

**ACS 442 ARTIFICIAL NEURAL NETWORK**

**TargetWise Analytics**

**Customer Segmentation for Kenyan Cooperatives**

**Group Members**

1. Kemp Kuria 22-2984
2. Joy Lutenyo 21-2104
3. Allen Otieno 20-0822
4. Samson Mutua 22-2931

**Documentation**

**Project Title**: TargetWise Analytics: Customer Segmentation for Kenyan Cooperatives  
**Authors**: Joy Lutenyo, Allen Otieno, Kemp Kuria, Samson Mutua  
**Developed in**: Google Colab  
**Tools Used**: Python, scikit-learn, Streamlit, ngrok, pandas, numpy, matplotlib, seaborn

**1. Introduction**

**1.1 Project Overview**

TargetWise Analytics is a machine learning solution designed to help Kenyan cooperative businesses, such as Saccos and agricultural coops under Coop Affairs, segment their customers for targeted marketing strategies. Cooperatives in Kenya serve over 14 million members, contributing significantly to financial inclusion and rural development (International Cooperative Alliance, 2023). However, many struggle with generic marketing due to diverse customer bases, including rural farmers, urban professionals, and small business owners. This project addresses this challenge by using KMeans clustering to group customers into meaningful segments and a Random Forest Classifier to predict segments for new customers, enabling personalized marketing.

The project consists of two scripts developed in Google Colab:

* **Full Pipeline App**: Handles data preprocessing, clustering, classification, and insights generation.
* **Real-Time Prediction App (app.py)**: Allows cooperatives to input new customer details and predict their segment in real-time, with marketing recommendations.

**1.2 Objectives**

* **Segment Customers**: Group customers into distinct segments based on financial and behavioral data using machine learning.
* **Predict Segment Membership**: Build a predictive model to classify new customers into segments for real-time marketing decisions.
* **Enhance Marketing Strategies**: Provide actionable insights for Coop Affairs to design targeted campaigns, improving customer engagement and retention.

**1.3 Kenyan Relevance**

Customer segmentation is highly relevant for Kenyan cooperatives, as it addresses the challenge of delivering personalized services to a diverse customer base. For example, rural farmers may need affordable loans for agricultural inputs, while urban professionals may seek investment products. By understanding these differences, Coop Affairs can design campaigns that resonate with each group, aligning with Kenya’s Vision 2030 goal of leveraging technology for economic development in the cooperative sector.

**2. Setup Instructions**

**2.1 Prerequisites**

To run this project, you need access to Google Colab and the following tools:

* **Google Colab Account**: A free Google account to access Colab.
* **Python Libraries**:
  + pandas, numpy: For data manipulation.
  + scikit-learn: For machine learning models (KMeans, Random Forest, StandardScaler, PCA).
  + matplotlib, seaborn: For visualizations.
  + Streamlit: For building the interactive app.
  + pyngrok: For hosting the app via a public URL.
* **ngrok Authtoken**: Required to create a public URL for the Streamlit app. Sign up at [ngrok.com](https://ngrok.com) to get your authtoken.

**2.2 Installation Steps**

1. **Open Google Colab**:
   * Go to [Google Colab](https://colab.research.google.com) and create a new notebook or upload Group\_02(copy).ipynb.
2. **Install Dependencies**:
   * Run the following cell to install the required libraries:

python

!pip install pandas numpy scikit-learn matplotlib seaborn streamlit pyngrok

* + This installs all necessary libraries for data processing, modeling, visualization, and app deployment.

1. **Download and Set Up ngrok**:
   * ngrok is used to create a public URL for the Streamlit app. Run these cells to download and unzip ngrok:

python

!wget https://bin.equinox.io/c/bNyj1mQVY4c/ngrok-v3-stable-linux-amd64.zip

!unzip ngrok-v3-stable-linux-amd64.zip

* + Set your ngrok authtoken (replace 'YOUR\_NGROK\_AUTHTOKEN' with your actual token):

python

from pyngrok import ngrok

ngrok.set\_auth\_token('YOUR\_NGROK\_AUTHTOKEN')

1. **Upload the Dataset**:
   * The project uses the Kaggle Credit Card Dataset. Download it from [Kaggle](https://www.kaggle.com/datasets) or use the provided dataset.
   * In the Colab notebook, run the cell to upload the dataset:

python

from google.colab import files

uploaded = files.upload()

* + Upload your CSV file (e.g., credit\_card\_data.csv).

**2.3 Running the Scripts**

* **Full Pipeline App**:
  + The first script is embedded in the Colab notebook (Group\_02(copy).ipynb). Run the cells sequentially to preprocess data, cluster customers, train the model, and save it.
* **Real-Time Prediction App (app.py)**:
  + The app.py script is written to a file in the notebook. Run the following cell to create and run the app:

python

!streamlit run app.py &>/dev/null &

public\_url = ngrok.connect(8501)

print(f"Streamlit app is running at: {public\_url}")

* + This will generate a public URL (e.g., https://<random-string>.ngrok-free.app) to access the app.

**3. Methodology**

**3.1 Dataset Description**

* **Source**: Kaggle Credit Card Dataset.
* **Size**: 1319 records, each representing a customer.
* **Features**:
  + card: Whether the customer has a credit card (yes/no).
  + reports: Number of derogatory reports.
  + age: Customer age in years.
  + income: Annual income in dollars.
  + share: Ratio of monthly credit card expenditure to yearly income.
  + expenditure: Average monthly credit card expenditure.
  + owner: Whether the customer owns their home (yes/no).
  + selfemp: Whether the customer is self-employed (yes/no).
  + dependents: Number of dependents.
  + months: Months living at current address.
  + majorcards: Number of major credit cards held.
  + active: Number of active credit accounts.
* **Limitation**: The dataset represents credit card users, which may not fully capture the demographics of Kenyan cooperative members, such as rural farmers who prioritize savings over credit.

**3.2 Full Pipeline App (First Script)**

The first script, embedded in the Colab notebook, handles the entire workflow from data preprocessing to insights generation.

1. **Data Preprocessing**:
   * **Load Dataset**: The dataset is loaded using pandas.read\_csv() after uploading the CSV file.
   * **Handle Missing Values**: Missing values are dropped using df.dropna() to ensure the data is complete for analysis.
   * **Feature Engineering**:
     + Categorical features (card, owner, selfemp) are converted to dummy variables using pd.get\_dummies() to make them numerical for modeling.
     + Numerical features (reports, age, income, share, expenditure, dependents, months, majorcards, active) are standardized using StandardScaler to put them on the same scale, which is crucial for KMeans clustering.
   * **Output**: A cleaned and standardized dataset ready for clustering.
2. **Clustering**:
   * **Model**: KMeans clustering is used to group customers into segments.
   * **Optimal Clusters**: The Elbow Method is applied to determine the optimal number of clusters. It plots the Within-Cluster Sum of Squares (WCSS) against the number of clusters, and the "elbow" point at k=4 indicates the best number of clusters.
   * **Implementation**: KMeans is applied with 4 clusters using KMeans(n\_clusters=4, init='k-means++', random\_state=42).
   * **Visualization**: Principal Component Analysis (PCA) reduces the data to 2D, and a scatter plot visualizes the clusters, showing how customers are grouped.
   * **Output**: Customers are assigned to one of 4 clusters, saved as a new column Cluster in the dataset.
3. **Classification**:
   * **Model**: A Random Forest Classifier is trained to predict cluster membership for new customers.
   * **Data Split**: The dataset is split into 80% training and 20% testing sets using train\_test\_split.
   * **Training**: The Random Forest model is trained with 100 trees (n\_estimators=100, random\_state=42).
   * **Evaluation**: The model achieves an accuracy of 97.35%, with a detailed classification report showing precision, recall, and F1-score for each cluster.
   * **Save Model**: The trained model and scaler are saved as rf\_model.pkl and scaler.pkl using pickle for use in the real-time app.
4. **Insights**:
   * **Cluster Analysis**:
     + Cluster 0: Moderate precision and recall, indicating potential overlap with other segments.
     + Cluster 1: High recall and precision, likely loyal customers with consistent behavior.
     + Cluster 2: Smallest group but well-defined, suggesting a niche segment.
     + Cluster 3: High precision, ideal for targeted campaigns.
   * **Recommendations**: Tailor marketing strategies for each cluster, such as loyalty programs for Cluster 1 and targeted campaigns for Cluster 3.

**3.3 Real-Time Prediction App (app.py)**

The second script, app.py, is a Streamlit app for real-time segment prediction.

1. **Input Form**:
   * Users input customer details through a form:
     + Numerical features (e.g., age, income, expenditure) are entered via sliders or number inputs.
     + Categorical features (e.g., card, owner, selfemp) are selected from dropdowns (yes/no).
   * The form is user-friendly, designed for non-technical users like cooperative staff.
2. **Prediction**:
   * The app loads the saved Random Forest model (rf\_model.pkl) and scaler (scaler.pkl).
   * Input data is standardized using the scaler, and categorical features are converted to dummy variables to match the training data format.
   * The model predicts the customer’s segment (e.g., Cluster 1) in real-time.
3. **Output**:
   * The predicted segment is displayed (e.g., "The customer belongs to Cluster 1").
   * Insights and recommendations are provided based on the predicted cluster, such as suggesting loyalty programs for Cluster 1.
4. **Styling**:
   * **Fixed Title**: "TargetWise Analytics" is fixed at the top using CSS (position: fixed).
   * **Background Colors**: Sidebar is #d3d8e8 (medium grayish-blue), main content is #f0f2f6 (light grayish-blue).
   * **Font**: Black font (#000000) ensures visibility on light backgrounds.

**3.4 Deployment**

* The app is deployed in Google Colab using Streamlit and ngrok.
* Streamlit creates an interactive web app, and ngrok generates a public URL (e.g., https://<random-string>.ngrok-free.app) to access the app.
* If you encounter the ERR\_NGROK\_324 error (tunnel limit exceeded), terminate existing tunnels using ngrok.kill() and retry.

**4. App Usage Guide**

**4.1 Full Pipeline App**

1. **Upload Data**:
   * Navigate to the "Data Upload & Preprocessing" page in the app (first script).
   * Upload your customer dataset in CSV or XLSX format.
   * The app will display the first 5 rows and preprocess the data (drop missing values, standardize features).
2. **Clustering**:
   * Go to the "Clustering" page.
   * The app uses the Elbow Method to determine the optimal number of clusters (k=4) and applies KMeans clustering.
   * View the cluster distribution and PCA scatter plot to see how customers are grouped.
   * Download the processed dataset with cluster labels.
3. **Classification**:
   * Navigate to the "Classification" page.
   * The app trains a Random Forest Classifier on the cluster labels and displays the accuracy (97.35%) and classification report.
   * The trained model and scaler are saved for use in the real-time app.
4. **Insights**:
   * Go to the "Insights & Recommendations" page.
   * Review cluster-specific insights (e.g., Cluster 1: loyal customers) and marketing recommendations (e.g., loyalty programs).

**4.2 Real-Time Prediction App (app.py)**

1. **Access the App**:
   * Run the deployment cell in the Colab notebook to get the public URL.
   * Open the URL in a browser to access the Streamlit app.
2. **Enter Customer Details**:
   * On the app’s main page, fill out the form with customer details:
     + Numerical inputs: Use sliders for age (18-100) and number inputs for income, expenditure, etc.
     + Categorical inputs: Select yes/no for card, owner, and selfemp.
   * Click "Predict Segment" to submit.
3. **View Results**:
   * The app displays the predicted segment (e.g., "The customer belongs to Cluster 1").
   * Expand the "Insights" section to see cluster-specific details (e.g., "Cluster 1: High recall and precision, possibly loyal customers").
   * Expand the "Recommendations" section for marketing tips (e.g., "Tailor campaigns to maximize engagement").
4. **Reset and Retry**:
   * Click "Reset" to clear the form and enter new customer details.

**5. Results and Insights**

**5.1 Clustering Results**

* **Number of Clusters**: 4 distinct customer segments were identified using KMeans clustering.
* **Visualization**: PCA scatter plots show clear separation between clusters, confirming the effectiveness of the segmentation.
* **Cluster Distribution**:
  + Cluster 0: 75 customers (moderate spenders with potential overlap).
  + Cluster 1: 15 customers (small but loyal group).
  + Cluster 2: 25 customers (niche segment).
  + Cluster 3: 149 customers (largest, distinct group).

**5.2 Classification Results**

* **Accuracy**: The Random Forest Classifier achieved an accuracy of 97.35% on the test set.
* **Classification Report**:
  + Cluster 0: Precision 0.973, Recall 0.973, F1-score 0.973.
  + Cluster 1: Precision 1.000, Recall 1.000, F1-score 1.000.
  + Cluster 2: Precision 0.889, Recall 0.960, F1-score 0.923.
  + Cluster 3: Precision 0.986, Recall 0.973, F1-score 0.980.
* **Interpretation**: The model performs well across all clusters, with Cluster 1 being the most accurately predicted, likely due to its consistent behavior.

**5.3 Insights**

* **Cluster 0**: Moderate precision and recall, indicating potential overlap with other segments. Marketing strategies should focus on distinguishing this segment more clearly, perhaps by offering general promotions.
* **Cluster 1**: High recall and precision, making it the most consistently identified group. This segment likely represents loyal customers, suitable for loyalty programs to encourage continued engagement.
* **Cluster 2**: Smallest group but well-defined, suggesting a niche segment. Targeted campaigns for this group can focus on their specific needs, such as specialized financial products.
* **Cluster 3**: High precision and recall, making it a distinct segment ideal for targeted campaigns. This group may include high spenders, perfect for premium product offerings.

**5.4 Impact**

* The app enables data-driven marketing for Kenyan cooperatives, improving customer engagement and retention.
* It supports business growth by helping cooperatives under Coop Affairs meet the diverse needs of their members through personalized campaigns.

**6. Challenges and Solutions**

**6.1 Dataset Bias**

* **Challenge**: The dataset overrepresents urban, high-income credit card users, potentially missing the needs of rural cooperative members like farmers who prefer savings products.
* **Solution**: This limitation was noted in the report for transparency. Future work will incorporate diverse data sources, such as M-Pesa transaction records, which are widely used in Kenya and can better represent rural customers.

**6.2 Imbalanced Dataset**

* **Challenge**: The dataset is imbalanced, with some clusters having fewer customers (e.g., Cluster 2: 25 instances, Cluster 3: 149), which can lead to poor model performance on smaller groups.
* **Solution**: The F1-score was used to evaluate performance, balancing precision and recall (e.g., Cluster 2 F1-score: 0.923). Future work will apply SMOTE (Synthetic Minority Oversampling Technique), which creates synthetic data to balance cluster sizes and improve model performance.

**6.3 Ethical Concerns**

* **Challenge**: Using customer data raises privacy concerns under Kenya’s Data Protection Act (2019), and there’s a risk of discriminatory marketing, such as excluding low-income customers.
* **Solution**: An anonymized dataset was used to protect customer identities, and marketing strategies were designed to be inclusive, focusing on empowering all customer groups rather than excluding any.

**6.4 Deployment Issues**

* **Challenge**: The free tier of ngrok limits users to 3 tunnels per session, leading to the ERR\_NGROK\_324 error when exceeded.
* **Solution**: Existing tunnels were terminated using ngrok.kill() before starting a new one, ensuring the app could be deployed successfully.

**7. Future Work**

* **Incorporate Diverse Data**: Use M-Pesa transaction data to better represent rural cooperative members, reducing dataset bias and improving segment relevance.
* **Balance Dataset**: Apply SMOTE to balance the dataset, creating synthetic data for smaller clusters to enhance model performance on underrepresented groups.
* **Expand Application**: Extend the app to other Kenyan sectors, such as retail or e-commerce, to broaden its impact and support more businesses with customer segmentation.
* **Advanced Models**: Explore deep learning models for clustering and classification, potentially improving accuracy and uncovering more complex patterns in customer data.
* **Real-Time Data Updates**: Integrate real-time data updates to keep the model current, ensuring predictions remain accurate as customer behaviors change.

**8. Conclusion**

TargetWise Analytics successfully addresses the challenge of customer segmentation for Kenyan cooperatives under Coop Affairs. The project uses KMeans clustering to identify 4 distinct customer segments and a Random Forest Classifier to predict segment membership with 97.35% accuracy. The Streamlit app enables real-time predictions, providing actionable marketing insights that improve customer engagement and retention. Despite challenges like dataset bias and imbalance, the project demonstrates the potential of machine learning to enhance marketing strategies in the cooperative sector, aligning with Kenya’s Vision 2030 goals. Future improvements will focus on incorporating diverse data, balancing the dataset, and expanding the app’s application to other sectors.

**9. References**

* International Cooperative Alliance (2023). *Cooperative Statistics*. Retrieved from [ICA website].
* Kenya Union of Savings and Credit Cooperatives (KUSCCO). *Annual Report on Sacco Assets*.
* Kenya’s Data Protection Act (2019). *Kenya Gazette Supplement*.
* Kaggle Credit Card Dataset. Available at [Kaggle](https://www.kaggle.com/datasets).
* scikit-learn Documentation. *KMeans and Random Forest*. Available at [scikit-learn.org](https://scikit-learn.org).
* Streamlit Documentation. Available at [streamlit.io](https://streamlit.io).
* ngrok Documentation. Available at [ngrok.com](https://ngrok.com).

**10. Appendix**

**10.1 Screenshots**

* **Elbow Method Plot**: Shows the optimal number of clusters (k=4) for KMeans clustering.  
  [Insert screenshot from the "Clustering" page of the Colab notebook]
* **PCA Scatter Plot**: Visualizes the 4 clusters in 2D using PCA.  
  [Insert screenshot from the "Clustering" page]
* **Streamlit App Interface**: Shows the input form and fixed title of the app.py app.  
  [Insert screenshot from the "Streamlit Deployment" section]

**10.2 Sample Code Snippets**

* **Data Preprocessing**:

python

*# Drop missing values and standardize features*

df = df.dropna()

scaler = StandardScaler()

df[features] = scaler.fit\_transform(df[features])

* **KMeans Clustering**:

python

*# Apply KMeans with 4 clusters*

kmeans = KMeans(n\_clusters=4, init='k-means++', random\_state=42)

df['Cluster'] = kmeans.fit\_predict(df[features])

* **Random Forest Training**:

python

*# Train Random Forest Classifier*

rf\_model = RandomForestClassifier(n\_estimators=100, random\_state=42)

rf\_model.fit(X\_train, y\_train)

**10.3 Dataset Sample**

* The first 5 rows of the dataset are saved as sample.csv in the Colab notebook. You can download it for reference:

python

df.head().to\_csv('sample.csv', index=False)

**How to Use This Documentation**

* **For Users**: Follow the "App Usage Guide" (Section 4) to use the app for customer segmentation and marketing.
* **For Developers**: Refer to the "Setup Instructions" (Section 2) and "Methodology" (Section 3) to understand the code and replicate or modify the project.
* **For Stakeholders**: Review the "Introduction" (Section 1), "Results and Insights" (Section 5), and "Conclusion" (Section 8) to understand the project’s purpose, outcomes, and impact.