the loan dataset predicts whether or not a borrower will default on a loan, and the fraud dataset predicts whether or not a credit card transaction is fraud. Both target variables are binary choices, making Logistic Regression a perfect model. K-Nearest Neighbors was chosen because it is good at detecting patterns in the data, and works with non-linear relationships. For KNN, k is set to 5. Finally, the Random Forest Classifier is based on decision trees, making it a diverse addition to the three models. Initially, the features were standardized with StandardScaler to optimize the performance of the Logistic Regression and KNN models.

During the model evaluation, a train/test split was used with 80% training and 20% testing for both datasets. In addition, the random_state was set to 42 for both datasets. The model performance was evaluated using the following metrics: Accuracy, precision, recall, F1-score, and ROC-AUC. The precision and recall metrics were especially important for the fraud dataset, due to it being an imbalanced dataset.

3. Hyperparameter Tuning and Model Comparison

The initial model evaluation provided a baseline for the three models, but using hyperparameters and model tuning allowed for improving the models. I tuned the models using GridSearchCV, and I used Stratified cross-validation sets as well. Due to the nature of fraud and defaults, the datasets were imbalanced, so using stratified splits allowed for proportional classes for the models.

Different hyperparameters were used for each model. For example, Logistic Regression had regularization strength (c), and penalty type (l2) tuned. The K-Nearest Neighbors model had the number of neighbors (n_neighbors) and weighting method (uniform or distance) tuned. Finally, the Random Forest model had the number of trees (n_estimators), maximum tree depth (max_depth), and feature selection strategy (max_features) tuned. Through this process, I was able to get the best parameters for each dataset and model, allowing me to test the models again.

```
Loan Logistic Regression Best Params: {'C': 0.1, 'penalty': 'l2'}
Fraud Logistic Regression Best Params: {'C': 0.1, 'penalty': 'l2'}
```

```
Loan KNN Best Params: {'n_neighbors': 9, 'p': 2, 'weights': 'uniform'}
Fraud KNN Best Params: {'n_neighbors': 3, 'p': 1, 'weights': 'distance'}
```

```
Loan Random Forest Best Params: {'max_depth': 5, 'max_features': 'log2', 'n_estimators': 100}
Fraud Random Forest Best Params: {'max_depth': 5, 'max_features': 'log2', 'n_estimators': 50}
```

4. Results, Evaluation, and Literature Comparison

After turning, the models were re-evaluated using the same performance metrics: Accuracy, precision, recall, F1-score, and ROC-AUC, allowing me to compare both versions of each model. For the loan dataset, the baseline model was decent in predicting loan default. Logistic Regression performed best with an accuracy of 0.7977, while K-Nearest Neighbors and Random Forest had 0.7602 and 0.7966 respectively. After tuning, all three models saw improvements, but Logistic Regression still performed the best with 0.7980.

For the fraud dataset, all three models performed well, with each getting more than 0.99 accuracy, although Random Forest was barely the highest scoring. Tuning the model improved accuracy even further, although by smaller margins.

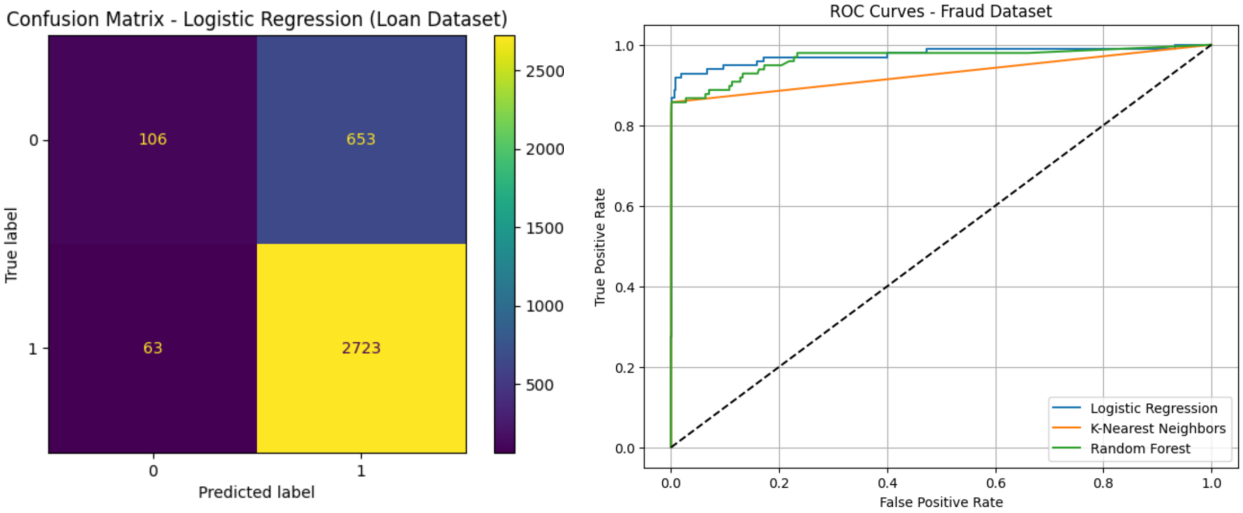
Results from the initial models:

	Dataset	Model	Accuracy	Precision	F1-Score	ROC-AUC
0	Credit Card Frad	Logistic Regression	0.999122	0.999046	0.999037	None
1	Credit Card Frad	K-Nearest Neighbors	0.999526	0.999508	0.999504	None
2	Credit Card Frad	Random Forest	0.999561	0.999552	0.999535	None
3	Loan Default	Logistic Regression	0.797743	0.767340	0.743294	None
4	Loan Default	K-Nearest Neighbors	0.760226	0.714005	0.727358	None
5	Loan Default	Random Forest	0.796615	0.764968	0.740538	None

Results after hyperparameter tuning:

	Dataset	Model	Accuracy	Precision	F1-Score	ROC-AUC
0	Credit Card Fraud	Logistic Regression (Tuned)	0.999087	0.999004	0.998991	None
1	Credit Card Fraud	K-Nearest Neighbors (Tuned)	0.999544	0.999530	0.999518	None
2	Credit Card Fraud	Random Forest (Tuned)	0.999333	0.999298	0.999277	None
3	Loan Default	Logistic Regression (Tuned)	0.798025	0.768175	0.743489	None
4	Loan Default	K-Nearest Neighbors (Tuned)	0.774048	0.721516	0.729475	None
5	Loan Default	Random Forest (Tuned)	0.785896	0.617632	0.691678	None

I used the results to create a confusion matrix for the loan dataset and an ROC curve for the fraud dataset.



Based on the confusion matrix for the loan dataset, there were a high number of false positives, meaning that while the model could accurately predict loans paid in full, it often incorrectly labeled loans as paid in full when they should have been defaulted. This gives the model a precision score of 0.80, which may be low for banks wanting to avoid risking a defaulted loan.

Meanwhile, the ROC curves for the fraud dataset shows that the Random Forest performs the best out of the three models, which is in line with the results I received. All of the models performed incredibly well with this dataset, so this graph confirms the accuracy of the models' ability to detect credit card fraud.

When comparing the models to each other, it is pretty clear that the fraud models performed better than the loan dataset. The fraud models were consistently and correctly detecting credit card fraud, while the loan dataset would often make false positive classifications. This is likely due to the differences in datasets. The loan dataset was much larger and required more preprocessing, while the credit card dataset had

been preprocessed already, and has likely been optimized for machine learning by its creator.

Looking back to the previous research papers using these datasets, the results appear to be similar. In the Stanford Loan study, the loan default models reported 0.89 weighted average in precision and recall, while my model had a weighted average of 0.85. These are similar, but my model did noticeably worse than the Stanford ones. In the fraud study, their most successful model was up to 99.94% accurate in catching fraud. My model's was 99.93% accurate, providing very similar results. I think the generally similar results across both datasets suggests that the quality of the datasets is playing a large role in the accuracy, precision, and recall of the models.

I believe the strengths of project and its models has to do with its variety. By using multiple types of models, I could compare how multiple algorithms could detect fraud and loan default using real datasets. Some limitations include the class imbalance, specifically in the fraud detection dataset. This could make it harder to be as accurate as possible. In addition, the lower than desired precision score needs improvement before banking institutions would want to use a tool like this. Despite this, this project is still useful as loan and fraud detection are real issues that banking institutions must deal with. These banks could use similar models to reduce financial risk, avoiding large resulting expenses.

Works Cited

Li, P,. & Han, G. (2018). *LendingClub loan default and profitability prediction*. Stanford University, CS229: Machine Learning. Retrieved from https://cs229.stanford.edu/proj2018/report/69.pdf

AlEmad, M. (2022). *Credit card fraud detection using machine learning* (Master's Thesis). Rochester Institute of Technology. Retrieved from https://repository.rit.edu/theses/11318/