Credit Card Customer Segmentation with Clustering

Objective

The objective of this analysis is to utilize machine learning clustering techniques to build and optimize a model which can be implemented to segment credit card users based on transaction history. Once the dataset has been preprocessed, multiple clustering models will be fit, trained, tested, and evaluated to find an optimal model for this use case.

This machine learning model could offer an immediate financial benefit by identifying and targeting high value customers to optimize marketing campaigns. Eventually the model could be fit to segment fraudulent transactions, allowing fraud to be detected early, resulting in prevention or early reversal of the fraudulent transactions.

Potential challenges for this analysis include working with an imbalanced dataset which would have to be resampled. Also, a large dataset with limited features could make a model inefficient to train.

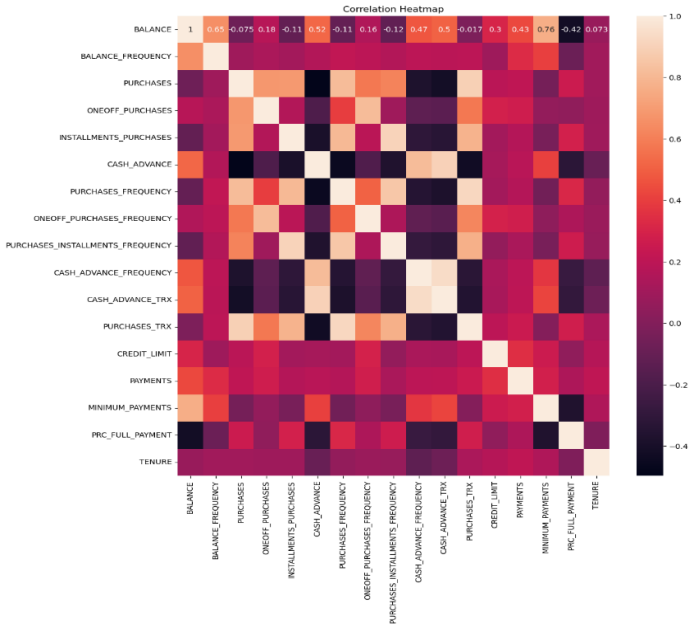
Dataset

The dataset chosen for this analysis was the "Credit Card Dataset for Clustering" from Kaggle. This dataset contains usage behavior for 9,000 credit card accounts over a duration of 6 months.

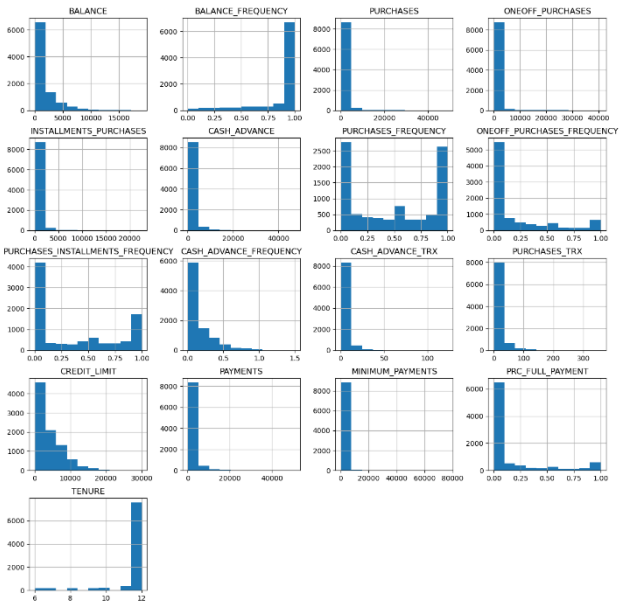
The columns in the dataset include:

- BALANCE – remaining balance in account
- BALANCE_FREQUENCY – how frequently the balance is updated
- PURCHASES – amount of purchases made from the account
- ONEOFF_PURCHASES - maximum purchase amount done in one-go
- INSTALLMENTS_PURCHASES – amount of purchase done in installment
- CASH_ADVANCE – cash in advance given by the user
- PURCHASES_FREQUENCY – how frequently purchases are being made
- ONEOFFPURCHASESFREQUENCY – how frequently purchases are happening in one-go
- PURCHASESINSTALLMENTSFREQUENCY – how frequently purchases in installments are being done
- CASHADVANCEFREQUENCY – how frequently the cash in advanced being paid
- CASHADVANCETRX – number of transactions made with "Cash in Advanced"
- PURCHASE_TRX – number of purchase transactions made
- CREDIT_LIMIT – credit limit for user
- PAYMENTS – amount of payments by the user

- MINIMUM_PAYMENTS – minimum amount of payments made by user
- PRCFULLPAYMENT – percent of fully payments made by the user
- TENURE – tenure of the credit account



The heatmap above indicates some positive as well as negative correlation between features, which is to be expected as some features in the dataset are related.



The features of this dataset were very skewed as shown by histograms to the left. Most features exhibited a right skew, but the level and direction of skew was varied between features. No features exhibited a normal distribution.

Data Cleaning and Feature Engineering

```
Out[6]: BALANCE                                  0
        BALANCE_FREQUENCY                        0
        PURCHASES                                0
        ONEOFF_PURCHASES                         0
        INSTALLMENTS_PURCHASES                   0
        CASH_ADVANCE                             0
        PURCHASES_FREQUENCY                      0
        ONEOFF_PURCHASES_FREQUENCY               0
        PURCHASES_INSTALLMENTS_FREQUENCY         0
        CASH_ADVANCE_FREQUENCY                   0
        CASH_ADVANCE_TRX                         0
        PURCHASES_TRX                            0
        CREDIT_LIMIT                             1
        PAYMENTS                                 0
        MINIMUM_PAYMENTS                       313
        PRC_FULL_PAYMENT                         0
        TENURE                                   0
        dtype: int64
```

The dataset was first checked for null values. As shown in the figure to the right, the 'CREDIT LIMIT' feature contained only one null value and as a result, that row was removed. The 'MINIMUM PAYMENTS' feature contained 313 null values, all of which were imputed with the median value which is a better estimator for skewed distributions.

The unnecessary 'CUST_ID' feature was dropped from the dataset as was the 'PURCHASES_INSTALLMENTS_FREQUENCY' feature due to high correlation.

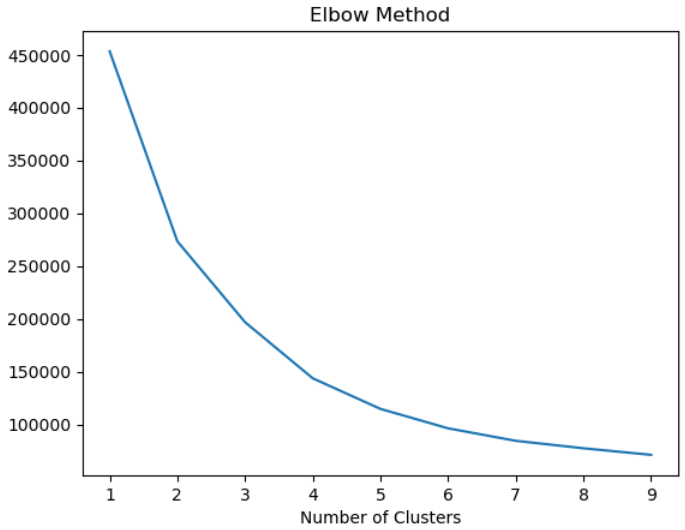
A log transformation was applied to the dataset which helped to normalize the large number of right-skewed features. Standard scaling was then implemented and Principal Component Analysis (PCA) dimensionality reduction reduced the dataset to one dimension.

Models

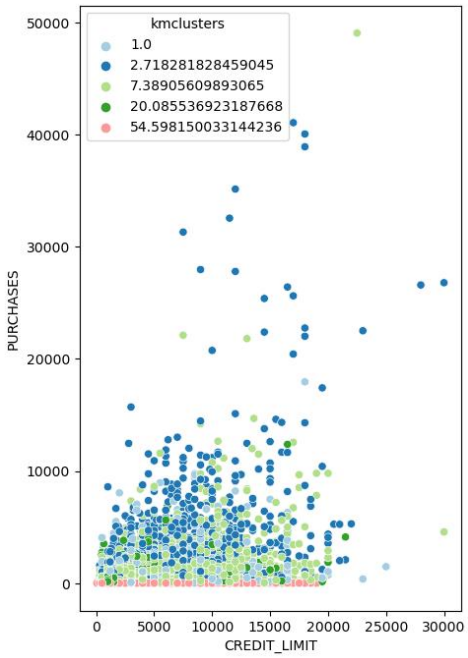
Once data analysis was completed, three models were fit utilizing three different clustering algorithms to perform segmentation on the dataset. The silhouette score performance metric was calculated and compared for each of the models.

K-Means Clustering Model

K-Means Clustering is a clustering algorithm where centroids are randomly placed, points are assigned to the nearest centroid, and then the centroids are adjusted to the mean of their clusters. This process is repeated until convergence is achieved.



The elbow method was implemented to determine the optimal number of clusters for the dataset. After examination of the figure shown above, the K-Means model was fit with five clusters. The K-Means model had a silhouette score of 0.4485, which is fair.

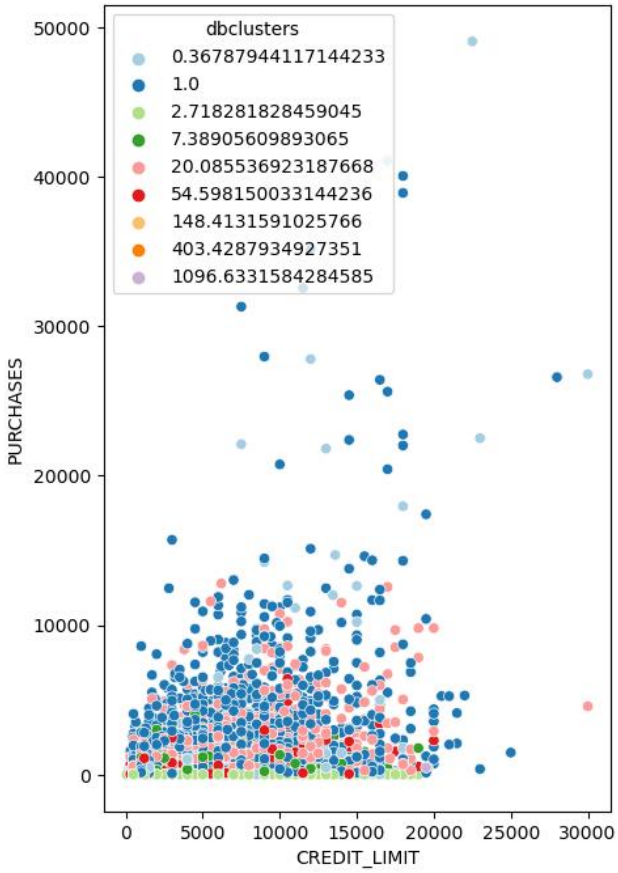


DBSCAN Model

The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm finds core points in high density regions and expands clusters from them. DBSCAN is a true clustering algorithm in that it can result in points which are not assigned to a cluster.

DBSCAN requires two hyperparameters: epsilon, the radius of the local neighborhood, and min_samples, the threshold for density of clusters. These hyperparameters were optimized using an iterative grid search algorithm. 1.9 was found to be the optimal epsilon and 17 was determined to be the best min_samples.

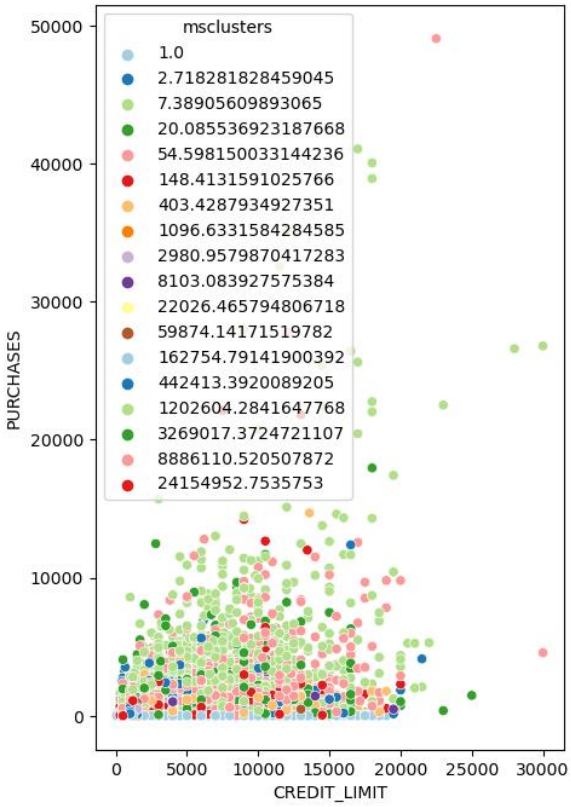
The DBSCAN model performed with a silhouette score of 0.293, significantly lower than K-Means.



Mean Shift Model

Mean Shift is a partitioning algorithm which assigns points to the nearest centroid until all points have been assigned to a cluster. The Mean Shift requires the hyperparameter bandwidth, the size of the window around each point, which was estimated using a quantile of .06, slightly above the median. The Mean Shift model

received a silhouette score of 0.4477, very close to the K-Means model but still underperforming.



Recommendation

The K-Means and Mean Shift models performed similarly according to the silhouette score metric. However, the K-Means benefits from simplicity of implementation and produced more defined clusters as show in the scatter plot visualizations above. The K-Means model is the recommended model because it can offer the most meaningful insights into the dataset in terms of customer segmentation for targeted marketing and detection.

Key Findings

- Data Preprocessing – The dataset contained a considerable amount of null values which had to be imputed and the features were significantly skewed
- Segmentation – The models were able to segment customers into fairly well defined clusters based on usage behavior

- K-Means – The first model utilizing the K-Means Clustering algorithm performed the best in regards to silhouette score and seemed to produce the most defined clusters in visualizations
- DBSCAN – The DBSCAN model performed poorly, almost half the silhouette score of the K-Means model and is not as easy to implement as K-Means due to complexity in estimating the hyperparameters
- Mean Shift – The final model which implemented the Mean Shift algorithm performed very close to the K-Means model in terms of silhouette score but resulted in many more clusters which we much less defined
- Recommendation – Of the three models which were trained, the K-Means offered the best results in terms of evaluation metrics and simplicity of implementation, making it the most likely to produce actionable results for customer segmentation

Next Steps

- Scaling – Standardization or normalization could result in a better performing model
- Transformations – Other feature transformations such as Box Cox could be implemented and optimized to correct the left skew of some features
- Hyperparameters – Testing the other elbow method inflection points as values for 'k' and seeing the impact on the model
- Evaluation Metrics – Other metrics such as the Davies-Bouldin Index, Adjusted Rand Index (ARI), or Mutual Information (MI) could produce more insights which could serve to improve the model