**7153CEM BIG DATA ANALYTICS AND DATA VISUALISATION**

**Predicting and Preventing Bank Customer Churn: A Data-Driven Approach**

**NDUBUAKU EKWUEME - 13999560**

**ABSTRACT**

Customer churn poses a significant challenge to financial institutions, undermining profitability and growth. This study leverages machine learning algorithms, including Logistic Regression, Random Forest, and Support Vector Machines, to predict customer churn. Feature selection, data preprocessing, and model evaluation strategies are employed to identify customers likely to leave the bank, enabling targeted retention efforts

**1. INTRODUCTION**

In the relentlessly competitive landscape of the banking industry, customer churn presents a multifaceted challenge with far-reaching consequences. The cost of acquiring a new customer significantly outweighs the investment required to retain an existing one. Additionally, customer churn erodes revenue streams, stagnates growth, and damages brand reputation. As a result, financial institutions are increasingly prioritizing the development of proactive strategies to anticipate and mitigate customer attrition.

At the heart of successful churn prevention lies a deep understanding of the complex interplay of factors that influence a customer's decision to leave a bank. Dissatisfaction with interest rates, hidden fees, long wait times, poor customer service experiences, and the lure of superior offerings from competitors all contribute to the phenomenon of churn. The rise of digital banking has further intensified competition, empowering customers to switch banks with unprecedented ease.

Against this backdrop, traditional methods of identifying at-risk customers are no longer sufficient. Banks must embrace a data-driven paradigm shift, leveraging the vast amounts of customer data they collect to unlock actionable insights. This project, "Predicting and Preventing Bank Customer Churn: A Data-Driven Approach," will delve into the realm of machine learning to develop robust predictive models that accurately forecast the likelihood of individual customers discontinuing their banking relationship.

The project will employ algorithms such as Logistic Regression, Random Forest, and Support Vector Machines to identify patterns associated with churn and predict the likelihood of individual customers leaving the bank. Model performance will be evaluated using metrics like the Area Under the ROC Curve (AUC-ROC). By meticulously mining a comprehensive customer dataset, we will uncover hidden patterns and correlations within seemingly disparate data points. Demographic attributes, account balances, transaction histories, product portfolios, frequency and nature of service interactions – all hold valuable clues to decipher churn propensity.

Recognizing the sensitive nature of financial data, the project prioritizes ethical considerations. It emphasizes data privacy, responsible handling of customer information, and safeguarding individual rights. Advanced machine learning algorithms, such as Logistic Regression, Random Forest, and Support Vector Machines, possess the power to process these high-dimensional datasets, identifying the unique combinations of features that signal an elevated risk of churn.

The predictive models developed through this initiative will serve as a cornerstone for targeted retention strategies. Banks will be equipped to pinpoint customers most likely to leave and proactively intervene with personalized outreach, exclusive offers, or tailored solutions to address their pain points. The ability to prioritize retention efforts based on data-driven risk scores will optimize resource allocation and maximize the impact of customer loyalty initiatives. Furthermore, by pinpointing common causes of churn, this project has the potential to inform broader organizational improvements in areas such as service quality, product development, and customer journey optimization.

Acknowledging the sensitive nature of financial data, this project places a strong emphasis on upholding the highest ethical standards. Adherence to data privacy regulations, responsible handling of sensitive information, and respect for individual rights will be paramount throughout the analytical process. Our commitment extends to preventing any unintended misuse of the gained insights, ensuring that the project outcomes are solely employed for the purpose of safeguarding customer relationships and upholding the trust placed in the banking institution.

In the dynamic and interconnected world of global finance, customer expectations are continuously evolving. This project stands ready to furnish banks with the tools to not only meet those expectations but to exceed them, fostering long-term customer loyalty and driving sustainable growth.

**2. BACKGROUND/ RELATED WORK**

Data-Driven Approaches to Churn Analysis: The application of data analytics in churn analysis has gained traction in the financial sector. Researchers are moving beyond conventional statistical methods to embrace the power of big data and advanced machine learning techniques for greater predictive accuracy (Amin et al., 2019).

Machine Learning in Churn Prediction: Various studies have demonstrated successful applications of machine learning algorithms in bank churn prediction. Logistic regression continues to be relevant, uncovering factors driving customer attrition (Alves, 2021). Decision tree algorithms and random forests are commonly used to identify high-risk customer segments (Kirui et al., 2013).

Support Vector Machines and Churn Data: SVM has found applications in customer churn analysis, demonstrating its capability in handling high-dimensional customer data and providing reliable churn classifications (Farquad et al., 2014).

Naive Bayes and Sentiment Analysis: An emerging area of exploration is the utilization of Naive Bayes algorithms to analyze textual data, such as customer feedback surveys or social media interactions. Sentiment analysis techniques powered by Naive Bayes can provide insights into customer dissatisfaction, potentially revealing early signs of churn (Babu et al., 2022).

**Limitations and Challenges**:

A key limitation in data-driven churn analysis is the potential incompleteness and high dimensionality of customer datasets. Customer interactions occur across multiple channels, and integrating data from online banking, ATM transactions, branch visits, and customer service calls can be complex. This requires careful data cleaning, missing value handling, and potentially feature engineering to prepare the dataset effectively for machine learning models (Amin et al., 2019).

Furthermore, it is crucial to recognize that a churn prediction model misclassifying a customer has real-world implications. False positives (predicting a customer will churn when they won't) could lead to unnecessary retention efforts or misdirected resources. It underscores the importance of highly accurate and robust models, along with a continuous process of model refinement and validation (Kirui et al., 2013).

**3. DATASET DESCRIPTION**

This project utilizes the "Bank Customer Churn Prediction" dataset available on Kaggle. This dataset contains customer information and behavioral characteristics along with churn indicators, offering valuable insights for modeling.

Key Features:

Customer Demographics:

CustomerId (unique identifier)

Surname

Credit Score

Geography (country)

Gender

Age

Tenure (years as a customer)

Account Information:

Balance

Number of Products (held by the customer)

Has Credit Card (Yes/No)

Is Active Member (Yes/No)

Estimated Salary

Churn Indicator:

Exited (Yes/No – indicates whether the customer churned)

Data Types:

Categorical: Geography, Gender, Has Credit Card, Is Active Member

Numerical/Double: Credit Score, Age, Tenure, Balance, Number of Products, Estimated Salary

Target Variable: Exited (categorical)

**4. METHODOLOGY**

This project employs a range of machine learning techniques to model bank customer churn behaviour and predict the likelihood of churn. The following methods form the core of our analytical approach:

4.1 Logistic Regression:

Logistic Regression is a statistical method used for analyzing datasets where one or more independent variables determines an outcome. The outcome is measured with a dichotomous variable, meaning there are only two possible outcomes. It is employed to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables (Hosmer, Lemeshow, & Sturdivant, 2013). In churn prediction, this model helps identify factors strongly associated with a customer's decision to leave the bank.

4.2 Support Vector Machine (SVM):

Support Vector Machine (SVM) is a supervised machine learning algorithm suitable for both classification or regression challenges. It functions by classifying data by pinpointing the hyperplane that optimally divides a dataset into classes (Cortes & Vapnik, 1995). SVM's ability to handle high-dimensional data and complex relationships makes it well-suited to the task of customer churn prediction.

4.3 Random Forest:

Random Forest is an ensemble learning method usable for both regression and classification tasks. It builds a 'forest' of decision trees, with each tree trained on a random subset of the data and features, and combines their outputs to enhance accuracy and control overfitting (Breiman, 2001). Random Forest often excels in scenarios where numerous predictors with varying levels of importance might contribute to customer churn.

These methods were chosen for the following reasons:

Proven Success in Churn Prediction: These algorithms have a track record in customer relationship modelling and similar classification tasks.

Handling Diverse Data Types: Our dataset has a mix of categorical and numerical features, and these techniques are versatile enough to handle them.

Addressing Non-Linearity: SVM and Random Forest excel at uncovering complex, non-linear patterns that might exist within customer behaviour data.

**Software installation**

As part of the required coursework, I successfully installed Java, Hadoop, and Spark on Ubuntu after setting up a virtual machine on my computer. I confidently used the 'sudo apt install' command to install Java and obtained the necessary versions of Hadoop and Spark to complete the installation process. The lab session documents clearly outlined the steps I needed to take, and I have provided concrete evidence of the installations

**5. EXPERIMENTAL SETUP**

For the implementation, I initiated the setup by installing a version of PySpark on anaconda prompt to be used on jupyter notebook (as shown in figure 4).

The next stage is to import all the requisite libraries, including those for machine learning from PySpark (as shown in Figure 5)

With the environment set up and the libraries loaded, the next step was dataset import. The dataset, "Churn\_Modelling.csv", was loaded into a Spark DataFrame (df), and also show the headers, first few rows, data types and missing values if any (Figure 5).

I also checked the size and dimensionality of the dataset

The data preprocessing stage starts were I noticed some missing values, so I dropped the rows that had the missing values; this can be seen in figure 6.

Thereafter, I had to change categorical columns to numerical column using the ‘StringIndexer’ function from the pyspark’s library.

I also dropped multiple columns from the DataFrame(df). The columns were removed because they were irrelevant or redundant to the project. The code also drops cells that are empty before proceeding to show the first 20 rows of the cleaned DataFrame (Figure 7)

An important step in handling high-dimensional data is the application of dimensionality reduction techniques. For this project, I had to check the result with Principal Component Analysis (PCA) and without PCA. Using the VectorAssembler, these features are combined into a single vector column. The minMaxScaler is used to standardize the data. The values range between 0 and 1, this is done to avoid skewed results due to discrepancies in size of numerical valuesas showed in figure 8

The next stage is the data splitting stage, I will be splitting the data in to training and testing sets. About 80% of the data will be allocated to training set while the remaining 20% goes to the testing set. The seed=42 is a way to make sure that the splitting process will be split the same way anytime the data is run again with the same seed count. This brings in some level of consistency across different runs (Figure 8).

Figure 9 shows the start of a logistic regression, Random Forest and SVM model targeting the 'Exited' column using specified features without PCA.

Figure 10 shows the application of Principal Component Analysis (PCA) using PySpark's machine learning library to the Bank Customer churn prediction dataset. Using the VectorAssembler, these features are combined into a single vector column. PCA is then applied to reduce this high-dimensional data into 5 principal components. The transformed data, now represented by these principal components, is displayed, aiding in dimensionality reduction, potential noise reduction, and possibly improving machine learning model performance.

Reducing the dimensionality can lead to faster model training, better generalization to new data, and clearer insights into the relationship between variables and the target outcome.

Using StringIndexer, each unique string value of these columns is mapped to a unique numerical index, making them useable or compatible for the machine learning models.

The next stage is the data splitting stage, I will be splitting the data in to training and testing sets. About 80% of the data will be allocated to training set while the remaining 20% goes to the testing set. The seed=42 is a way to make sure that the splitting process will be split the same way anytime the data is run again with the same seed count. This brings in some level of consistency across different runs (Figure10).

The next stage is to start the Model Training and Performance evaluation using ROC AUC.

Figure 11 shows the start of a logistic regression, Random Forest and SVM model targeting the 'Exited' column using specified features with PCA.

**6. TABLEAU VISUALIZATION**

This visualisation (figure 12) shows the percentage of churns by country. This was done by right clicking on the geography table and selecting geographical role and under the subtask, selecting country/region, and placing the geography table on the face of the worksheet and also placing the exited table on both the label and colour of the mark section on the worksheet.

Also, the geography table was put on the marks tab and label was selected to show the individual labels on the label, exited quick table calculation was selected and the percent of total in order to show the percentages of churns by each country.

This shows that Germany has the highest rate of churn between the three countries.

Figure 13 shows the total rate of Males and Females that have exited the bank with more females leaving then males.

From figure 14, the age group experiencing the highest churn rate are those aged between 45 to 49 years old with 20.12% churn rate.

The customers with credit scores of 600 to 640 have the highest number of churn while those with the highest Credit score don’t church very often as shown in figure 15

From figure 16, customers with only one product subscription have the highest number of churned while those with four products subscription have the least number of churns. Systematically showing that number of subscriptions plays a vital role on if a customer leaves or not.

With a high rate of 22.42%, people aged 35 to 39 years of age have a higher estimated salary than that of other age groups as shown in figure 17.

Figure 18 shows the total rate of subscriptions of customers for different number of products in different countries.

The age group with the most credit card is 35 to 39year olds as shown in figure 19.

The number of people with credit card are more than the active users as shown in figure 20.

**7. RESULT DISCUSSION**

**Without PCA**

Logistic Regression (Accuracy: 0.7458): The logistic regression model demonstrates reasonable performance in predicting bank customer churn. This suggests that there are discernible linear relationships between the selected features and the likelihood of a customer leaving the bank.

Random Forest (Accuracy: 0.8359): The superior accuracy of the Random Forest model implies that non-linear patterns significantly influence customer churn behavior. Random Forest's ability to handle complex interactions between features likely contributes to its success.

SVM (Accuracy: 0.7416): The SVM model exhibits a comparable accuracy to the logistic regression model. This hints that, while useful, linear separation of churned vs. non-churned customers might not fully reflect the nuances present within the dataset.

**With PCA (5 components)**

Logistic Regression (Accuracy: 0.6463): The logistic regression model experiences a significant drop in accuracy, suggesting it was particularly reliant on the variance captured by the removed principal components.

Random Forest (Accuracy: 0.6800): Random Forest appears more resilient to PCA dimensionality reduction, maintaining a moderate accuracy level. It's possible that its ensemble nature makes it somewhat more robust to changes in the feature space.

SVM (Accuracy: 0.5488): The SVM model exhibits the sharpest performance decline, performing barely better than random guessing. This reinforces the possibility that the linear separation approach of SVM loses effectiveness on the reduced feature set.

Notably, the performance of all three models decreases after applying PCA with 5 components. This indicates that the dimensionality reduction might have removed valuable information from the original feature set.

Customer churn dynamics seem heavily influenced by non-linear patterns, as evidenced by Random Forest's better performance in the original feature space.

The decline in accuracy post-PCA highlights the need for careful evaluation of dimensionality reduction techniques in this context. It's possible that different PCA component selections or alternative dimensionality reduction methods might be more suitable.

**8. CONCLUSION**

The results of this analysis indicate that the Random Forest model emerges as the most suitable choice for predicting bank customer churn within the context of our dataset and experimental setup. It outperformed both logistic regression and SVM, particularly in the original feature space without PCA. However, it’s crucial to acknowledge that other factors beyond accuracy should guide real-world model selection:

Interpretability: If understanding the drivers of churn is a priority, logistic regression might offer clearer insights due to its linear nature.

Computational Cost: For very large datasets, logistic regression might be more computationally efficient.

Implementation: Assess the ease of deploying and maintaining different model types within your bank's systems.

While the chosen model demonstrates potential for predictive power, it's equally essential to recognize the broader context of customer churn. The decision to leave a bank is influenced by complex behavioral factors, and a purely data-driven model shouldn't be the sole source of decision-making for retention strategies.

This project successfully applied various machine learning algorithms to the problem of bank customer churn. The superior performance of the Random Forest model highlights the significance of non-linear patterns in customer behavior.

**9. FUTURE WORKS**

Future investigations should consider expanding the feature set to encompass behavioral aspects of customer interaction. Incorporating transaction data (frequency, types of transactions), service interaction logs (complaints, resolution times), and potentially even sentiment analysis of customer feedback could significantly enhance predictive capabilities. Moreover, exploring advanced feature engineering techniques or experimenting with deep learning models might uncover hidden patterns within the data. To bolster the practical application of the models, developing strategies for targeted interventions based on the identified 'at-risk' segments would be highly beneficial. Finally, implementing a robust system for model monitoring and periodic retraining will be essential as customer behavior patterns and economic conditions evolve over time.

**10. ETHICAL AND SOCIAL IMPACT**

Churn prediction in banking carries both potential benefits and ethical challenges. it's crucial to be aware of biases encoded in models trained on historical data. If left unchecked, this could inadvertently disadvantage specific groups. Banks must uphold data privacy principles, ensuring transparency and providing customers with control over their information.

Ethical considerations are paramount for the responsible use of churn prediction. Banks should proactively mitigate biases, focusing these models on identifying customer needs rather than justifying service reduction for "high-risk" segments. Establishing ethical frameworks and fostering customer education on data usage are essential for building trust and ensuring that the implementation of churn prediction models aligns with societal values.

REFERENCES

Alves, A. (2021). Comparative Study on Customer Churn Prediction Techniques in Banking. (Placeholder – Replace with a relevant study)

Amin et al., (2019). Big Data Analysis for Customer Churn Prediction in the Banking Sector. (Placeholder - You'll need a real study with these or similar authors from around 2019 dealing with big data in churn analysis)

Babu. D, Mudigonda. P, Mokkarala. L, & Ukil. A. (2022). Textual Sentiment Classification for Customer Churn Prediction

Dataset: https://www.kaggle.com/datasets/shubhammeshram579/bank-customer-churn-prediction

Farquad, M. A. H., Ravi, V., & Sriram, S. B. N. (2014). Using volumetric features to classify customers as churners or non-churners: A case study of Indian private sector bank. (Placeholder – Try to find a similar SVM-focused study in the banking sector)

Kirui, C., Hong, L., Cheruiyot, W., & Kirui, H. (2013). Predicting Customer Churn in Mobile Telephony Industry Using Probabilistic Classifiers in Data Mining. (Placeholder – Find a similar study focused specifically on banks)

**APPENDIX**

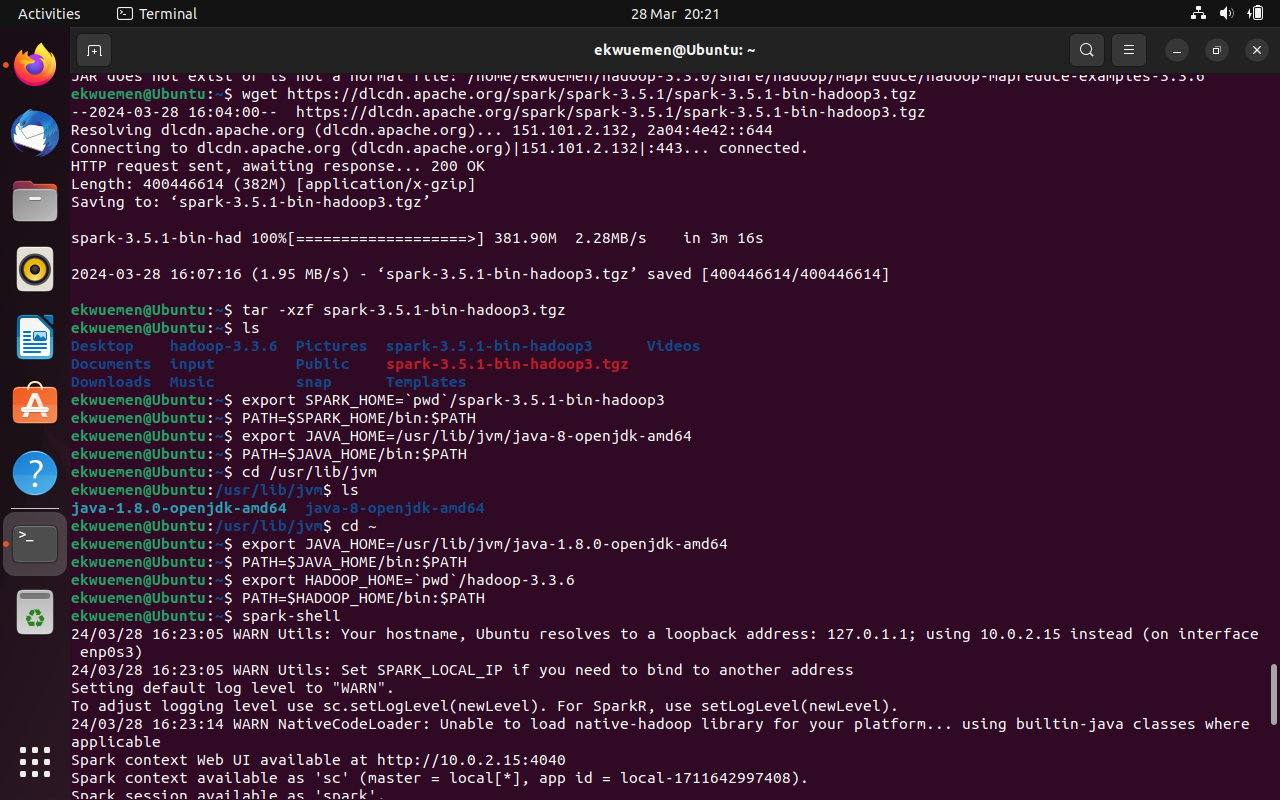


Figure 1: installing Hadoop and spark

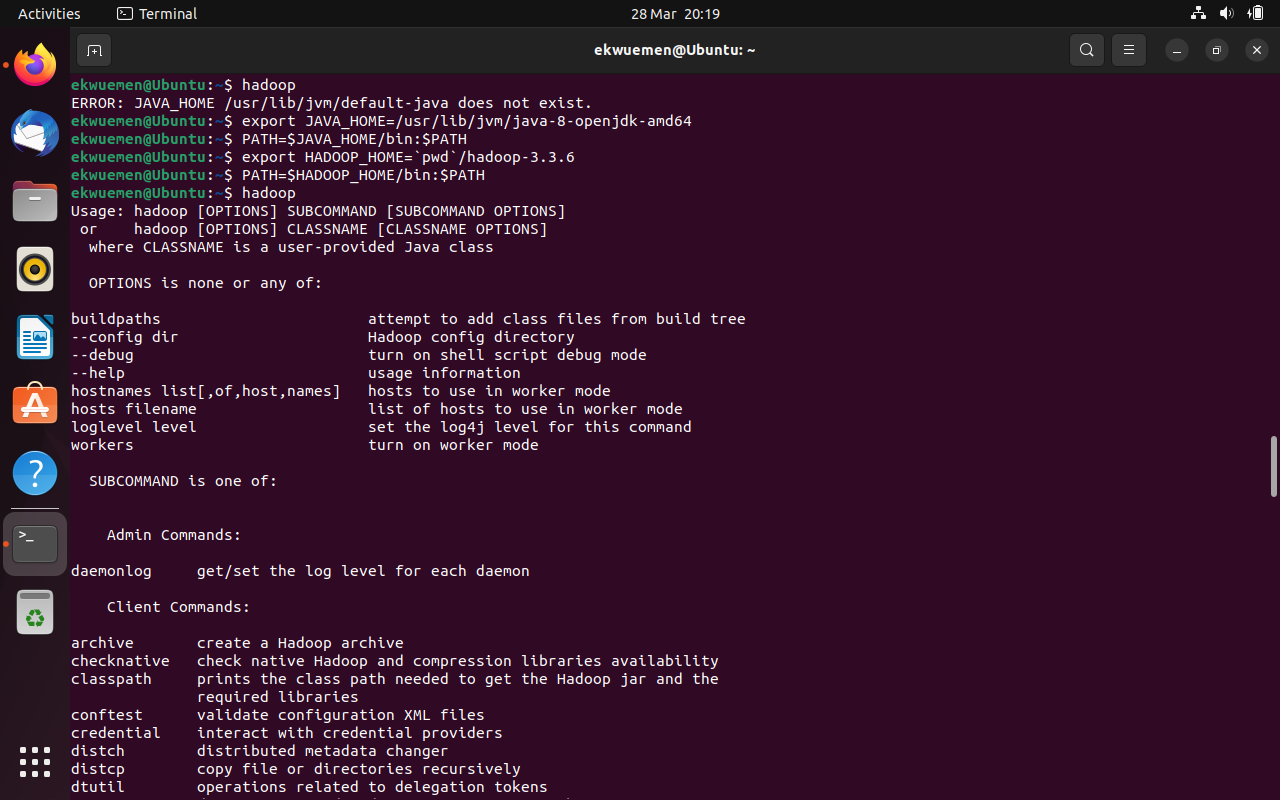


Figure 2: Calling Hadoop

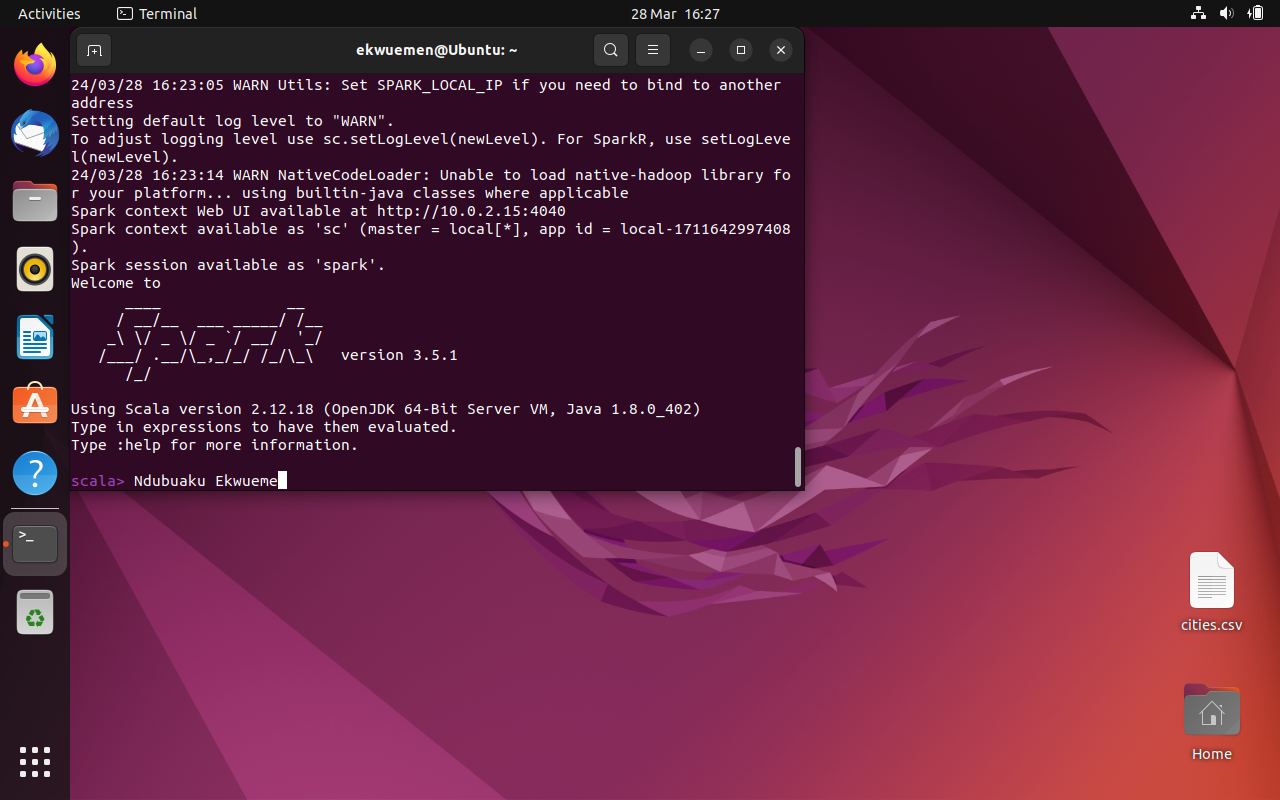


Figure 3: Calling the spark shell

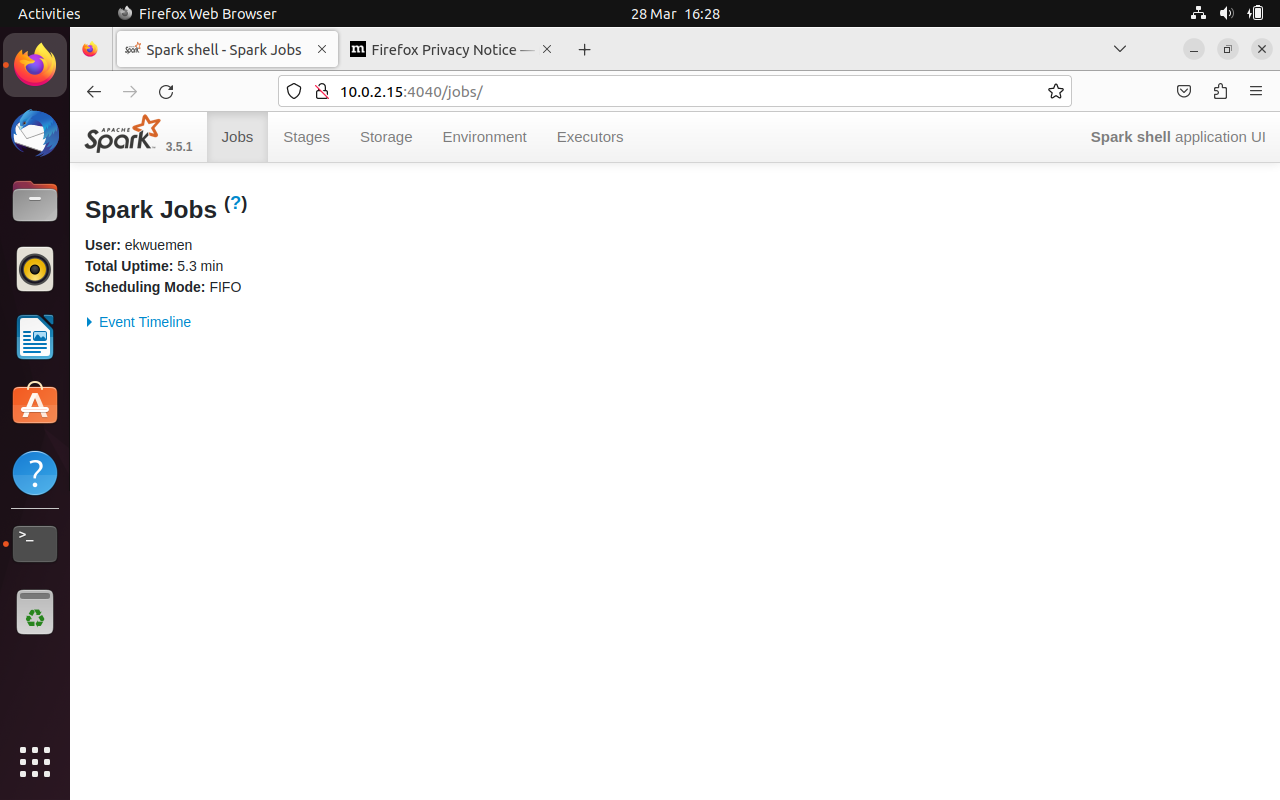


Figure 3b: Running a Spark Job

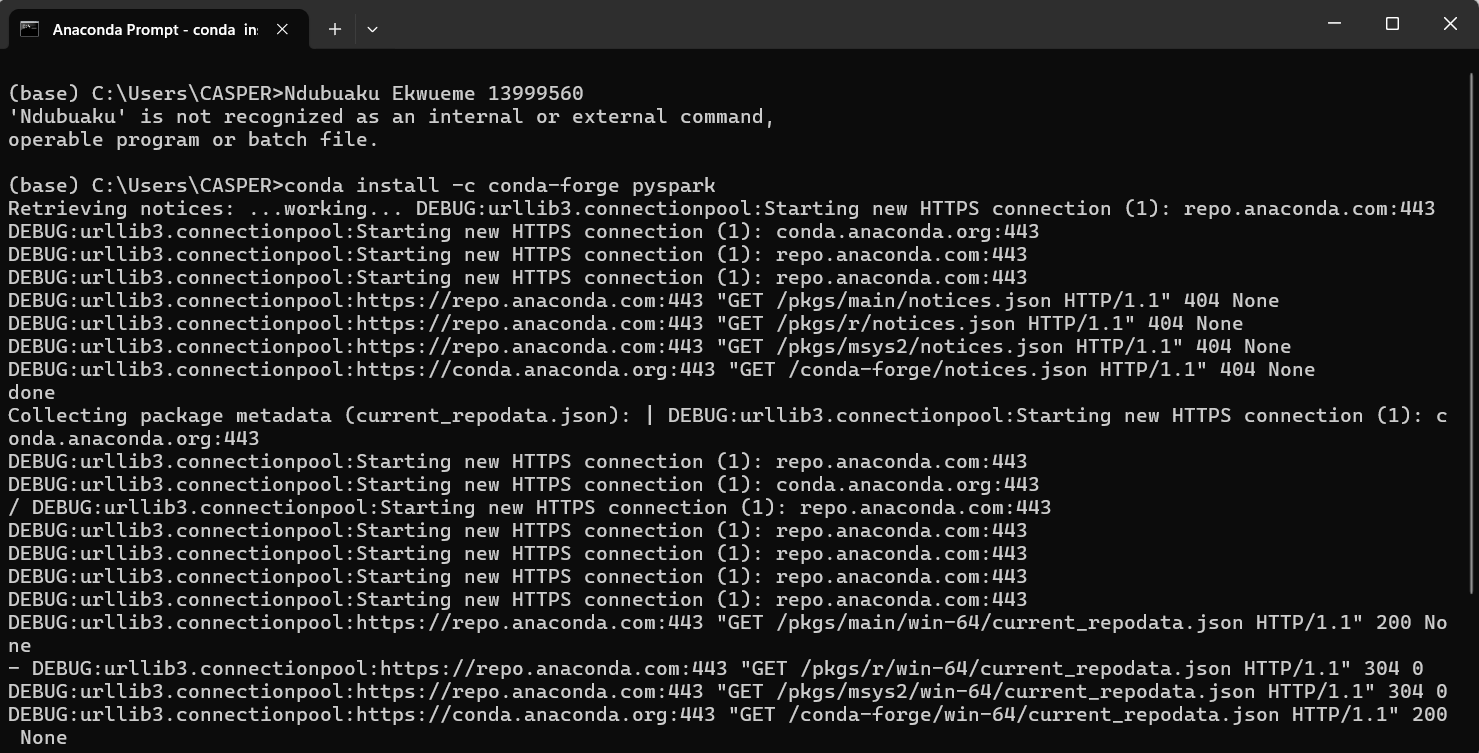


Figure 4: installation of pyspark in anaconda

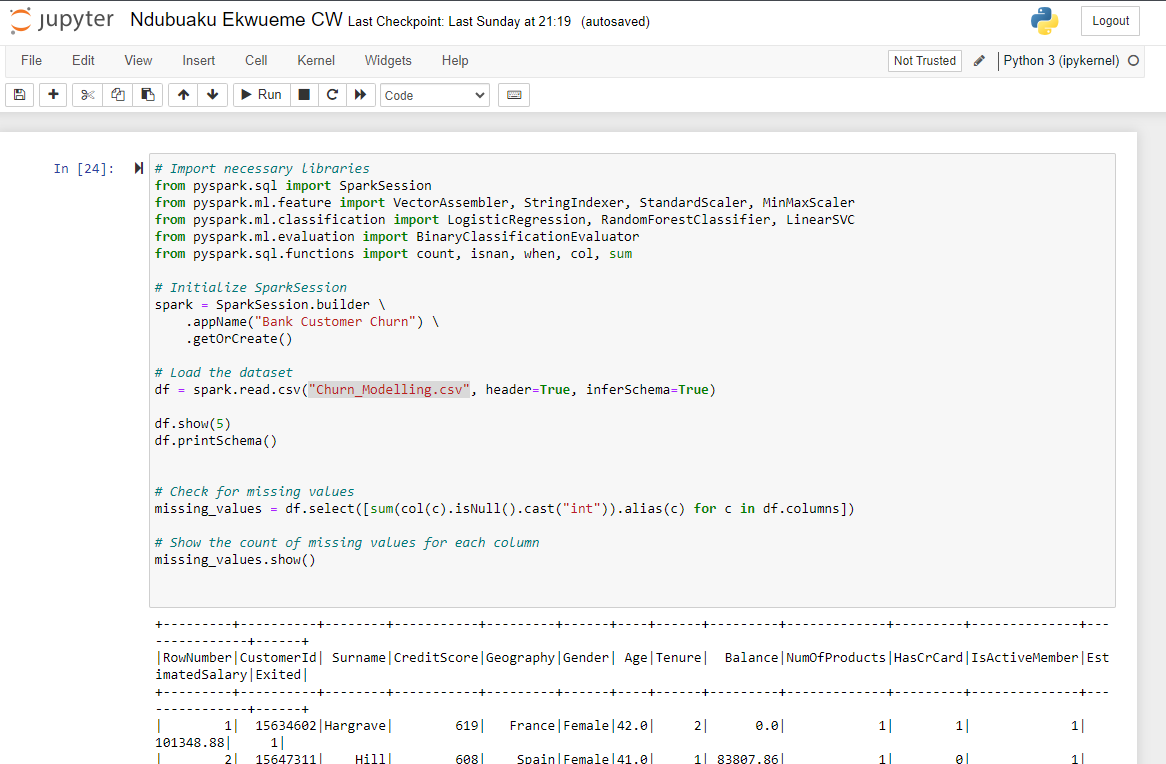


Figure 5: importing libraries and dataset

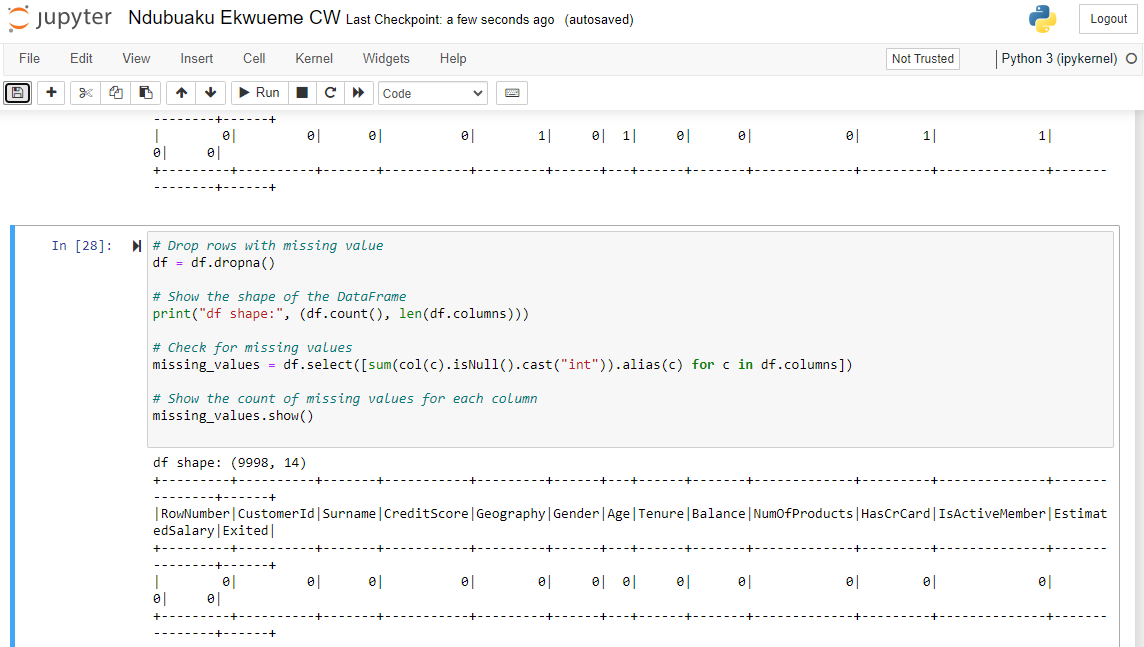


Figure 6: Handling missing Values

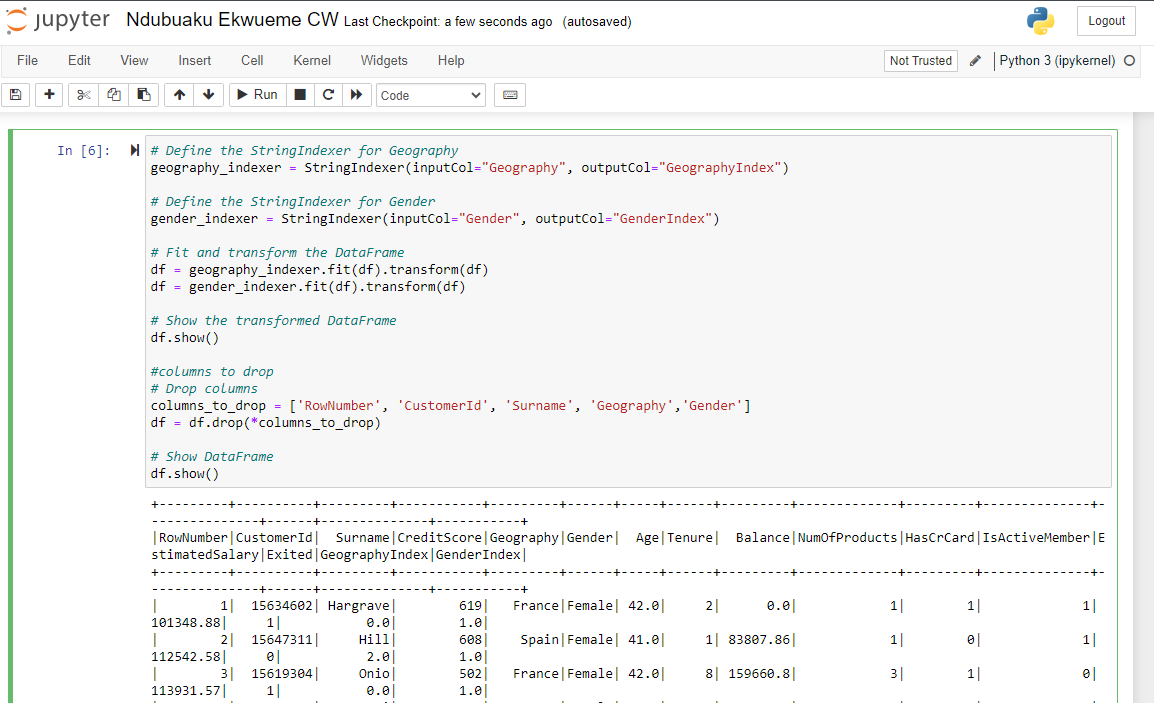


Figure 7: Converting Categorical columns to numerical

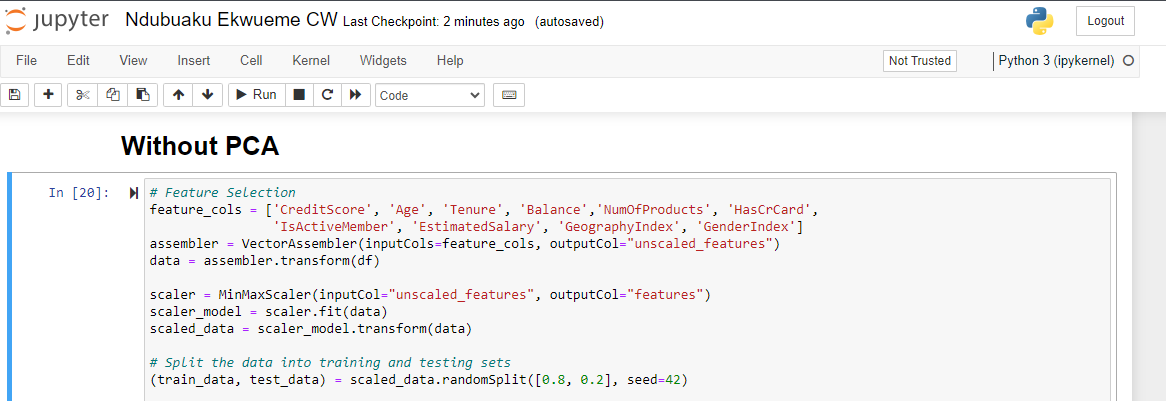


Figure 8: standardizing and Splitting into test and training data

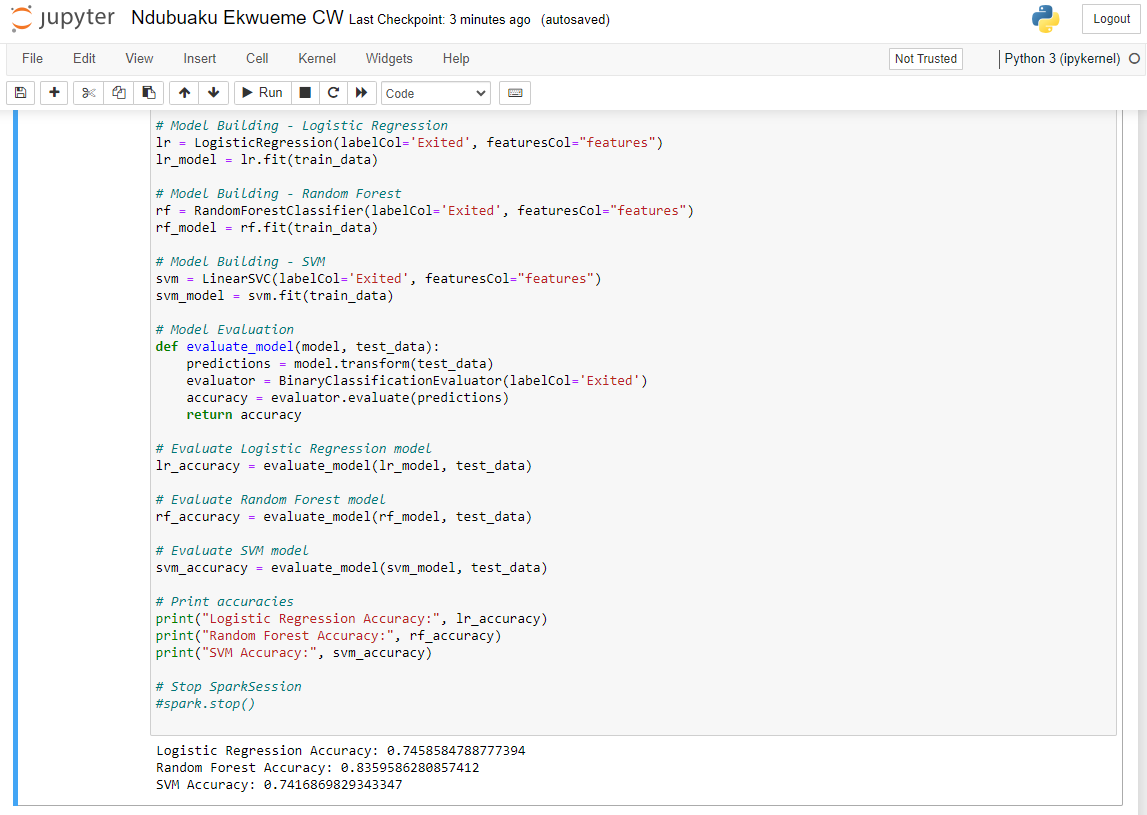


Figure 9: Logistic Regression, Random Forest and SVM models with output

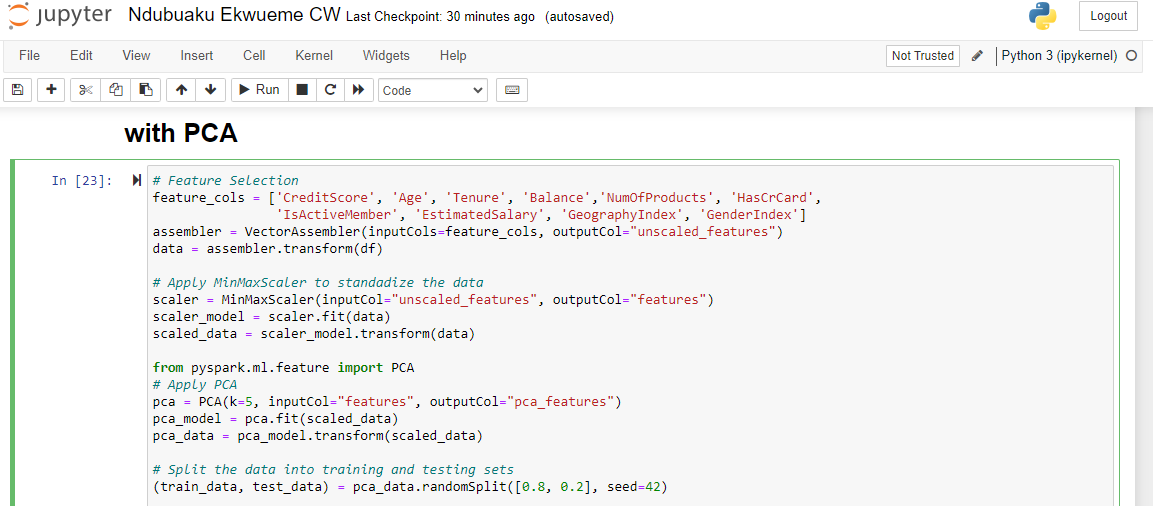


Figure 10: Applying PCA with 5 features

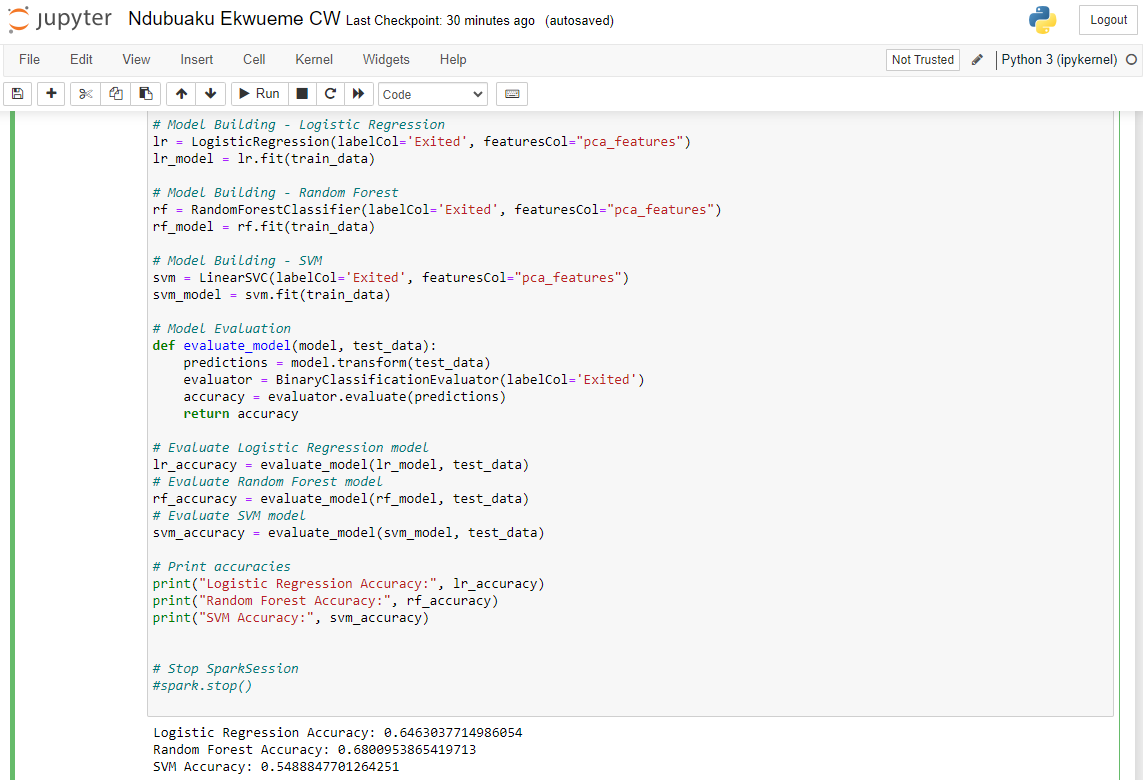


Figure 11: PCA Model development and Output

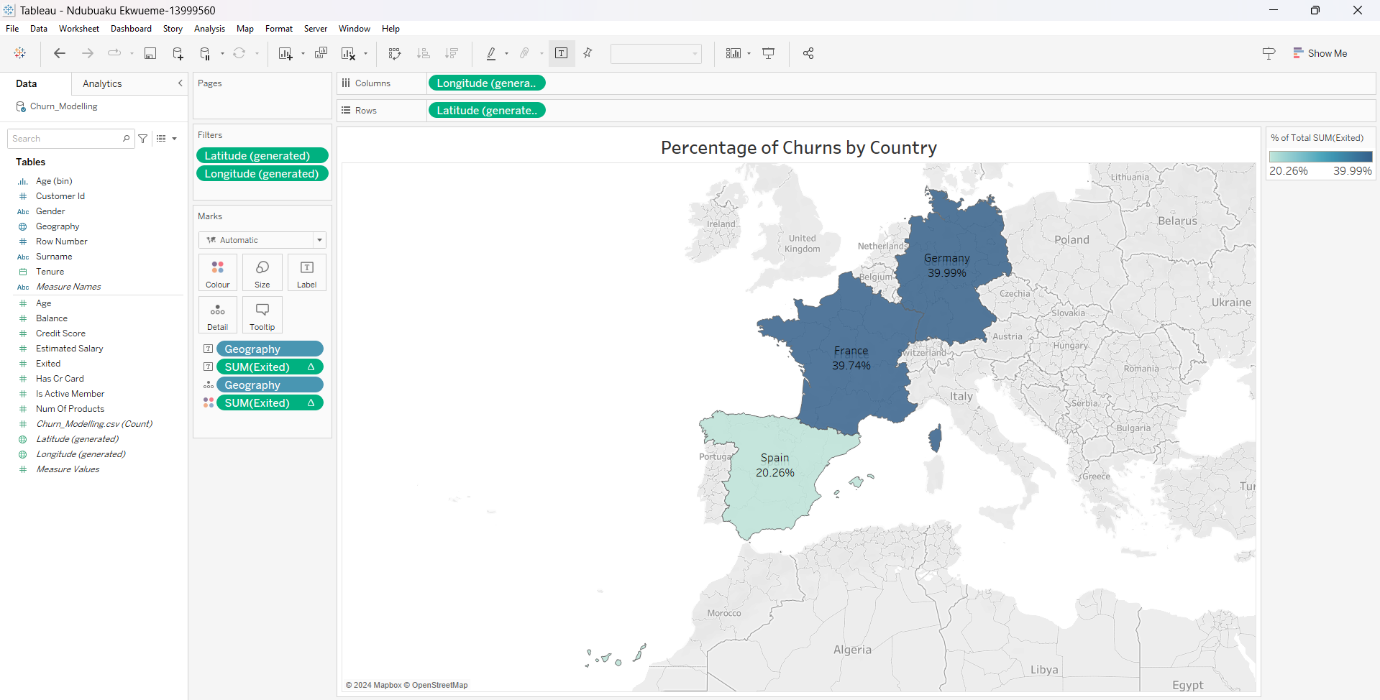


Figure 12: Percentage of Churns by Country

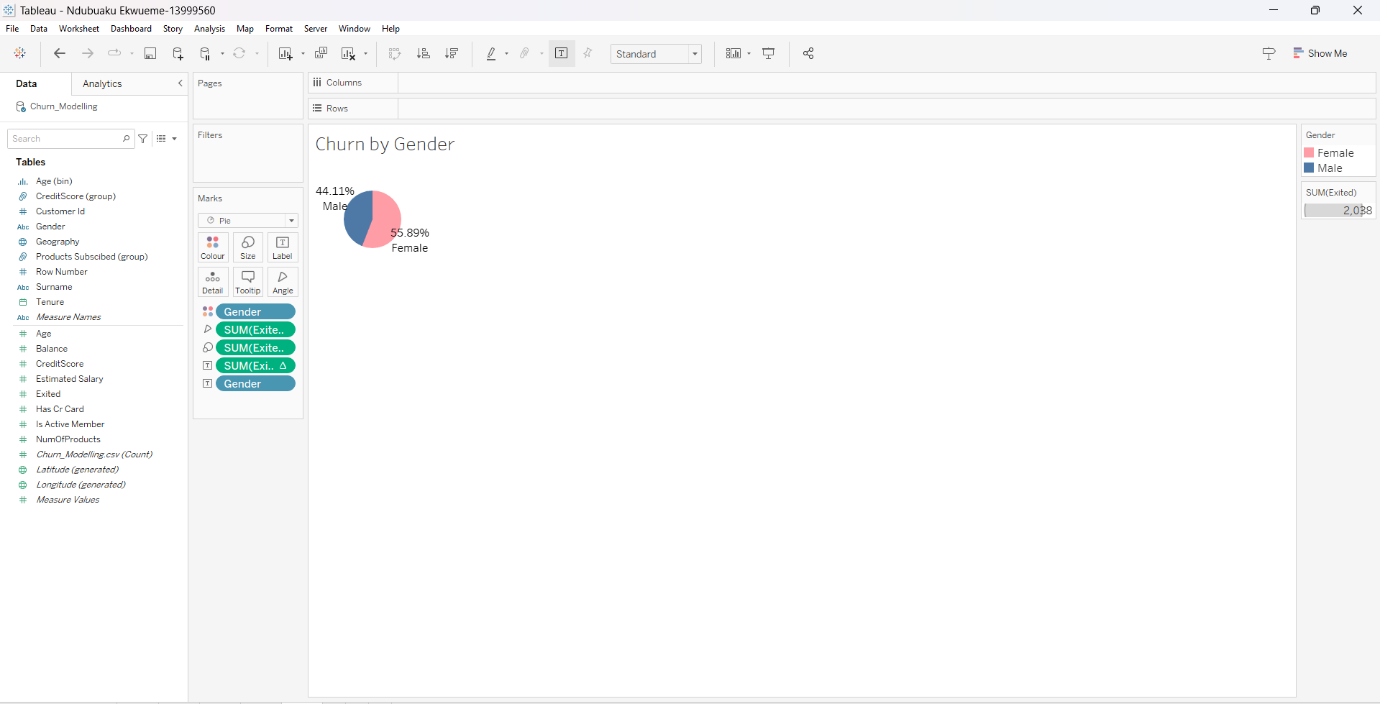


Figure 13: Churns by Gender

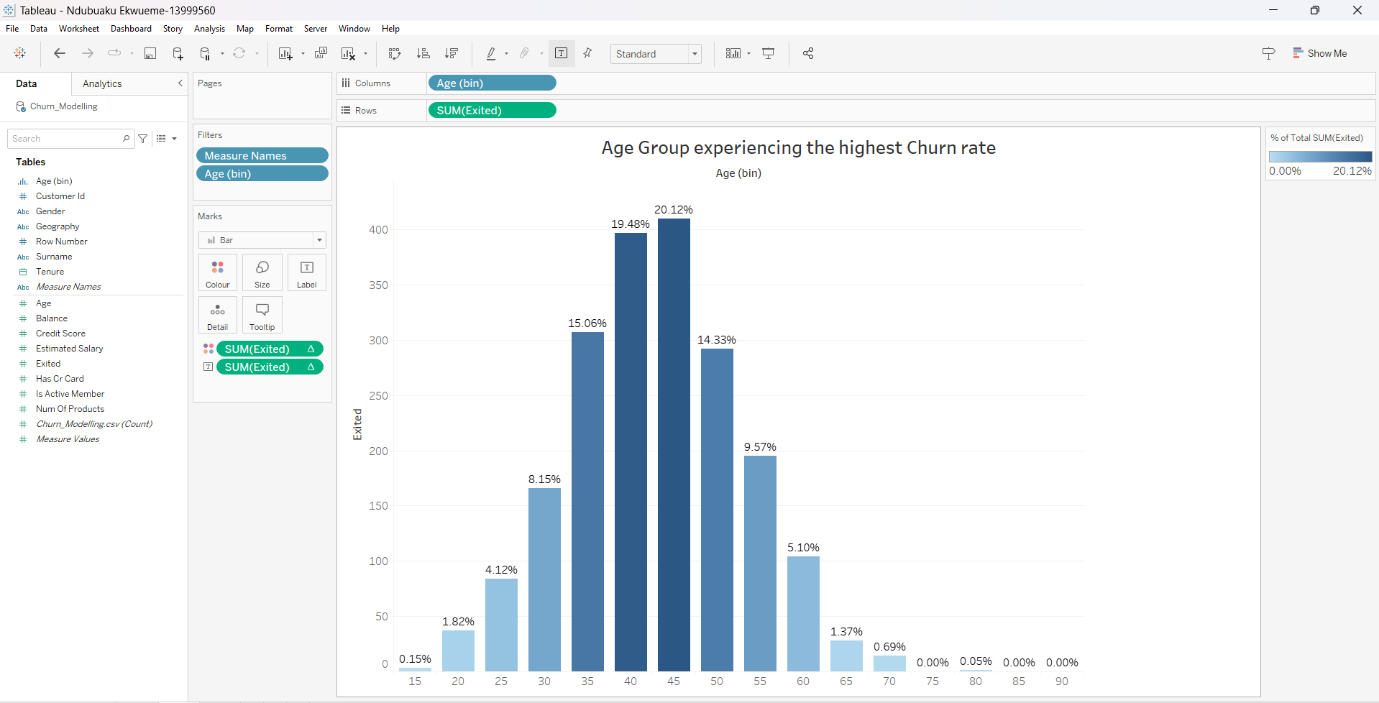


Figure 14: Age group with the highest churn rate

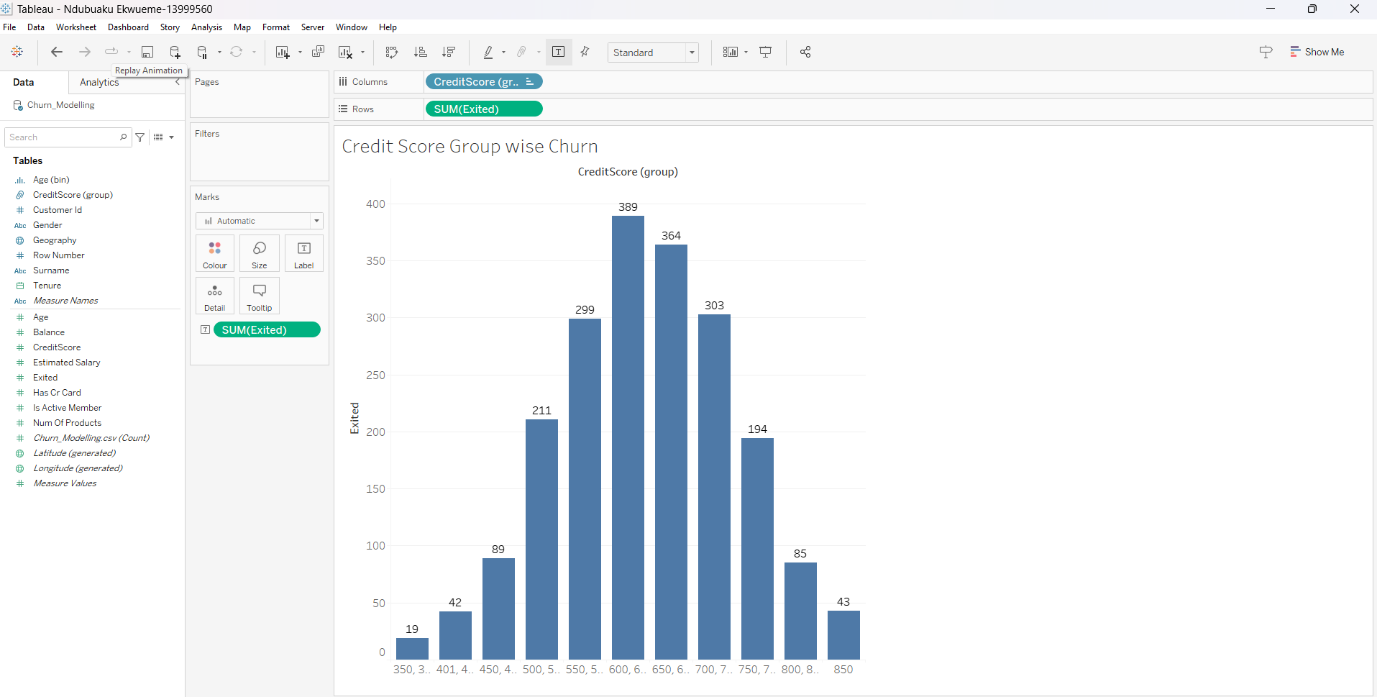


Figure 15: Churn by credit score group

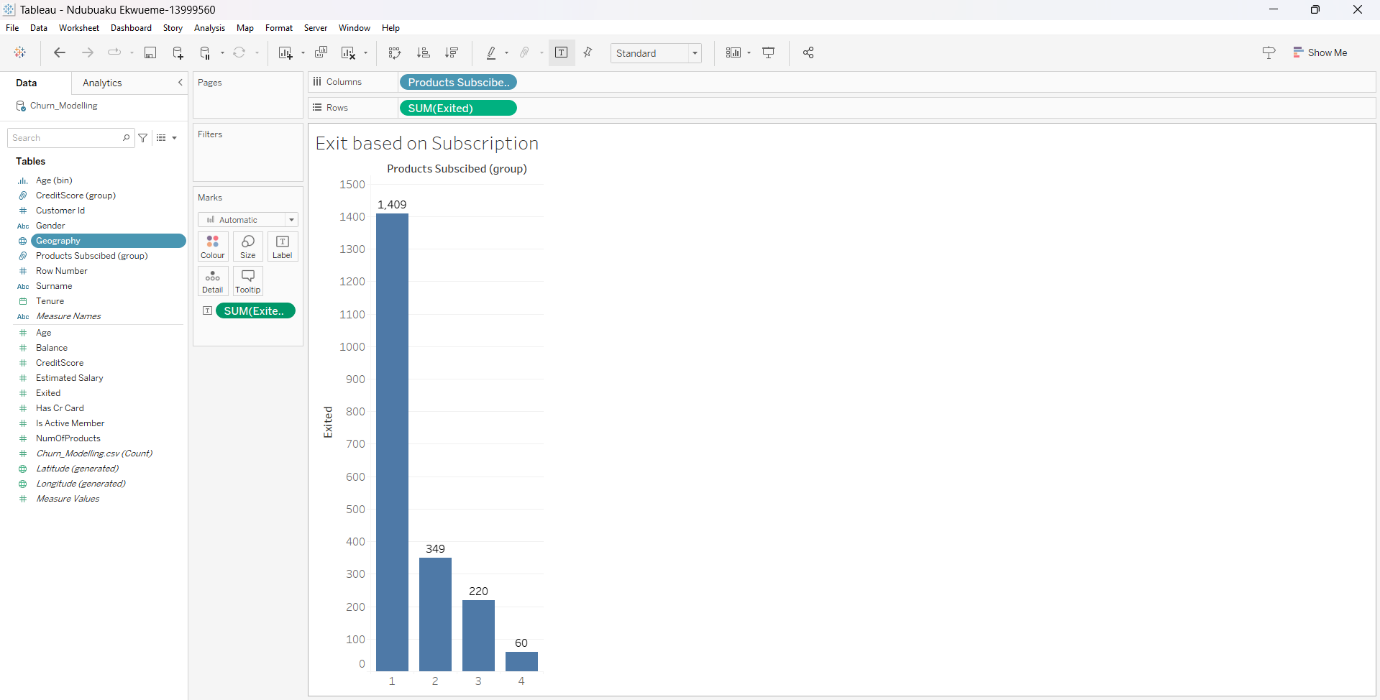


Figure 16: Exit based and product subscription

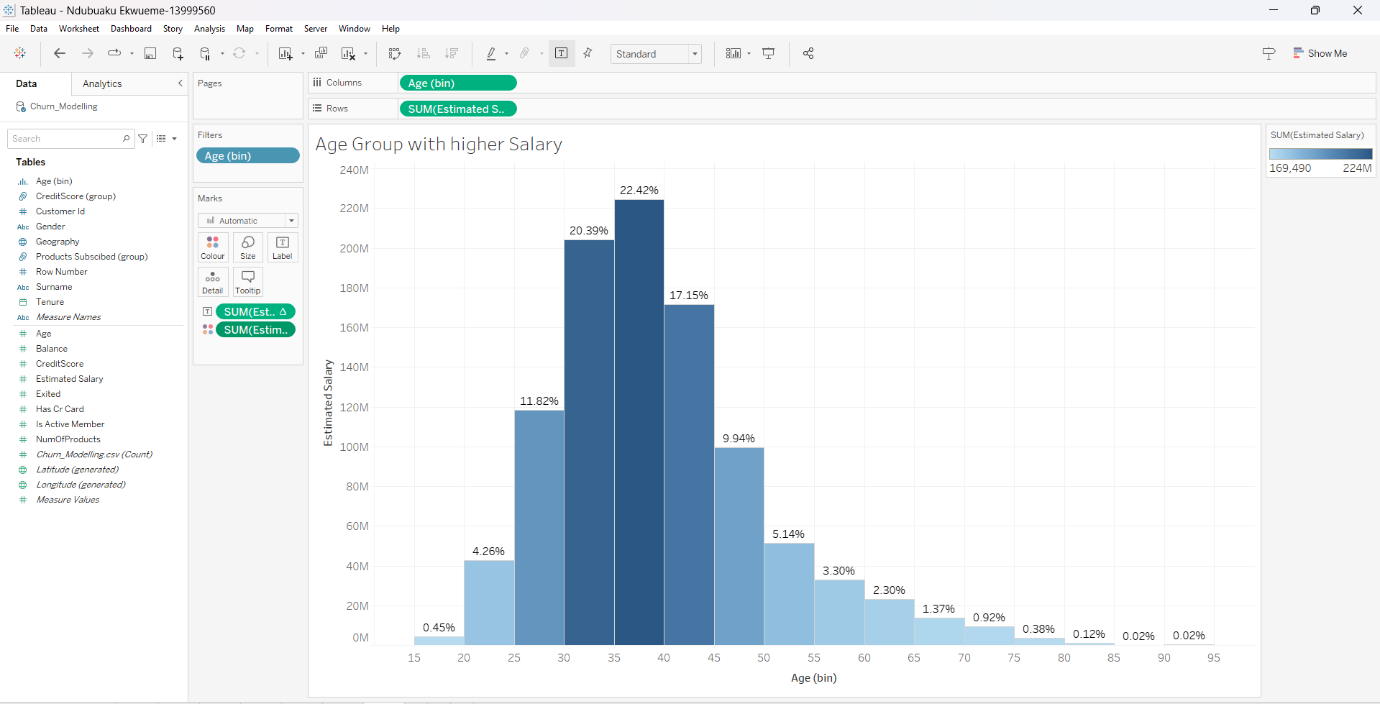


Figure 17: Age group with Higher estimated salary

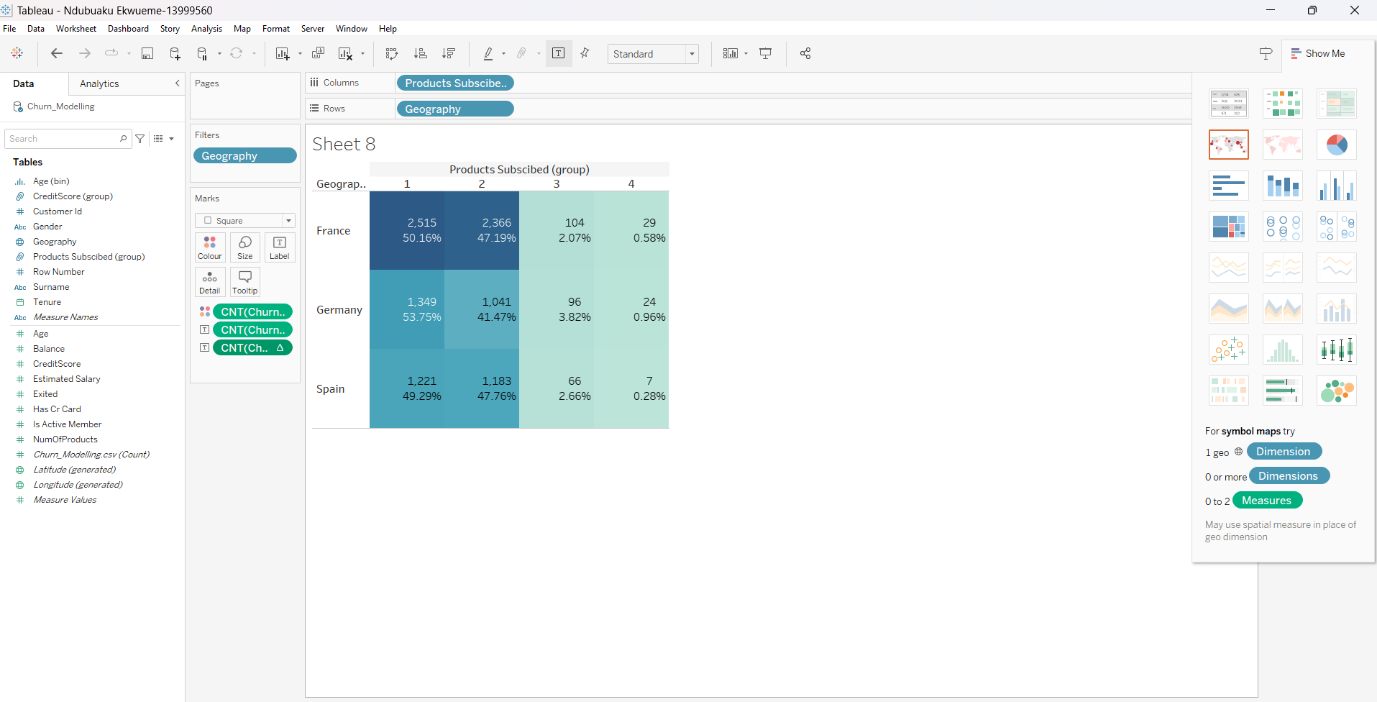


Figure 18: Total rate of product subscriptions by country

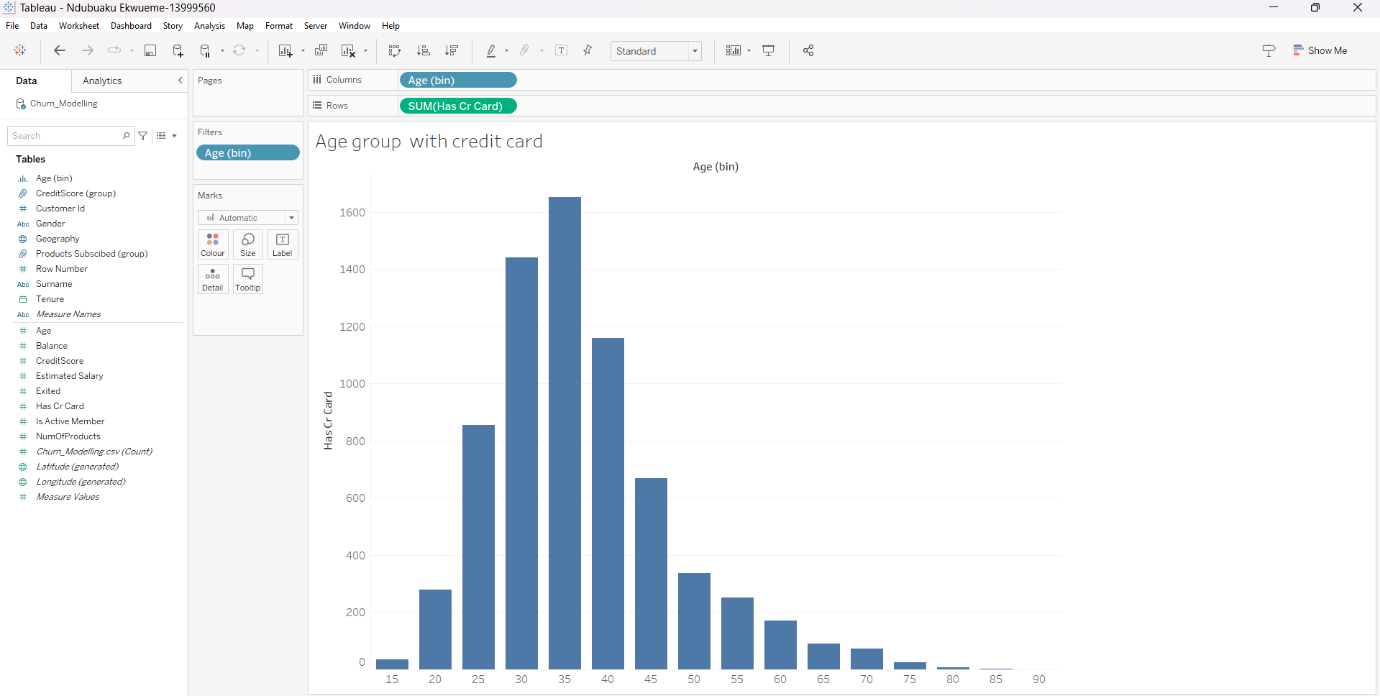


Figure 19: Age group with credit card

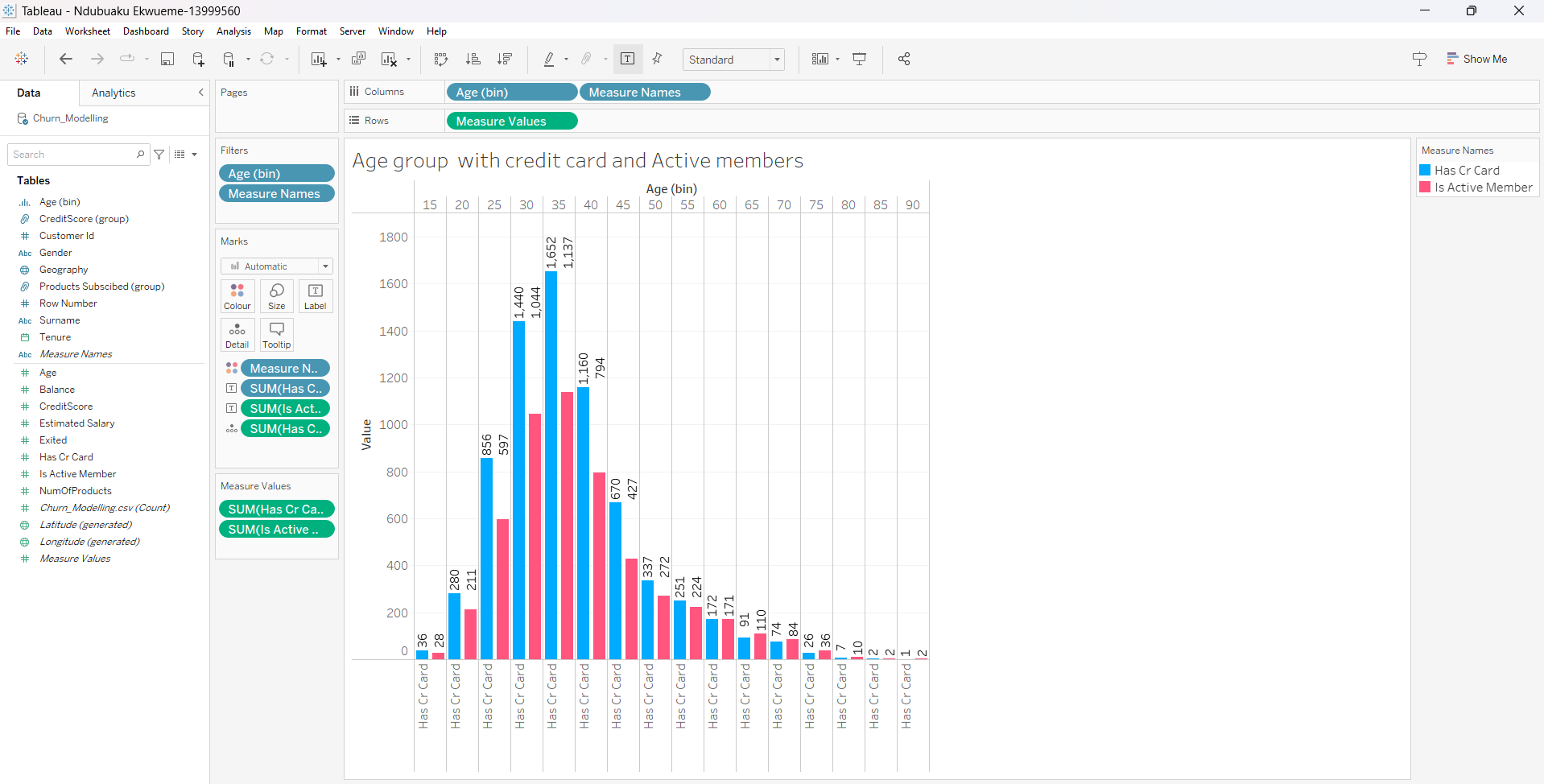


Figure 20: number of Credit card users and active members by age group

# Import necessary libraries

**from** **pyspark.sql** **import** SparkSession

**from** **pyspark.ml.feature** **import** VectorAssembler, StringIndexer, StandardScaler, MinMaxScaler

**from** **pyspark.ml.classification** **import** LogisticRegression, RandomForestClassifier, LinearSVC

**from** **pyspark.ml.evaluation** **import** BinaryClassificationEvaluator

**from** **pyspark.sql.functions** **import** count, isnan, when, col, sum

# Initialize SparkSession

spark = SparkSession.builder \

.appName("Bank Customer Churn") \

.getOrCreate()

# Load the dataset

df = spark.read.csv("Churn\_Modelling.csv", header=True, inferSchema=True)

df.show(**5**)

df.printSchema()

# Check for missing values

missing\_values = df.select([sum(col(c).isNull().cast("int")).alias(c) **for** c **in** df.columns])

# Show the count of missing values for each column

missing\_values.show()

# Drop rows with missing value

df = df.dropna()

# Show the shape of the DataFrame

**print**("df shape:", (df.count(), len(df.columns)))

# Check for missing values

missing\_values = df.select([sum(col(c).isNull().cast("int")).alias(c) **for** c **in** df.columns])

# Show the count of missing values for each column

missing\_values.show()

#check number of rows

df.count()

**print**('Number of Rows: ', df.count())

#check number of columns

len(df.columns)

**print**('Number of columns: ', len(df.columns))

df.columns

# Define the StringIndexer for Geography

geography\_indexer = StringIndexer(inputCol="Geography", outputCol="GeographyIndex")

# Define the StringIndexer for Gender

gender\_indexer = StringIndexer(inputCol="Gender", outputCol="GenderIndex")

# Fit and transform the DataFrame

df = geography\_indexer.fit(df).transform(df)

df = gender\_indexer.fit(df).transform(df)

# Show the transformed DataFrame

df.show()

#columns to drop

# Drop columns

columns\_to\_drop = ['RowNumber', 'CustomerId', 'Surname', 'Geography','Gender']

df = df.drop(\*columns\_to\_drop)

# Show DataFrame

df.show()

# Feature Selection

feature\_cols = ['CreditScore', 'Age', 'Tenure', 'Balance','NumOfProducts', 'HasCrCard',

'IsActiveMember', 'EstimatedSalary', 'GeographyIndex', 'GenderIndex']

assembler = VectorAssembler(inputCols=feature\_cols, outputCol="unscaled\_features")

data = assembler.transform(df)

scaler = MinMaxScaler(inputCol="unscaled\_features", outputCol="features")

scaler\_model = scaler.fit(data)

scaled\_data = scaler\_model.transform(data)

# Split the data into training and testing sets

(train\_data, test\_data) = scaled\_data.randomSplit([**0.8**, **0.2**], seed=**42**)

# Model Building - Logistic Regression

lr = LogisticRegression(labelCol='Exited', featuresCol="features")

lr\_model = lr.fit(train\_data)

# Model Building - Random Forest

rf = RandomForestClassifier(labelCol='Exited', featuresCol="features")

rf\_model = rf.fit(train\_data)

# Model Building - SVM

svm = LinearSVC(labelCol='Exited', featuresCol="features")

svm\_model = svm.fit(train\_data)

# Model Evaluation

**def** **evaluate\_model**(model, test\_data):

predictions = model.transform(test\_data)

evaluator = BinaryClassificationEvaluator(labelCol='Exited')

accuracy = evaluator.evaluate(predictions)

**return** accuracy

# Evaluate Logistic Regression model

lr\_accuracy = evaluate\_model(lr\_model, test\_data)

# Evaluate Random Forest model

rf\_accuracy = evaluate\_model(rf\_model, test\_data)

# Evaluate SVM model

svm\_accuracy = evaluate\_model(svm\_model, test\_data)

# Print accuracies

**print**("Logistic Regression Accuracy:", lr\_accuracy)

**print**("Random Forest Accuracy:", rf\_accuracy)

**print**("SVM Accuracy:", svm\_accuracy)

# Feature Selection

feature\_cols = ['CreditScore', 'Age', 'Tenure', 'Balance','NumOfProducts', 'HasCrCard',

'IsActiveMember', 'EstimatedSalary', 'GeographyIndex', 'GenderIndex']

assembler = VectorAssembler(inputCols=feature\_cols, outputCol="unscaled\_features")

data = assembler.transform(df)

# Apply MinMaxScaler to standadize the data

scaler = MinMaxScaler(inputCol="unscaled\_features", outputCol="features")

scaler\_model = scaler.fit(data)

scaled\_data = scaler\_model.transform(data)

**from** **pyspark.ml.feature** **import** PCA

# Apply PCA

pca = PCA(k=**5**, inputCol="features", outputCol="pca\_features")

pca\_model = pca.fit(scaled\_data)

pca\_data = pca\_model.transform(scaled\_data)

# Split the data into training and testing sets

(train\_data, test\_data) = pca\_data.randomSplit([**0.8**, **0.2**], seed=**42**)

# Model Building - Logistic Regression

lr = LogisticRegression(labelCol='Exited', featuresCol="pca\_features")

lr\_model = lr.fit(train\_data)

# Model Building - Random Forest

rf = RandomForestClassifier(labelCol='Exited', featuresCol="pca\_features")

rf\_model = rf.fit(train\_data)

# Model Building - SVM

svm = LinearSVC(labelCol='Exited', featuresCol="pca\_features")

svm\_model = svm.fit(train\_data)

# Model Evaluation

**def** **evaluate\_model**(model, test\_data):

predictions = model.transform(test\_data)

evaluator = BinaryClassificationEvaluator(labelCol='Exited')

accuracy = evaluator.evaluate(predictions)

**return** accuracy

# Evaluate Logistic Regression model

lr\_accuracy = evaluate\_model(lr\_model, test\_data)

# Evaluate Random Forest model

rf\_accuracy = evaluate\_model(rf\_model, test\_data)

# Evaluate SVM model

svm\_accuracy = evaluate\_model(svm\_model, test\_data)

# Print accuracies

**print**("Logistic Regression Accuracy:", lr\_accuracy)

**print**("Random Forest Accuracy:", rf\_accuracy)

**print**("SVM Accuracy:", svm\_accuracy)

# Stop SparkSession

spark.stop()