PROGRAMMING FOR DATA ANALYTICS

PROJECT REPORT

TOPIC: CUSTOMER SEGMENTATION AND CLASSIFICATION USING MACHINE LEARNING

CODE: 04-638A

Instructor: Dr. George Okeyo

Name: Ange Izabayo

Andrew ID: aizabayo

MS ECE 25’

Submission date on December 17, 2023

Technical Report

Abstract

For the financial banking analysis Credit card Customer segmentation play essential role tailoring Marketing strategies for diverse user groups. This project focuses on credit card customer segmentation, by addressing the need for personalizing financial services. The main purpose is to employ data-driven approach for effective customer segmentation, whereby studying on sample of 8950 credit card holders and 18 behaviour variables. The analysis involves handling missing values, outlier detection, and scaling data. Unsupervised learning, specifically KMeans clustering, is utilized to group customers. Supervised models, including Logistic Regression, Random Forest, and Support Vector Machine, are employed to determine the optimum accuracy within clusters. Hyperparameter tuning enhances model performance. The final model is deployed through a Flask web application for predicting customer segmentation, aiding in tailoring marketing strategies based on diverse user groups.

Background and Problem Description

This project focuses on credit card customer segmentation to enhance personalized financial services using a dataset from Kaggle. The primary challenge addressed is the ineffective categorization of customers, leading to low credit card usage rates and a prevalence of underutilized sleeping cards. Institutions often issue cards indiscriminately, neglecting consumer behaviour research and hindering the ability to provide tailored services, impacting customer loyalty. Customer segmentation is pivotal for targeted marketing. This project addresses the challenge of predicting customer segments for effective marketing strategies..[1], [2]

Approach

1. **Data preparation:** The initial step in data preparation involved reading the Kaggle dataset into a Pandas dataframe. To gain an insightful overview, the first and last 10 rows of the dataframe were displayed, offering a glimpse into the dataset's structure.
2. **Exploratory Data Analysis:** Perform the basic statistical measures and visualizations such as histograms and pairwise plots were used to understand feature distributions and relationships. A heatmap highlighted notable correlations, like 0.92 between purchases and one-off purchases, and identified independent features, e.g., -0.066 correlation between purchase transactions and cash advances.
3. **Pre-processing:** During data pre-processing I handled missing value by replacing with the mean, and then check the outlier in the dataset if they will not affect the result. And due to small dataset, I did not handle outlier. Then scale the data using standard scaler. Regarding the label encoder I did not use it because the categorical columns was dropped. Where the fitted scaler is saved as pickle.
4. **Unsupervised Model creation and Evaluation:** Utilized the Elbow Method and silhouette score in the KMeans unsupervised model to determine the optimal number of clusters for customer segmentation. The analysis identified three clusters, balancing within-cluster sum of squares and inter-cluster separation. The silhouette score of 0.251 affirmed the reasonable coherence and separation within these clusters. And save the label csv file.
5. **Supervised Model and Evaluation:** Implemented a supervised classification model employing Logistic Regression, Random Forest, and Support Vector Machine (SVM) algorithms on a labelled dataset. Feature scaling using the Standard Scaler preceded model evaluation, where cross-validation and accuracy served as the primary metric due to balanced classes. Learning curves were utilized to assess model fitting.

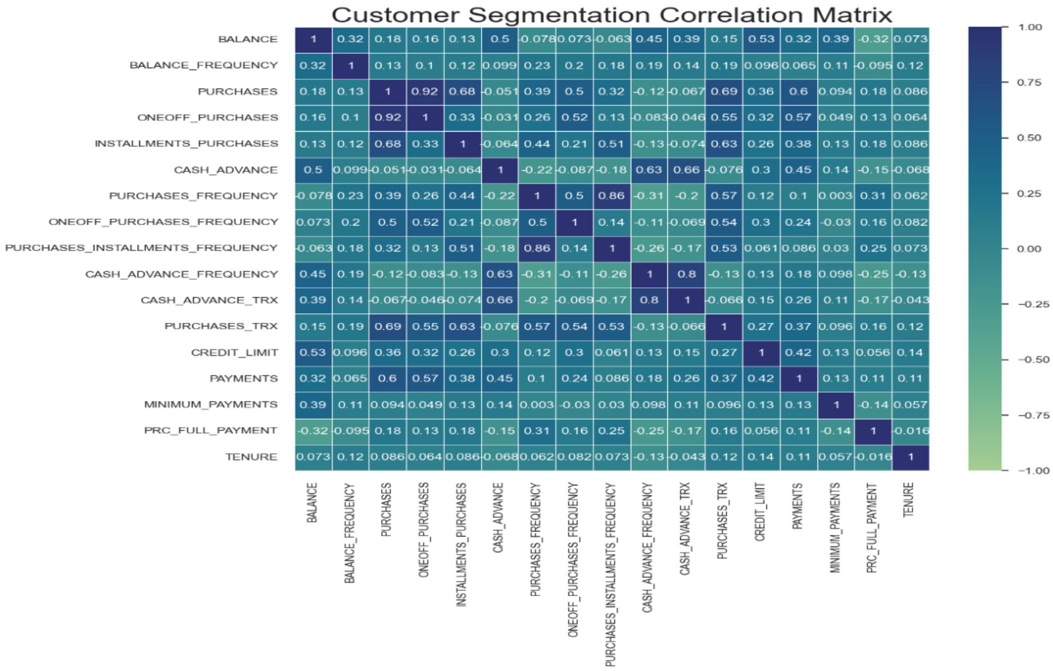
6. **Feature Selection and Engineering**: Utilized SelectKBest and f_classif for feature selection, followed by PCA for dimensionality reduction. Integrated Logistic Regression, Random Forest, and Support Vector Classifier (SVC) with the selected features and PCA-transformed data. Cross-validation assessed model performance, and average accuracy scores were recorded. Trained models were saved as pickle files for future use.[3]
7. **Hyperparameter tuning:** Comparison of model performance before and after feature selection highlighted Logistic Regression as the benchmark. Employed Grid Search for hyperparameter tuning, exploring parameters like regularization strength, penalty type, and solver type. Identified the best hyperparameter combination and saved the tuned Logistic Regression model. Evaluated the tuned model's performance and generated a visualization for comparative accuracy.[4]
8. **Model Debugging:** Learning curves are employed for model debugging, providing insights into performance during training and validation. These curves facilitate the identification of overfitting or underfitting issues by visualizing trends in training and validation accuracy. This iterative debugging process ensures the delivery of accurate and reliable credit card customer segmentation results.[5]
9. **Model deployment:** Flask, a Python web framework, is utilized to create an interactive interface. The trained model is seamlessly integrated into the Flask web application, enabling users to input features and receive real-time predictions. This approach prioritizes accessibility and usability for diverse stakeholders in the banking domain.
10. **Data Preparation**

    The exploration of the dataset revealed valuable information about its structure and content. By displaying the head and tail of the dataframe, I ensured a comprehensive understanding of the data's characteristics. Additionally, the examination of columns facilitated the identification ofkey features, setting the stage for further analysis. The process also involved checking for missing values, a crucial step in ensuring data integrity.

Result

1. Exploratory Data Analysis

The statistical overview, histograms, pairwise plots, and heatmap collectively contributed to a nuanced exploration of the dataset. Key observations included notable correlations between specific features and instances of independence, enhancing the understanding of feature dynamics. Example purchase and on/off purchases with coefficient of 0.92. For negative correlation Purchase _TXR and Cash Advanced TR with -0.066 which shows that there are not correlated each is independent.



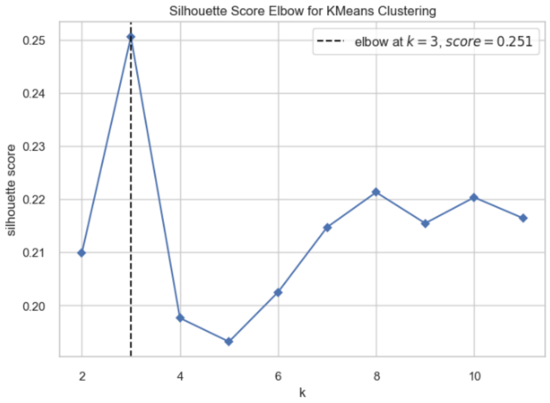
	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY	ONEOFF_PURCHASES_FREQUENCY	PURCHASES_INSTALLMENTS_FREQUENCY	CASH_ADVANCE_FREQUENCY	CASH_ADVANCE_TRX	PURCHASES_TRX	CREDIT_LIMIT	PAYMENTS	MINIMUM_PAYMENTS	PRC_FULL_PAYMENT	TENURE
BALANCE	1	0.32	0.18	0.16	0.13	0.5	-0.078	0.073	-0.063	0.45	0.39	0.15	0.53	0.32	0.39	-0.32	0.073
BALANCE_FREQUENCY	0.32	1	0.13	0.1	0.12	0.099	0.23	0.2	0.18	0.19	0.14	0.19	0.096	0.065	0.11	-0.095	0.12
PURCHASES	0.18	0.13	1	0.92	0.68	-0.051	0.39	0.5	0.32	-0.12	-0.067	0.69	0.36	0.6	0.094	0.18	0.086
ONEOFF_PURCHASES	0.16	0.1	0.92	1	0.33	-0.031	0.26	0.52	0.13	-0.083	-0.046	0.55	0.32	0.57	0.049	0.13	0.064
INSTALLMENTS_PURCHASES	0.13	0.12	0.68	0.33	1	-0.064	0.44	0.21	0.51	-0.13	-0.074	0.63	0.26	0.38	0.13	0.18	0.086
CASH_ADVANCE	0.5	0.099	-0.051	-0.031	-0.064	1	-0.22	-0.087	-0.18	0.63	0.66	-0.076	0.3	0.45	0.14	-0.15	-0.068
PURCHASES_FREQUENCY	-0.078	0.23	0.39	0.26	0.44	-0.22	1	0.5	0.86	-0.31	-0.2	0.57	0.12	0.1	0.003	0.31	0.062
ONEOFF_PURCHASES_FREQUENCY	0.073	0.2	0.5	0.52	0.21	-0.087	0.5	1	0.14	-0.11	-0.069	0.54	0.3	0.24	-0.03	0.16	0.082
PURCHASES_INSTALLMENTS_FREQUENCY	-0.063	0.18	0.32	0.13	0.51	-0.18	0.86	0.14	1	-0.26	-0.17	0.53	0.061	0.086	0.03	0.25	0.073
CASH_ADVANCE_FREQUENCY	0.45	0.19	-0.12	-0.083	-0.13	0.63	-0.31	-0.11	-0.26	1	0.8	-0.13	0.13	0.18	0.098	-0.25	-0.13
CASH_ADVANCE_TRX	0.39	0.14	-0.067	-0.046	-0.074	0.66	-0.2	-0.069	-0.17	0.8	1	-0.066	0.15	0.26	0.11	-0.17	-0.043
PURCHASES_TRX	0.15	0.19	0.69	0.55	0.63	-0.076	0.57	0.54	0.53	-0.13	-0.066	1	0.27	0.37	0.096	0.16	0.12
CREDIT_LIMIT	0.53	0.096	0.36	0.32	0.26	0.3	0.12	0.3	0.061	0.13	0.15	0.27	1	0.42	0.13	0.056	0.14
PAYMENTS	0.32	0.065	0.6	0.57	0.38	0.45	0.1	0.24	0.086	0.18	0.26	0.37	0.42	1	0.13	0.11	0.11
MINIMUM_PAYMENTS	0.39	0.11	0.094	0.049	0.13	0.14	0.003	-0.03	0.03	0.098	0.11	0.096	0.13	0.13	1	-0.14	0.057
PRC_FULL_PAYMENT	-0.32	-0.095	0.18	0.13	0.18	-0.15	0.31	0.16	0.25	-0.25	-0.17	0.16	0.056	0.11	-0.14	1	-0.016
TENURE	0.073	0.12	0.086	0.064	0.086	-0.068	0.062	0.082	0.073	-0.13	-0.043	0.12	0.14	0.11	0.057	-0.016	1

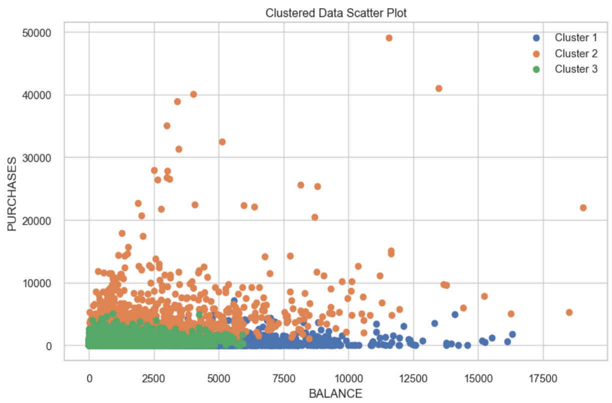
2. Pre-processing

During data pre-processing I handled missing value by replacing with the mean, and then check the outlier in the dataset if they will not affect the result. And due to small dataset I did not handle outlier. Then scale the data using standard scaler. Regarding the label encoder I did not use it because the categorical columns was dropped. Where the fitted scaler is saved as pickle.

3. Unsupervised Model creation and Evaluation

The selection of three clusters as shown in figure below is pivotal for effectively grouping customers based on underlying patterns in the data, providing a foundation for targeted marketing strategies or personalized services tailored to the distinct needs of each cluster. The selection of three clusters forms a crucial basis for effective customer grouping, facilitating targeted marketing strategies and personalized services aligned with the unique characteristics of each cluster.

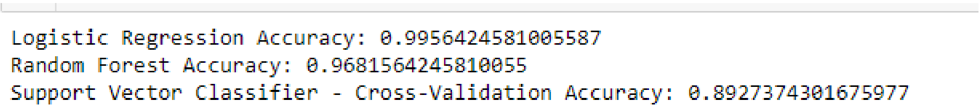




Example plotting to visually represent the clustering results in a two-dimensional space defined by the features BALANCE and PURCHASES. The code iterates through each cluster, extracts the data points belonging to that cluster, and plots them with distinct colours for each cluster.

4. Supervised model and Evaluation

Supervised classification model was built and evaluated using Logistic Regression, Random Forest, and Support Vector Machine (SVM) algorithms on a labelled dataset. The features were first scaled using the Standard Scaler. For model evaluation, cross-validation was employed, and accuracy was chosen as the primary metric for its suitability in balanced classes. Learning curves were generated to assess model fitting, indicating that all three models: Logistic Regression, Random Forest, and SVM

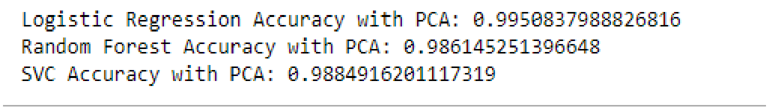


```
Logistic Regression Accuracy: 0.9956424581005587
Random Forest Accuracy: 0.9681564245810055
Support Vector Classifier - Cross-Validation Accuracy: 0.8927374301675977
```

The output shows that the logistic regression has high accuracy compared to other models.

5. Feature Selection and Engineering

With the selected features are 12 in 17 features where below is the cross-validation score. After feature selection, Logistic Regression showed a slight decrement in performance, while other models exhibited performance improvement. The bar graph below illustrates the comparative accuracy of each model.

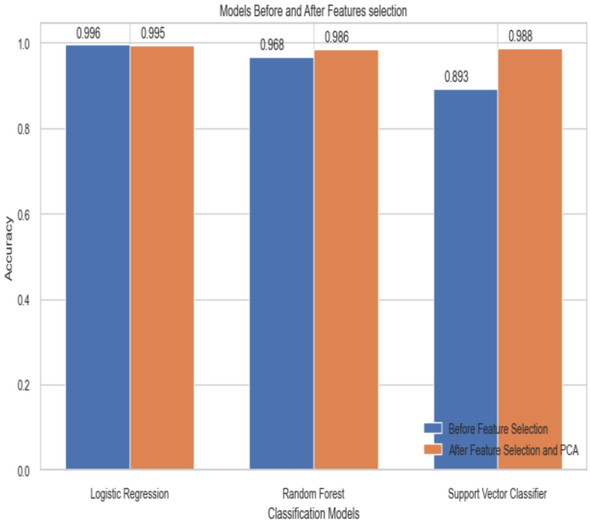


```
Logistic Regression Accuracy with PCA: 0.9950837988826816
Random Forest Accuracy with PCA: 0.986145251396648
SVC Accuracy with PCA: 0.9884916201117319
```

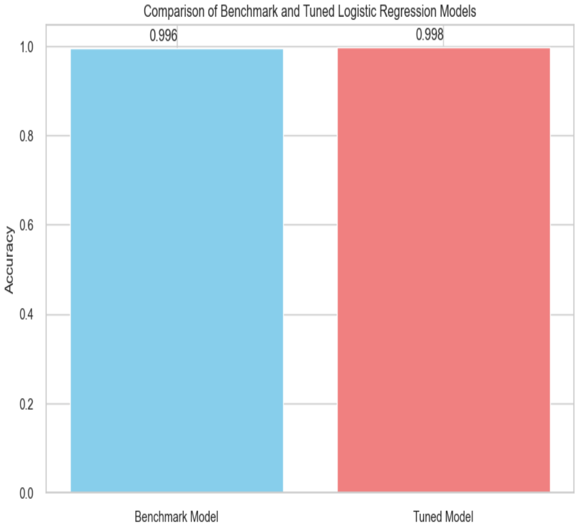
Comparing the performance after feature selection logistic regression decremented a bit but other model performance increases and the bar graph below or comparing accuracy of each.

6. Hyperparameter tuning

Hyperparameter tuning significantly improved model performance, enhancing accuracy from 0.996 in the benchmark Logistic Regression to 0.988 after tuning, as illustrated in the figure above. This iterative debugging process ensures the delivery of accurate and reliable credit card customer segmentation results.



Models Before and After Features selection

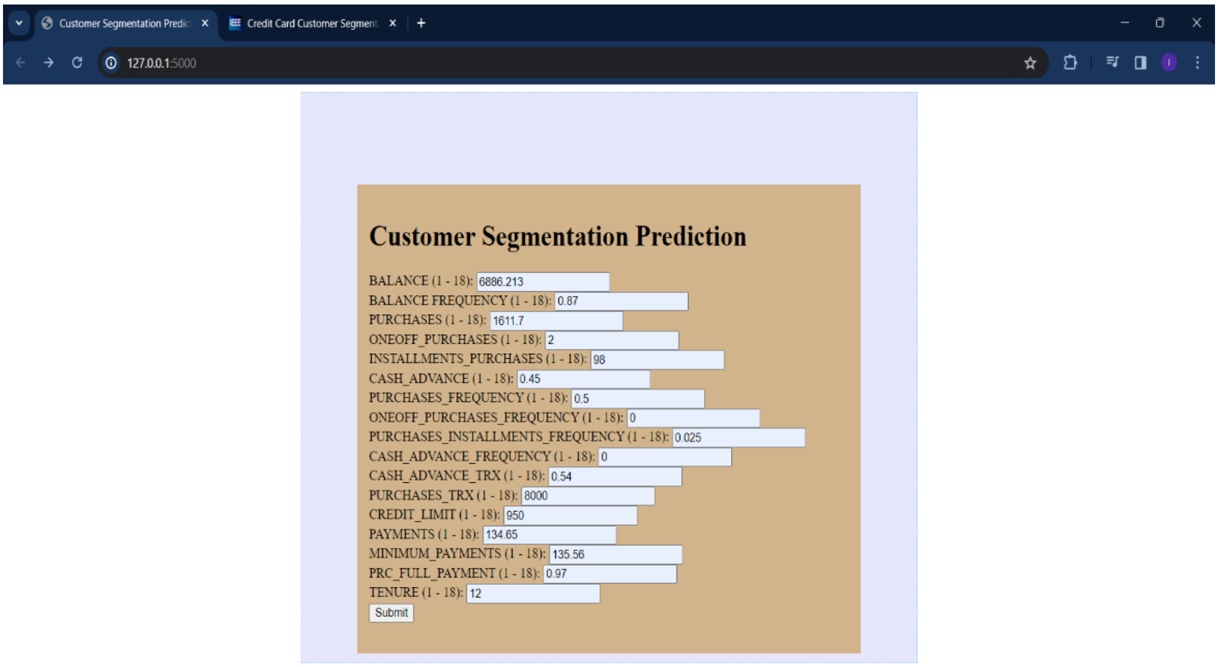


Comparison of Benchmark and Tuned Logistic Regression Models

.

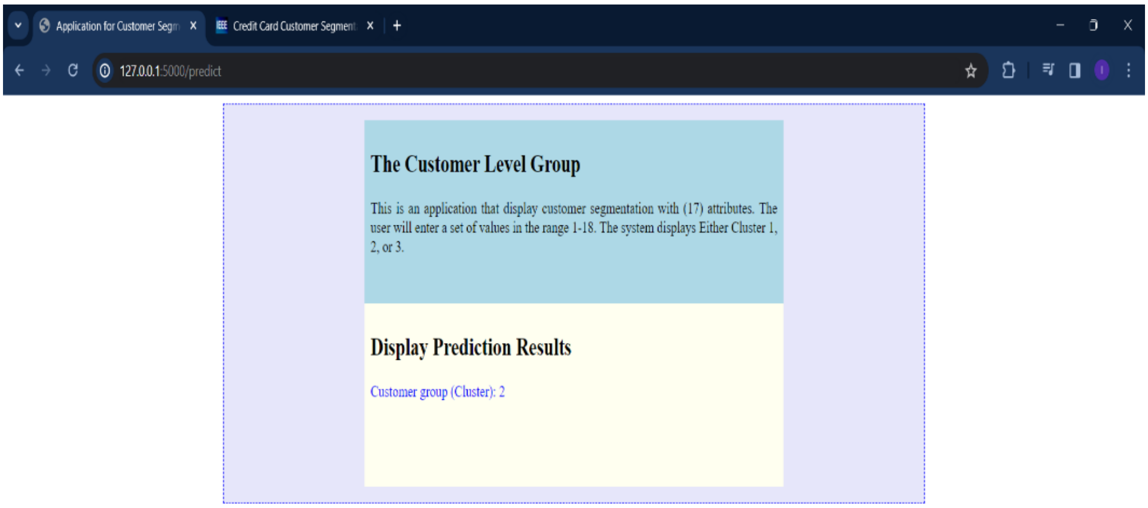
Model debugging using learning Curve: To ensure the robustness and effectiveness of the models, learning curves are utilized for debugging. Learning curves provide insights into the model's performance during training and validation phases, aiding in identifying issues such as overfitting or underfitting. By visualizing the trends in training and validation accuracy. This iterative process of debugging ensures the delivery of accurate and reliable credit card customer segmentation results.

Model deployment: Flask, a web framework for Python, is employed for creating an interactive and user-friendly interface. The trained model is integrated into the Flask web application, allowing users to input relevant features and receive real-time predictions. This deployment strategy ensures accessibility and usability for diverse stakeholders in the banking domain.



Customer Segmentation Prediction

BALANCE (1 - 18): 6886.213
BALANCE FREQUENCY (1 - 18): 0.87
PURCHASES (1 - 18): 1611.7
ONEOFF_PURCHASES (1 - 18): 2
INSTALLMENTS_PURCHASES (1 - 18): 98
CASH_ADVANCE (1 - 18): 0.45
PURCHASES_FREQUENCY (1 - 18): 0.5
ONEOFF_PURCHASES_FREQUENCY (1 - 18): 0
PURCHASES_INSTALLMENTS_FREQUENCY (1 - 18): 0.025
CASH_ADVANCE_FREQUENCY (1 - 18): 0
CASH_ADVANCE_TRX (1 - 18): 0.54
PURCHASES_TRX (1 - 18): 8000
CREDIT_LIMIT (1 - 18): 950
PAYMENTS (1 - 18): 134.65
MINIMUM_PAYMENTS (1 - 18): 135.56
PRC_FULL_PAYMENT (1 - 18): 0.97
TENURE (1 - 18): 12



The Customer Level Group

This is an application that display customer segmentation with (17) attributes. The user will enter a set of values in the range 1-18. The system displays Either Cluster 1, 2, or 3.

Display Prediction Results

Customer group (Cluster): 2

The Figure above illustrate a user-friendly Flask web application designed for credit card customer segmentation prediction. Users input values for 17 financial attributes, ranging from balance and purchase frequency to credit limit and tenure. The application promptly displays the predicted customer group or cluster, exemplified by Cluster 2 in the provided instance. This interactive platform, with its intuitive design and real-time predictions, offers stakeholders valuable insights into customer behaviour, fostering informed decision-making for tailored marketing strategies and improved customer engagement.

Conclusion

In Conclusion, the project utilized a data-driven approach for credit card customer segmentation, employing unsupervised KMeans and supervised models and different means for accurate predictions. The deployment of the tuned model using Flask ensures a user-friendly interface for real-time customer group predictions. The results offer financial institutions valuable insights for tailored marketing strategies and customer-level decision-making. For more clarification markdown in notebook details the work.

Reference

[1] W. Li, X. Wu, Y. Sun, and Q. Zhang, “Credit Card Customer Segmentation and Target Marketing Based on Data Mining,” in *2010 International Conference on Computational Intelligence and Security*, Dec. 2010, pp. 73–76. doi: 10.1109/CIS.2010.23.

[2] “Segmenting credit card customers for insight and profit.” Accessed: Dec. 17, 2023. [Online]. Available: https://www.electronicpaymentsinternational.com/features/segmenting-credit-card-customers-for-insight-and-profit/

[3] “sklearn.feature_selection.SelectKBest,” scikit-learn. Accessed: Dec. 17, 2023. [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.feature_selection.SelectKBest.html

[4] M. Gusarov, “Do I need to tune logistic regression hyperparameters?,” CodeX. Accessed: Dec. 17, 2023. [Online]. Available: https://medium.com/codex/do-i-need-to-tune-logistic-regression-hyperparameters-1cb2b81fca69

[5] “Underfitting vs. Overfitting,” scikit-learn. Accessed: Dec. 17, 2023. [Online]. Available: https://scikit-learn/stable/auto_examples/model_selection/plot_underfitting_overfitting.html