Executive Summary

Sajid Anjum
Raymond Bell
Han Le
April Gao

I. Introduction:

In the twentieth century, banking underwent significant changes that made financial services more accessible to more people. Advancements in technology and government policies played key roles in this transformation. Banks began to offer new products and services, such as personal loans and credit cards, that allowed individuals to access credit and manage their finances more easily. The development of automated teller machines (ATMs) made banking more convenient, as people no longer had to wait in long lines to make deposits or withdrawals. Additionally, the creation of online banking platforms and mobile apps further increased accessibility, allowing customers to check their account balances, transfer funds, and pay bills from the comfort of their homes. These advancements helped to democratize banking, making it more accessible to people across different socioeconomic backgrounds.

However, no matter how accessible a bank makes its services, if the banker does not make sound decisions about which services to offer which customers, it will not be able to maintain solvency. Our goal is to develop a machine learning web app that helps a banker make decisions about whether to approve or deny credit card services or loan services. In addition, we would also like to use the predictions of our machine learning algorithm to give customers some more insight about how the bank makes its decisions, and what the bank looks for when it approves or denies customers access to services in a fair and impartial manner.

II. Models

We built the model by processing and transforming datasets from Kaggle. The goal was to create a classification model that would predict whether or not a credit card or loan applicant should be approved. We used a RandomForest Classifier to create our loan data approval model and a Light Gradient Boost Classifier to create our credit card model. The confusion matrices for the testing set for the credit card model (right) and loan data model (left) are below:

```
METRICS FOR THE TESTING SET:
----------------------------
[[2100    0]
 [ 353 1760]]

              precision    recall  f1-score   support

           0       0.86      1.00      0.92      2100
           1       1.00      0.83      0.91      2113

    accuracy                           0.92      4213
   macro avg       0.93      0.92      0.92      4213
weighted avg       0.93      0.92      0.92      4213
```

```
METRICS FOR THE TESTING SET:
----------------------------
[[7621    0]
 [   0 7591]]

              precision    recall  f1-score   support

           0       1.00      1.00      1.00      7621
           1       1.00      1.00      1.00      7591

    accuracy                           1.00     15212
   macro avg       1.00      1.00      1.00     15212
weighted avg       1.00      1.00      1.00     15212
```

The Models were fairly accurate, and the features lists for both models put income and home ownership close to the top, as expected. However, in practice, we felt that the credit card model was making some interesting decisions that if a human made, he would definitely be asked to explain his reasoning. Even though we created the loan approval machine learning model, we did not have time to integrate it with the form in our website.

III. Tableau

We made a total of five Tableau Dashboards.

We made two dashboards from the credit card dataset which explored the links between marital status, education level, and average income.

We made one dashboard from the Loan Data set to examine why people took out loans (purpose), and the links between home ownership and loan repayment and the links between debt-to-income ratio and loan repayment.

Since the loan dataset had location data, we create two map dashboards to see how the number of applicants and their approval rates for loans were distributed across America

IV. Web Application

We created our website using a Flask app on Python and deployed it using the http://www.pythonanywhere.com website. The website was designed using a bootstrap grid with a bootswatch theme and has twelve pages—five for the dashboards, one home page, two for the approval forms, two for the datasets, one for about us, and one for data references.

The credit card form takes in data, sends it to the Flask app as a JSON file where it is processed and fed into our machine learning model (saved as a pickle file) and the decision to approve the loan along with the probability of approval is fed back to the website and displayed.

IV. Conclusions

1. The tree models worked the best for the classification algorithms that we made. For credit cards, both the LGBMClassifier and the RandomForestClassifier gave spectacular results, but we picked the LGBMClassifier because it showed less evidence of overfitting. We did not try to optimize the algorithms too much, or even to use neural networks as the extra effort did not seem worth it based on how well the models are already working.
2. Age, number of months with a previous credit card, total income, and total years of employment are, not surprisingly, the categories that had the largest impact on the approval algorithm.
3. While the majority of applicants were homeowners, it turns out that home ownership did not have a massive impact on approval.
4. The average income does not vary much, but high earners tend to be either married or single. It is rare for separated or widowed people to be very high earners.
5. Total Income was one of the most important factors, but very far from being determining.
6. Most people want loans for debt consolidation or move credit card debt into a loan debt.

V. Limitations and Future Work

1. One of the reasons we chose the loan dataset was that it had location data. Even though we constructed maps, we would like to delve deeply into the location data to show how variation in location or type of environment (city/suburb/rural) connects with access to finance and ability to pay off debt.
2. The machine learning models ran very quickly despite the fact that they were trained on very large datasets. It is clear that there is room for much more data to make even better predictions. We wonder if we may find similarly structured data with millions of rows that could help us train a much better model because even though the model was able to see patterns and the model statistics are beautiful, we do not feel confident in this model's predictions. One of the reasons is that since there were only 9709 rows in the credit card dataset, and the set was imbalanced, there just would not be enough data to accurately predict for a large number of situations in an input model that takes seventeen different inputs.