AMEX: Credit Card Default Prediction Competition

Efthimios Vlahos

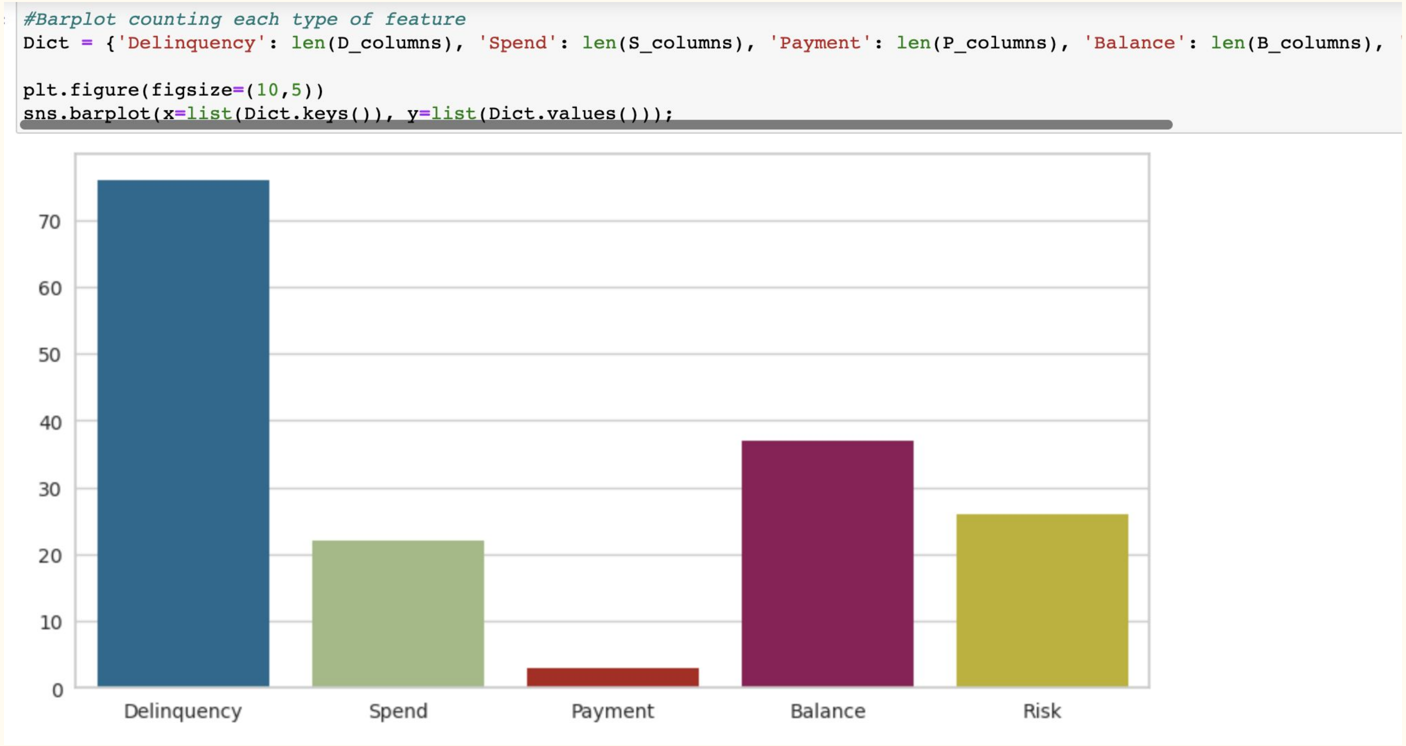
Why this Project and Project Objective

- Overall, I thought this would be a good project to look at various binary classification techniques and work with the standard python libraries for data analysis and machine learning
- The convenience of credit cards has become an indispensable aspect of modern life, facilitating daily purchases without the need to carry large amounts of cash
- However, as credit card issuers extend credit to customers, the challenge of predicting the likelihood of payment default arises. This is a complex problem that has been addressed by many existing solutions, with opportunities for further improvements
- If we are able to predict default better, the banks or creditors would, hopefully, lend money to customers more frequently which could potentially make the economy stronger as a whole and promote growth for the economy

Look at the Data

- The data for this report consists of aggregated profile features for each customer at each statement date. These features have been anonymized and normalized and can be broadly categorized into five groups: Delinquency (D_*), Spend (S_*), Payment (P_*), Balance (B_*), and Risk (R_*)
- The Delinquency variables provide information on customer payment behavior, while Spend variables describe their transaction patterns. The Payment variables relate to the amount and timing of customer payments. Balance variables provide insight into the customer's debt and available credit, and Risk variables offer information on the likelihood of default
- Overall, these anonymized and normalized features provide a comprehensive view of customer credit card usage and behavior, forming the basis of the analysis and modeling presented in this report

Look at the Data

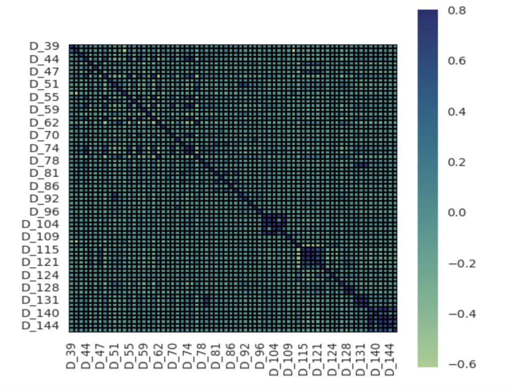


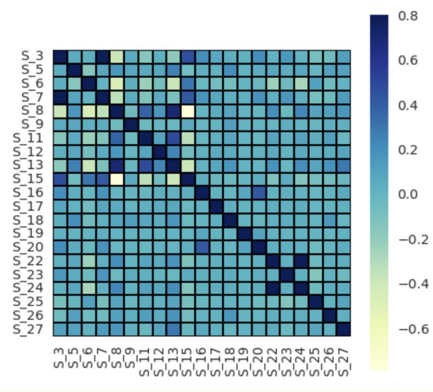
```
#Barplot counting each type of feature
Dict = {'Delinquency': len(D_columns), 'Spend': len(S_columns), 'Payment': len(P_columns), 'Balance': len(B_columns), '

plt.figure(figsize=(10,5))
sns.barplot(x=list(Dict.keys()), y=list(Dict.values()));
```

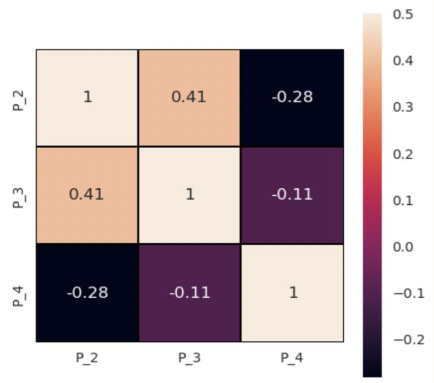
EDA

- Exploratory Data Analysis (EDA) was conducted to gain a better understanding of the competition dataset. I used Python libraries such as Pandas, Matplotlib, and Seaborn to visualize and summarize the data for each type of variable
- Through the analysis, it was revealed that the dataset was imbalanced. This led me to to include oversampling and undersampling techniques when creating the pipeline for model evaluation
- Additionally, I identified missing values in several features, which were subsequently handled using imputation techniques

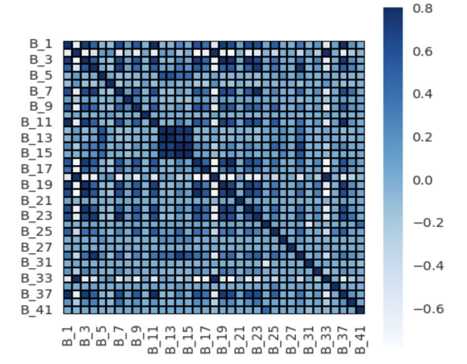
Pictures of various correlation plots for EDA

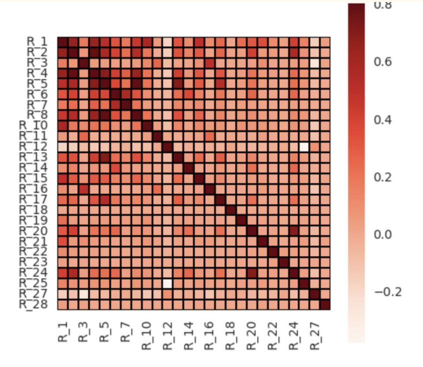


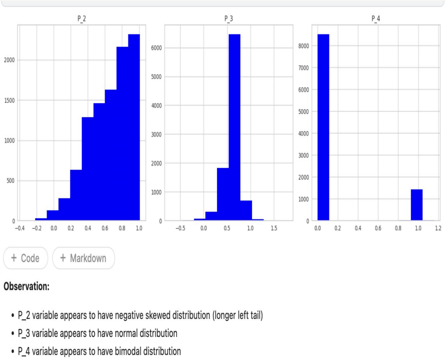




	P_2	P_3	P_4
P_2	1	0.41	-0.28
P_3	0.41	1	-0.11
P_4	-0.28	-0.11	1



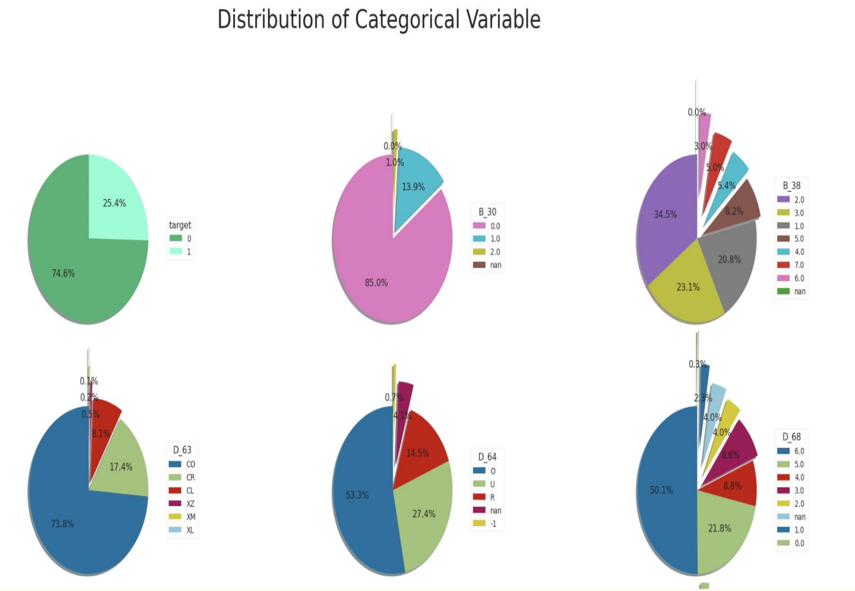


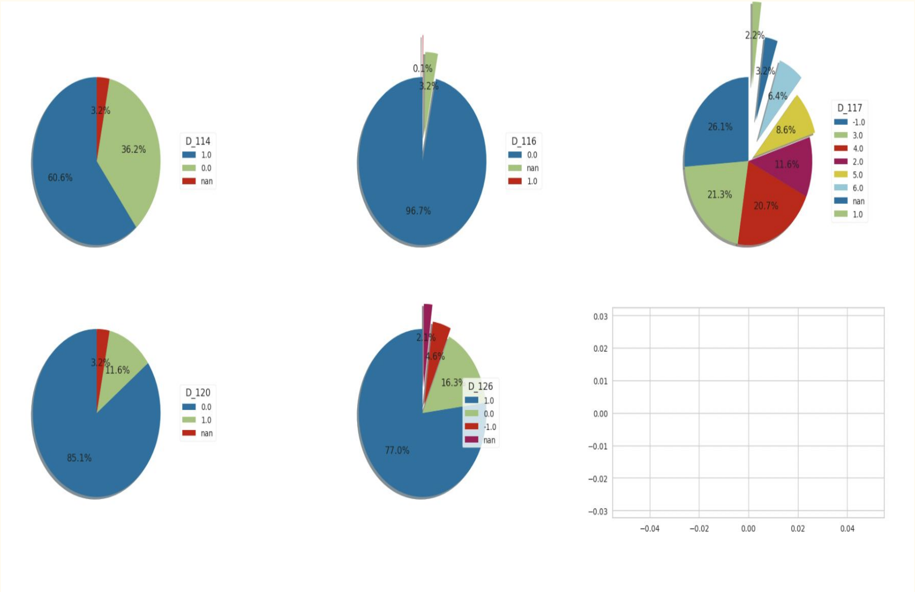


Observation:

- P_2 variable appears to have negative skewed distribution (longer left tail)
- P_3 variable appears to have normal distribution
- P_4 variable appears to have bimodal distribution

Plot of Categorical Variables

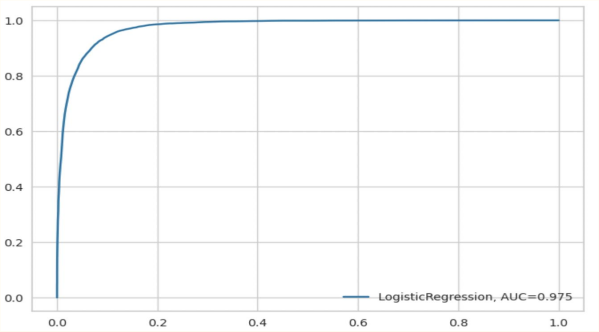


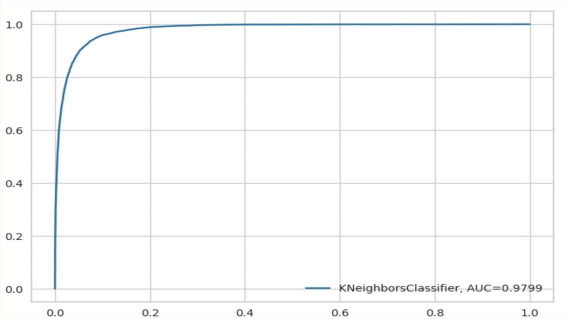


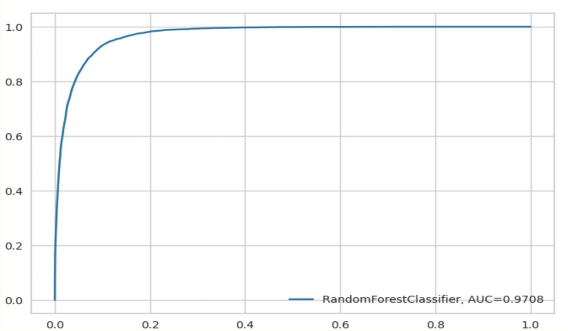
Model Building/Evaluation

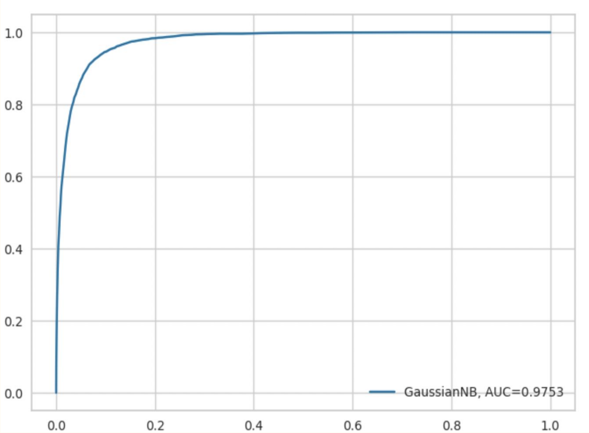
- In this study, I compared the performance of several machine learning classification models, specifically Logistic Regression, Random Forest, Naive Bayes, KNN, and Gradient Boosting for default prediction using metrics such as accuracy, precision, recall, F-1 score, validation scores, and AUC/ROC scores to evaluate model performance
- To make the code more clear and organized, I created a pipeline that streamlined the entire process from data pre-processing to model evaluation
- The pipeline consisted of several stages, including data cleaning, feature engineering/reduction, model selection, and hyperparameter tuning
- The pipeline allowed me to iterate through different models relatively quickly and experiment with various approaches

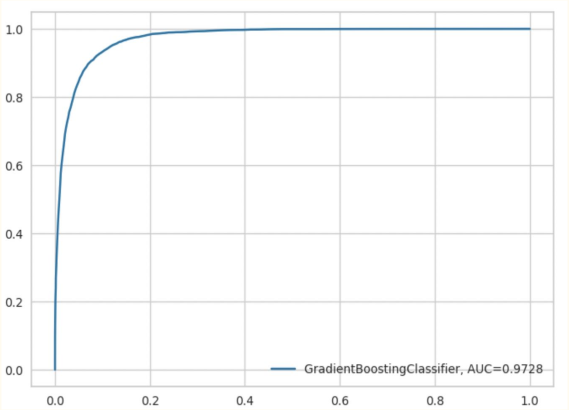
Model Building/Evaluation

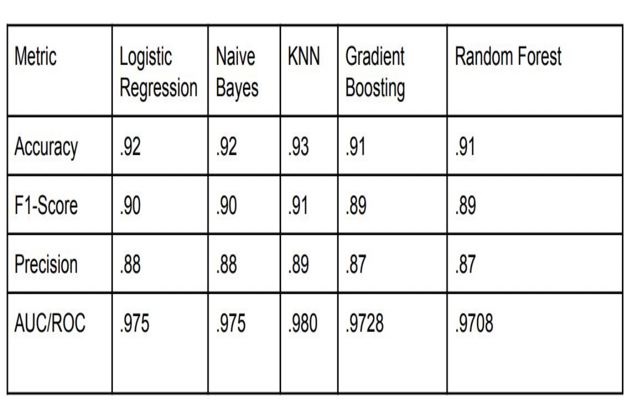












Metric	Logistic Regression	Naive Bayes	KNN	Gradient Boosting	Random Forest
Accuracy	.92	.92	.93	.91	.91
F1-Score	.90	.90	.91	.89	.89
Precision	.88	.88	.89	.87	.87
AUC/ROC	.975	.975	.980	.9728	.9708

Conclusion

- In conclusion, the machine learning models that were developed and evaluated performed exceptionally well in predicting whether a customer will default
- The AUC/ROC scores provide reliable indication of the overall performance of the models
- Overall, the competition has demonstrated the potential of machine learning in predicting default, which can have significant implications for credit risk management in the banking and financial sector