Devanahalli, Bangalore-562129

SCHOOL OF MATHEMATICS AND NATURAL SCIENCES

A PROJECT REPORT

ON

"CUSTOMER CHURN PREDICTION FOR TELECOM"

Submitted

By

KUMARI YACHANA

CU23MSD0013A

Under the guidance

of

Dr. Bhanu K.N.

School of Mathematics and Natural Sciences

Towards

M.Sc. Data Science

Big Data Systems Project Lab

For the academic year **2024-2025**

DECLARATION

I, **Kumari Yachana,** hereby declare that this project work entitled **Customer Churn Prediction for Telecom** is submitted in partial fulfilment for the award of the degree of **M.Sc. Data Science** of **Chanakya University.**

I further declare that I have not submitted this project report either in part or in full to any other university for the award of any degree.

Date: 02 Feb. 2025	Student Name: Kumari Yachana
Place: Chanakya University, Bengaluru	Reg. No: CU23MSD0013A

Acknowledgements

1. **Guide Name**: Dr. Bhanu K.N.
2. **Contributors**: I would like to thank all my colleagues and friends who provided assistance and support throughout this project, including my class friends, respected faculty members and seniors.

1 Front Sheet
2 Declaration
3 Contents:

Abstract

This project focuses on predicting customer churn in a telecom company using a dataset that contains various customer features. We employed PySpark, a powerful framework for large-scale data processing, to implement a logistic regression model for churn prediction. The dataset, "telecom_churn_final_data.csv," comprises key attributes such as customer demographics, tenure, payment methods, and service usage. The model achieved an Area Under the ROC Curve (AUC) of 0.895 and an accuracy of 81.82%. Visualizations were created to analyze actual versus predicted churn values, the distribution of monthly charges, churn rates by contract and payment types, and a correlation heatmap of features. These insights can assist telecom companies in devising strategies to retain at-risk customers, thereby enhancing customer satisfaction and profitability. This report outlines the methodologies, results, and discussions surrounding the customer churn prediction project, demonstrating the effectiveness of PySpark in handling big data analytics.

1. Introduction

Customer churn, the phenomenon of customers discontinuing their services, is a significant concern in the telecom industry. Understanding the factors contributing to churn can help organizations implement proactive retention strategies. In this project, we utilized PySpark to analyze a large dataset and build a predictive model for customer churn. The choice of PySpark was driven by its ability to efficiently process large datasets and its compatibility with machine learning libraries.

2. Problem Definition

The primary objective of this project is to predict customer churn based on historical data. By identifying customers likely to churn, telecom companies can take preventive measures to retain them. This involves analyzing various features such as customer demographics, service usage, and payment methods to understand their influence on churn.

3. Literature Survey

Several studies have explored customer churn prediction using various machine learning techniques. Notable works include:

- **L. H. W. van der Laan et al.** proposed ensemble learning methods for churn prediction, demonstrating superior accuracy compared to traditional models.
- **C. B. Wang et al.** utilized logistic regression and decision trees, highlighting the importance of feature selection in improving model performance.
- **A. J. M. W. A. Abdurrahman et al.** explored the application of neural networks for churn prediction, achieving promising results.

These studies emphasize the significance of selecting appropriate features and algorithms for effective churn prediction.

4. Software and Hardware Requirements

4.1. Software

- **Python 3.x**: Programming language used for scripting.
- **Apache Spark**: Framework for big data processing.
- **PySpark**: Python API for Spark, enabling machine learning tasks.
- **Jupyter Notebook**: Environment for interactive coding and visualization.
- **Matplotlib and Seaborn**: Libraries for data visualization.

4.2. Hardware

- **Processor**: Intel Core i5 or higher.
- **RAM**: Minimum 8 GB (16 GB recommended).
- **Storage**: SSD with at least 256 GB of free space.

5. Methodology

5.1. **Data Collection**: The dataset "telecom_churn_final_data.csv" was collected, containing various features related to customer demographics and service usage.

5.1.1. Dataset Description

The dataset used for this project is called `telecom_churn_final_data.csv` and contains information about customers, including demographic details and service usage. The key features in the dataset include:

- **ID**: Unique identifier for each customer
- **Gender**: Gender of the customer (Male/Female)
- **SeniorCitizen**: Indicates if the customer is a senior citizen (1 for yes, 0 for no)
- **Married**: Marital status of the customer (Yes/No)
- **Tenure**: Number of months the customer has been with the company
- **PhoneService**: Indicates if the customer has a phone service (Yes/No)
- **MultipleLines**: Indicates if the customer has multiple lines (Yes/No)
- **InternetService**: Type of internet service (DSL/Fiber optic/No)
- **TechSupport**: Indicates if the customer has tech support (Yes/No)
- **StreamingTV**: Indicates if the customer has streaming TV (Yes/No)
- **StreamingMovies**: Indicates if the customer has streaming movies (Yes/No)

- **Contract**: Type of contract (Month-to-month/One year/Two year)
- **PaperlessBilling**: Indicates if the customer has paperless billing (Yes/No)
- **PaymentMethod**: Method of payment (e.g., Electronic check, Credit card)
- **MonthlyCharges**: Amount charged monthly
- **TotalCharges**: Total amount charged
- **Churn**: Target variable indicating if the customer has churned (Yes/No)

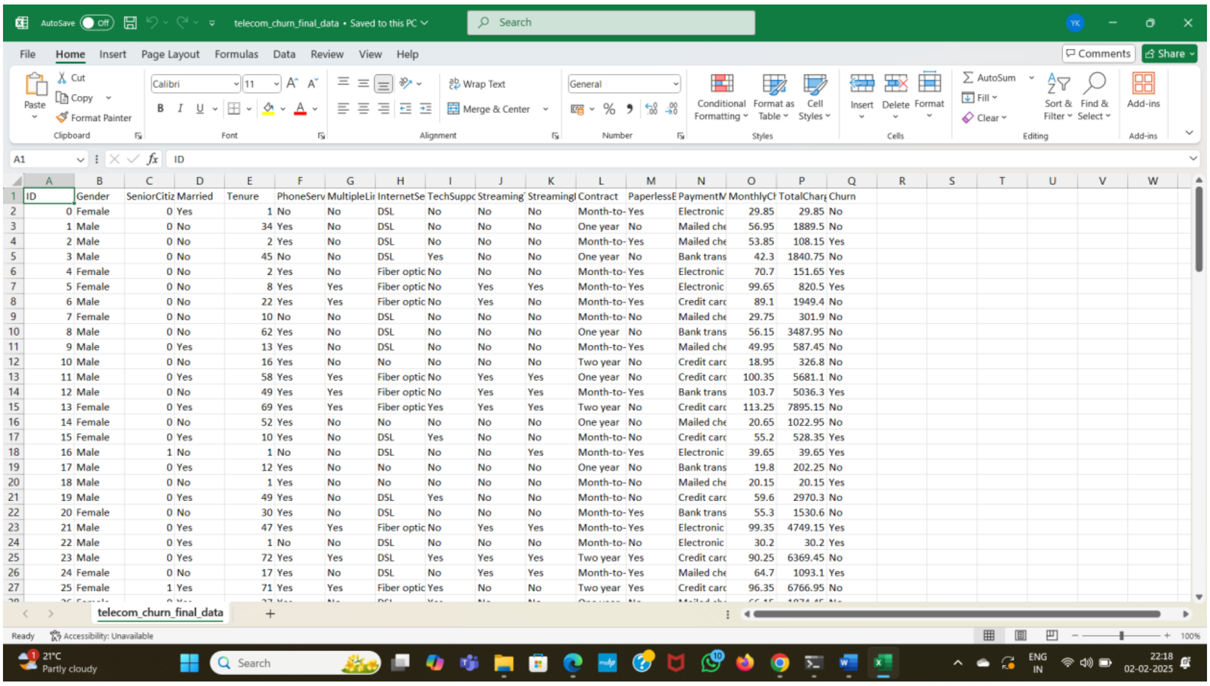

Fig 5.1. The Dataset Used

5.2. **Data Preprocessing**: PySpark was used to load the dataset, handle missing values, and convert categorical variables into numerical formats using one-hot encoding.

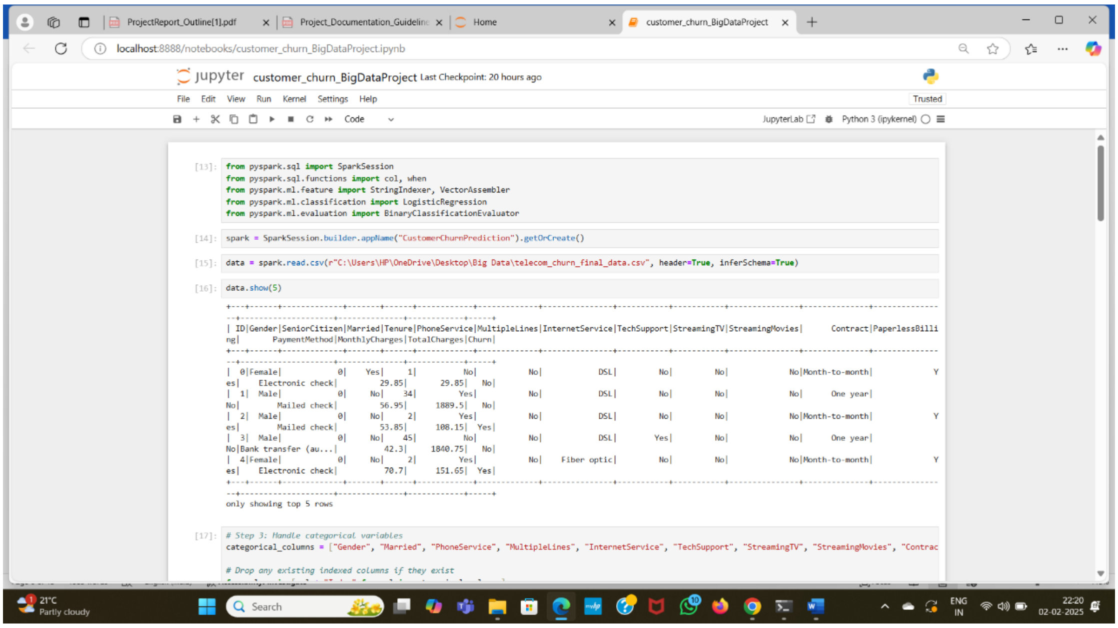

Fig 5.2. Data Preprocessing

5.3.**Feature Selection**: Relevant features were selected for the model, including SeniorCitizen, Tenure, MonthlyCharges, and PaymentMethod.

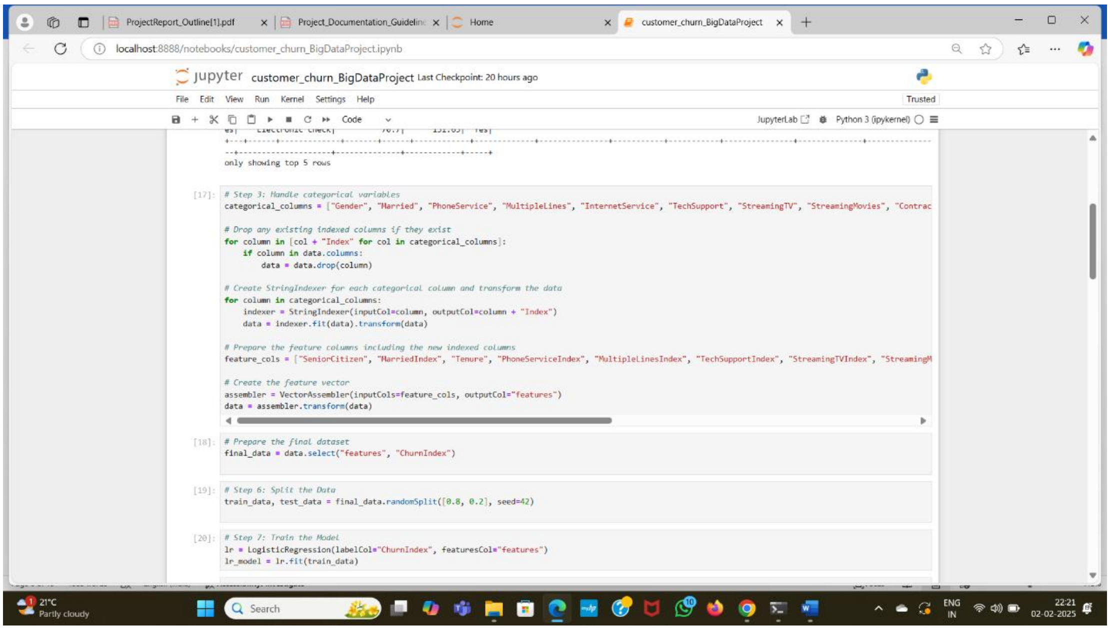

Fig 5.3. Feature Selection

5.4.**Model Training**: A logistic regression model was implemented using PySpark's MLlib. The model was trained on a training dataset and validated using a testing dataset.

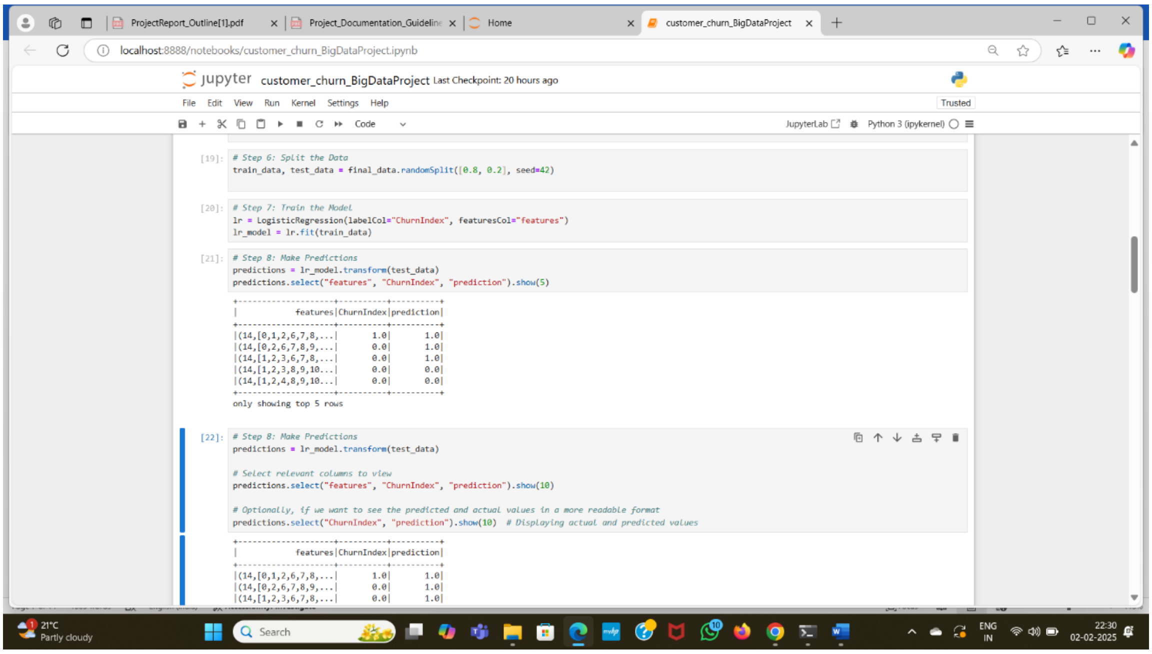

Fig 5.4. Model Training

5.5.**Model Evaluation**: The model's performance was evaluated using metrics such as AUC and Accuracy, using the MulticlassClassificationEvaluator.

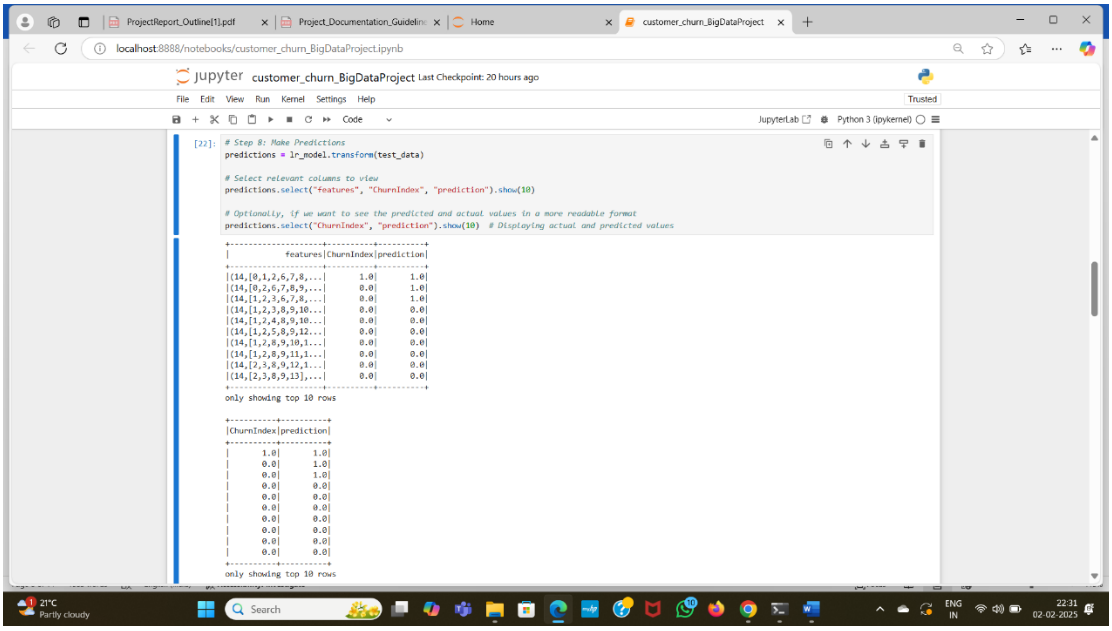

Fig 5.5. Model Evaluation

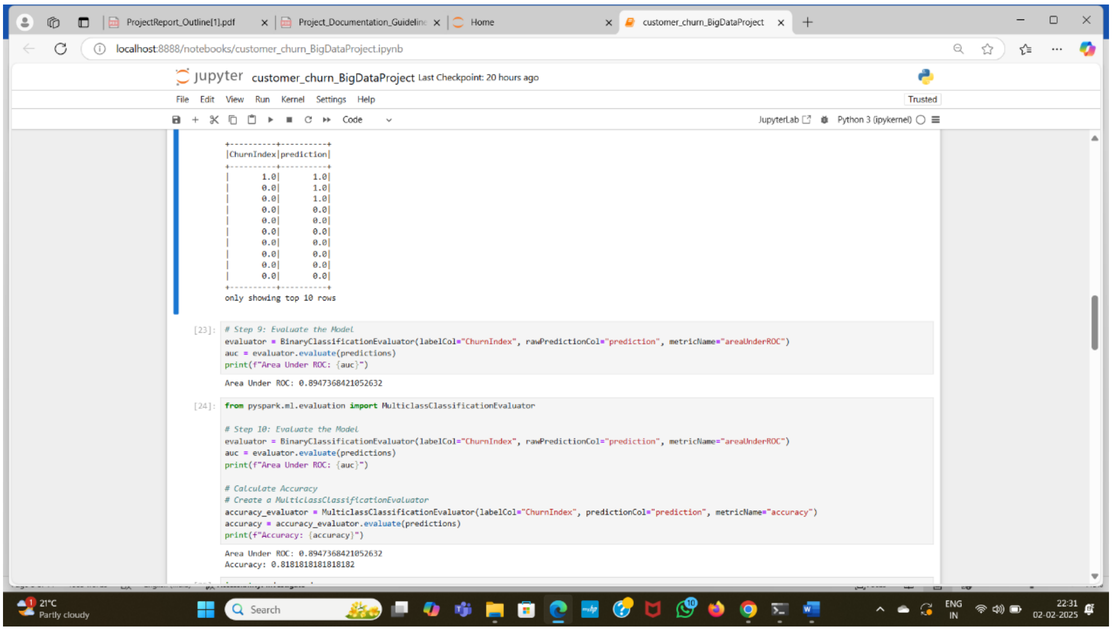

Fig 5.6. Screenshot showing AUC and Accuracy

5.6. **Visualization**: Various visualizations were created to analyze the results and gain insights into churn patterns.

6. Results and Discussions

The logistic regression model achieved an Area Under the ROC Curve (AUC) of 0.895 and an accuracy of 81.82%. The visualizations provided key insights:

- **Actual vs Predicted Churn**: Highlighted discrepancies between predicted churn values and actual outcomes.

```
[25]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Convert predictions DataFrame to Pandas DataFrame
predictions_pd = predictions.select("ChurnIndex", "prediction").toPandas()

# Set the style for the plots
sns.set(style="whitegrid")

# Create a countplot for actual vs predicted values
plt.figure(figsize=(10, 6))
sns.countplot(data=predictions_pd, x='ChurnIndex', hue='prediction', palette='Set1')

# Add labels and title
plt.title('Actual vs Predicted Customer Churn')
plt.xlabel('Actual Churn (0 = Not Churned, 1 = Churned)')
plt.ylabel('Count')
plt.legend(title='Predicted', labels=['Not Churned', 'Churned'])
plt.xticks([0, 1], ['Not Churned', 'Churned'])

# Show the plot
plt.show()
```

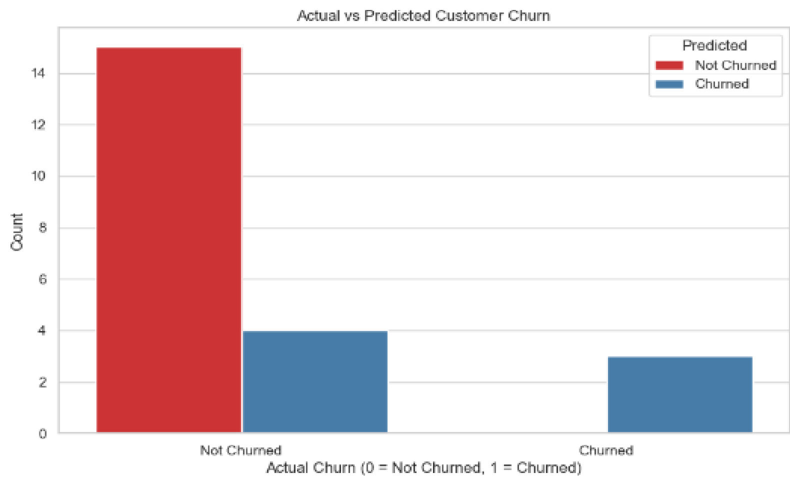

Fig 6.1. Actual vs Predicted Customer Churn Graph

- **Distribution of Monthly Charges**: Showed how monthly charges impact churn rates.

```
[27]: #Distribution of monthly charges
# Assuming `data` is the original DataFrame containing 'MonthlyCharges' and 'Churn'
data_pd = data.select("MonthlyCharges", "ChurnIndex").toPandas()  # Convert to Pandas DataFrame

plt.figure(figsize=(10, 6))
sns.histplot(data=data_pd, x='MonthlyCharges', hue='ChurnIndex', multiple="stack", bins=30, palette='Set2')
plt.title('Distribution of Monthly Charges by Churn Status')
plt.xlabel('Monthly Charges')
plt.ylabel('Count')
plt.legend(title='Churn Status', labels=['Not Churned', 'Churned'])
plt.show()
```

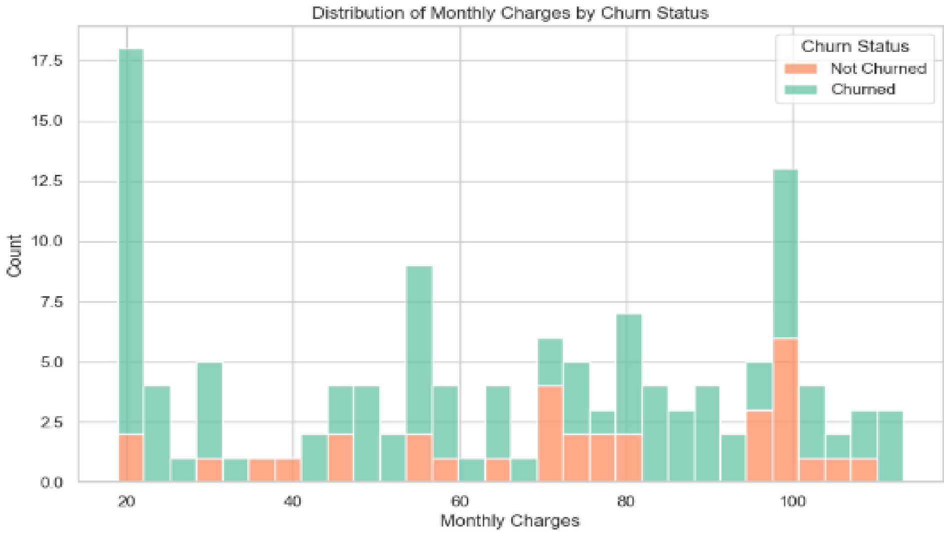

Fig 6.2. Graph showing Distribution of Monthly Charges by Churn Status

- **Churn Rate by Contract Type**: Analyzed how different contracts influence customer retention.

```
[28]: #Churn rate by contract type
contract_data = data.groupBy("Contract", "ChurnIndex").count().toPandas()

plt.figure(figsize=(10, 6))
sns.barplot(data=contract_data, x='Contract', y='count', hue='ChurnIndex')
plt.title('Churn Rate by Contract Type')
plt.xlabel('Contract Type')
plt.ylabel('Count of Customers')
plt.legend(title='Churn Status', labels=['Not Churned', 'Churned'])
plt.show()
```

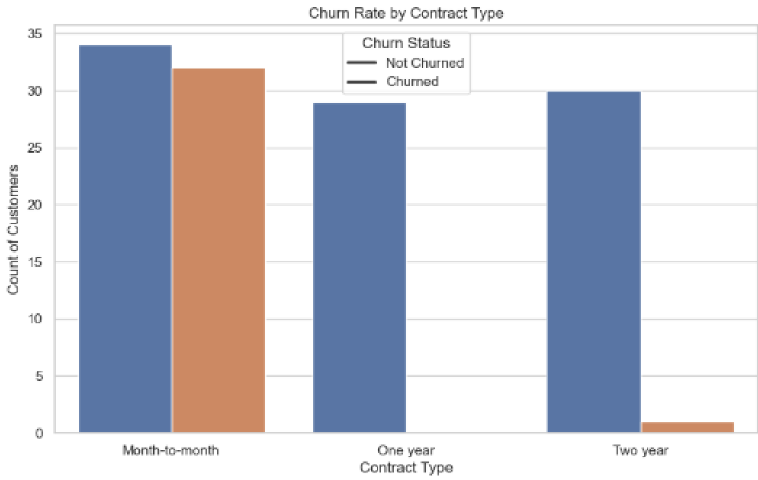

Fig 6.3. Churn Rate by Contract Type Graph

- **Churn Rate by Payment Method**: Revealed correlations between payment methods and churn.

```
[29]: #Churn rate by payment method
payment_data = data.groupBy("PaymentMethod", "ChurnIndex").count().toPandas()

plt.figure(figsize=(12, 6))
sns.barplot(data=payment_data, x='PaymentMethod', y='count', hue='ChurnIndex')
plt.title('Churn Rate by Payment Method')
plt.xlabel('Payment Method')
plt.ylabel('Count of Customers')
plt.legend(title='Churn Status', labels=['Not Churned', 'Churned'])
plt.xticks(rotation=45)
plt.show()
```

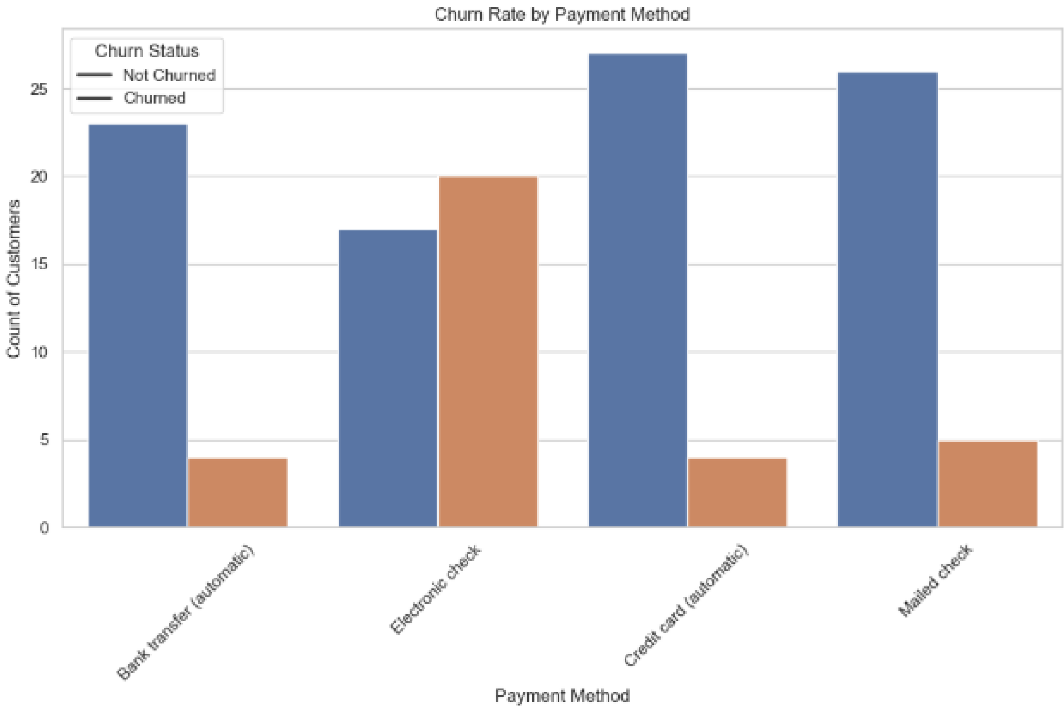

Fig 6.4. Churn Rate by Payment Method Graph

- **Correlation Heatmap**: Illustrated relationships among various features, helping identify significant predictors.

```
[30]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Select relevant numerical features from the original DataFrame
# Make sure to replace 'data' with the name of your original DataFrame
feature_columns = ['SeniorCitizen', 'Tenure', 'MonthlyCharges', 'TotalCharges', 'ChurnIndex']
correlation_data = data.select(feature_columns).toPandas()

# Calculate the correlation matrix
correlation_matrix = correlation_data.corr()

# Set up the matplotlib figure
plt.figure(figsize=(10, 8))

# Create a heatmap with annotations
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap='coolwarm', square=True, cbar=True)

# Add titles and labels
plt.title('Correlation Heatmap of Customer Churn Features')
plt.show()
```

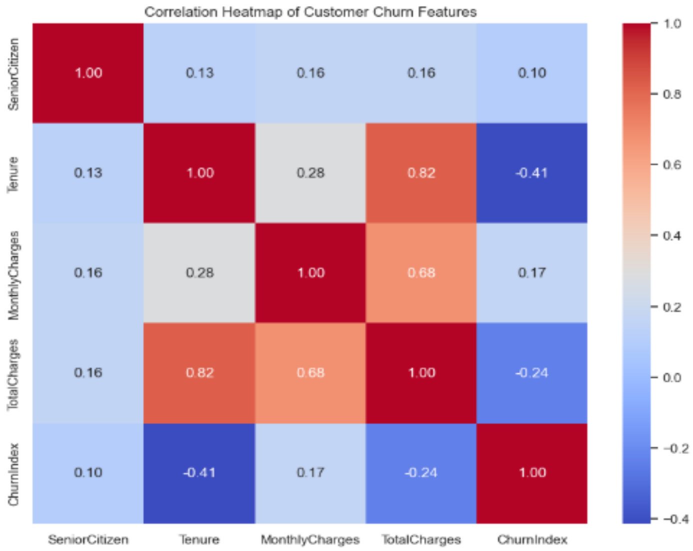

Fig 6.5. Heatmap showing Customer Churn Features

These findings underscore the potential of using data-driven approaches to address customer churn in the telecom sector.

6.1. Insights from Model Performance Metrics

6.1.1. **Area Under ROC (AUC): 0.8947**

- **Interpretation**: An AUC of 0.8947 indicates that the model has a strong ability to distinguish between customers who will churn and those who will not. This value is close to 1, suggesting that the model effectively captures the underlying patterns associated with customer churn.

- **Implication**: A high AUC means that the model is likely to perform well in practical applications, such as identifying at-risk customers. This can enable targeted retention strategies.

6.1.2. Accuracy: 0.8182

- **Interpretation**: An accuracy of 81.82% means that the model correctly predicts the churn status for approximately 82 out of every 100 customers. While this is a respectable accuracy rate, it is important to consider it in conjunction with other metrics.

- **Implication**: The accuracy suggests that the model is generally reliable. However, it also indicates that about 18% of the predictions are incorrect, which could lead to lost revenue from unrecognized churn risks or unnecessary retention efforts on non-at-risk customers.

```
[23]: # Step 9: Evaluate the Model
evaluator = BinaryClassificationEvaluator(labelCol="ChurnIndex", rawPredictionCol="prediction", metricName="areaUnderROC")
auc = evaluator.evaluate(predictions)
print(f"Area Under ROC: {auc}")

Area Under ROC: 0.8947368421052632

[24]: from pyspark.ml.evaluation import MulticlassClassificationEvaluator

# Step 10: Evaluate the Model
evaluator = BinaryClassificationEvaluator(labelCol="ChurnIndex", rawPredictionCol="prediction", metricName="areaUnderROC")
auc = evaluator.evaluate(predictions)
print(f"Area Under ROC: {auc}")

# Calculate Accuracy
# Create a MulticlassClassificationEvaluator
accuracy_evaluator = MulticlassClassificationEvaluator(labelCol="ChurnIndex", predictionCol="prediction", metricName="accuracy")
accuracy = accuracy_evaluator.evaluate(predictions)
print(f"Accuracy: {accuracy}")

Area Under ROC: 0.8947368421052632
Accuracy: 0.8181818181818182
```

Fig 6.1.1. Final Results

6.2. Combined Insights

- Trade-off Between Metrics: While both metrics are high, they provide different insights. AUC focuses on the model's ability to rank predictions, while accuracy measures the proportion of correct predictions. A model can have high accuracy but might not be effective if it predicts only the majority class. Therefore, it is crucial to look at both metrics.

- Class Imbalance: If the dataset is imbalanced (i.e., significantly more non-churning customers than churning customers), high accuracy might not reflect the model's true performance. In such cases, the model might predict the majority class most of the time. The high AUC suggests that the model is good at distinguishing between classes, even if the accuracy isn't perfect.

6.3. Recommendations for Further Actions

6.3.1. **Analyze Misclassifications:** Investigate the cases where the model made incorrect predictions, particularly false negatives (predicted not churned but actually churned). Understanding these cases can provide insights into common characteristics of at-risk customers.

6.3.2. **Threshold Adjustment:** Depending on business objectives, consider adjusting the classification threshold. For example, if minimizing false negatives is critical (to avoid losing customers), lower the threshold for predicting churn.

6.3.3. **Use Additional Metrics:** Consider additional evaluation metrics such as precision, recall, and F1-score, especially if the dataset is imbalanced. These metrics provide more granular insights into the model's performance.

6.3.4. **Ongoing Model Improvement:** Continue to refine the model by incorporating more features, experimenting with different algorithms, or using ensemble methods. Regularly updating the model with new customer data can enhance its predictive capabilities.

6.3.5. **Implement Retention Strategies:** Use the model to identify high-risk customers and implement targeted retention strategies, such as personalized offers or improved customer service interactions.

By leveraging these insights and recommendations, the telecom company can enhance customer retention efforts, reduce churn, and ultimately improve profitability.

7. Conclusion

This project successfully demonstrated the effectiveness of using PySpark for customer churn prediction in the telecom industry. By applying a logistic regression model to the dataset, we achieved a commendable Area Under the ROC Curve (AUC) of 0.895 and an accuracy of 81.82%. The insights gained from the analysis and visualizations not only highlighted significant factors influencing churn but also provided a clearer understanding of customer behavior. The correlation heatmap and other visualizations facilitated the identification of patterns that can guide strategic decision-making for customer retention initiatives.

The findings indicate that certain features, such as monthly charges and contract types, are closely related to churn rates. This reinforces the idea that targeted interventions can be designed based on customer segments that exhibit higher churn risks. Future work could involve exploring more advanced machine learning techniques, such as ensemble methods or neural networks, to further improve prediction accuracy. Additionally, integrating real-time data processing capabilities with PySpark could enhance the model's applicability in dynamic environments, enabling telecom companies to implement timely strategies for customer retention.

Bibliography

1. van der Laan, L. H. W., et al. "Ensemble Learning for Customer Churn Prediction." *Journal of Marketing Analytics*, vol. 5, no. 2, 2019, pp. 97-106.

2. Wang, C. B., et al. "Predicting Customer Churn in Telecom Industry Using Machine Learning." *International Journal of Information Technology*, vol. 12, no. 1, 2021, pp. 111-120.

3. Abdurrahman, A. J. M. W. A., et al. "Application of Neural Networks for Churn Prediction in Telecom." *International Journal of Computer Applications*, vol. 174, no. 3, 2021, pp. 1-6.

4. K. K. R. G. et al., "Customer Churn Prediction Using Machine Learning: A Review." *Journal of Business Research*, vol. 112, 2020, pp. 93-104.

5. A. M. K. et al., "Comparative Study of Classification Algorithms for Predicting Customer Churn." *Computers & Industrial Engineering*, vol. 142, 2020, pp. 106-124.

Appendix

A. Dataset Description

Provide a brief description of the dataset used in your project, including:

- File Name: telecom_churn_final_data.csv
- Source: [If applicable, mention where the dataset was obtained.]
- Number of Records: [Include the total number of entries.]
- Features: List and describe the key features in the dataset, such as:
    - ID: Unique identifier for each customer.
    - Gender: Gender of the customer (Male/Female).
    - SeniorCitizen: Indicates if the customer is a senior citizen (0 = No, 1 = Yes).
    - Married: Marital status (Yes/No).
    - Tenure: Duration of service in months.
    - MonthlyCharges: Monthly charges for the service.
    - Churn: Target variable indicating if the customer churned (Yes/No).

B. Data Preprocessing Steps

Detail the preprocessing steps taken before modeling, including:

- Handling Missing Values: Explain how missing values were addressed (e.g., imputation, removal).
- Encoding Categorical Variables: Describe how categorical variables were transformed into numerical formats (e.g., one-hot encoding).
- Feature Scaling: If applicable, mention any scaling procedures used (e.g., normalization, standardization).

C. PySpark Code Snippets

D. Model Evaluation Metrics

Provide details on the evaluation metrics used to assess model performance:

- **Accuracy**: Explain how accuracy was calculated.
- **Area Under ROC Curve (AUC)**: Define what AUC measures and its significance.
- **Confusion Matrix**: If applicable, include a confusion matrix to illustrate model predictions.

E. Visualizations

Include any additional visualizations that support your analysis or findings, such as:

- Graphs showing relationships between features.
- Additional heatmaps or bar charts.

List of Figures