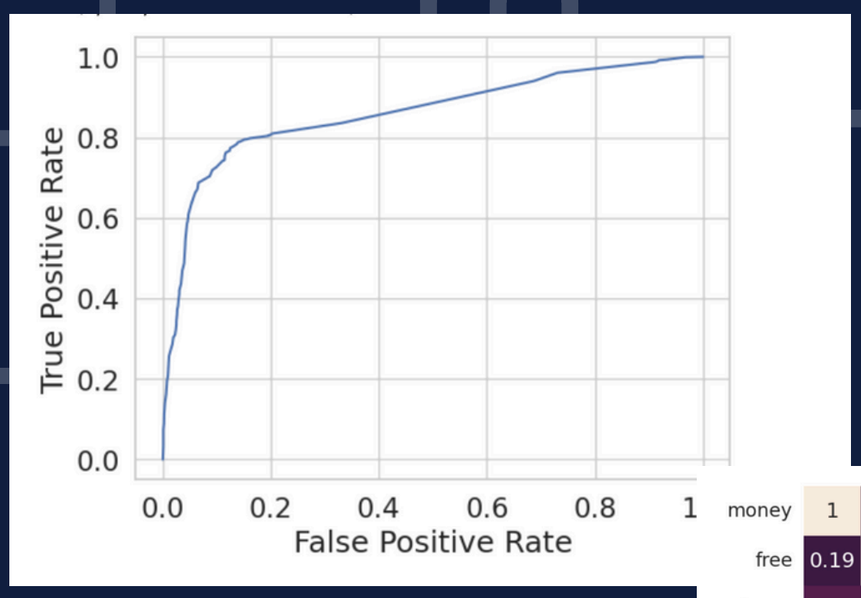

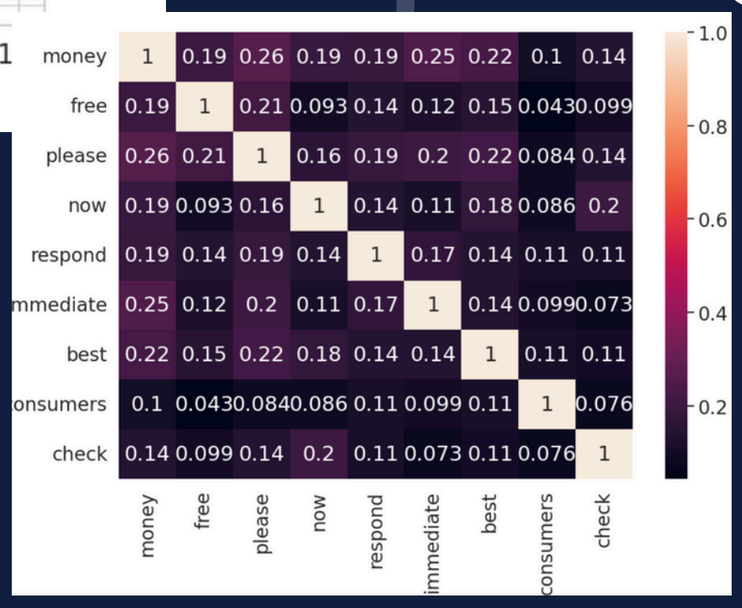

Spam Email Classifier

Python
pandas, numpy, sklearn

Steps and Challenges:

In this project, I used logistic regression on a dataset of spam and regular emails to predict whether an email was spam based on the words it contained. The goal was to identify key words that could serve as indicators of spam. I began by visualizing different words within the dataset, relying on my own understanding of common spam phrases to inform my classifier. I used one-hot encoding and Principal Component Analysis (PCA) to quantify the data and identify the most important features for my predictor. These tools, along with various packages within Scikit-learn, were essential in building the model.

One of the biggest challenges I faced was avoiding overfitting. Initially, the model's training accuracy was too high, making it ineffective on new data. This issue arose because the model relied heavily on specific words, causing it to overfit or underfit depending on the selected features. To combat this, I reduced the number of features and identified additional keywords in the spam emails, which helped balance the model. Another challenge was dealing with ambiguous emails that were difficult to classify even after I manually reviewed them. This is a common issue in prediction models, as not all cases are clear-cut. Despite this, fine-tuning the model to better handle such ambiguities was crucial. By refining the feature set and adjusting the model's parameters, I was able to improve its performance, highlighting the importance of careful model tuning in predictive analytics.

Takeaways:

- Dealing with qualitative data.
  - Understanding how to deal with and properly utilize non-numerical data was a learning curve that I experienced throughout this project. Through the use of feature engineering and one -hot-encoding, I learned how to think about ways to manipulate data while conserving its true properties in order to best apply to my model. This creative thinking is a skill that I was able to pick up on and will continue to improve on in the data science field.
- Ways to avoid overfitting.
  - With so much data, it became apparent that overfitting my model was going to pose a great challenge. This project helped me to tune into the usage of PCA and how to determine if a feature was essential to the predictor. By visualization and pure trial and error, this project gave a great introduction to how to avoid this common error.
- The differences in classifier choices matter.
  - The numerous amount of classifier choices from accuracy to recall and precision lead the way for confusion. Unaware of what to test for and what I should optimize, this project helped solidify my understanding in these classifiers and when/how to use them. This skill was incredibly useful when testing the accuracy of other models that I have constructed.

Given this is a class project, I cannot put my entire code online. However, I would be more than ready to discuss the code by request.