STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY

Big Data Technologies - BIA 678

Final Project
Customer Churn Prediction with PySpark

Instructor : Denghui Zhang

Group 6 : Gahana Nagaraja
Namratha Nagathihalli Anantha
Niharika Mysore Prakasha

PRESENTERS

Gahana Nagaraja
Data Science
Bengaluru, Karnataka, India

Namratha Nagathihalli Anantha
Data Science
Bengaluru, Karnataka, India

Niharika Mysore Prakasha
Software Engineering
Mysore, Karnataka, India

CONTENT

INTRODUCTION

Background

- As the name suggests, churn prediction is a methodology that involves the data-driven pinpointing of customer accounts that are at high risk of downgrading their engagement, canceling their subscription, or disengaging with your company. Predicting customer churn is crucial for businesses because it allows them to proactively identify at-risk customers and take appropriate actions to retain them.

Objective

- Identifying at-risk customers, building predictive models, and improving retention strategies.
- Machine learning algorithms are then applied to this data to build predictive models that can classify customers as churners or non-churners based on various features.

PySpark

- PySpark is the Python interface for Apache Spark, a powerful open-source framework for large-scale data processing.
- Combines the power of Apache Spark with the simplicity and flexibility of Python, enabling data engineers and data scientists to build scalable solutions for complex data problems.
- PySpark is a versatile tool that combines Python's simplicity with Spark's scalability, making it a popular choice for big data processing, analytics, and machine learning applications.
- In customer churn prediction, PySpark enables data preprocessing, feature engineering, model training, and evaluation at scale, allowing for faster analysis and prediction. Its integration with machine learning libraries like MLlib facilitates the development of sophisticated predictive models for identifying at-risk customers.
- Overall, PySpark plays a crucial role in enhancing the speed, scalability, and effectiveness of customer churn prediction.

DATA COLLECTION

- Data has been collected from Telecom churn datasets (Kaggle) where, each row represents a customer and each column contains customer's attributes.

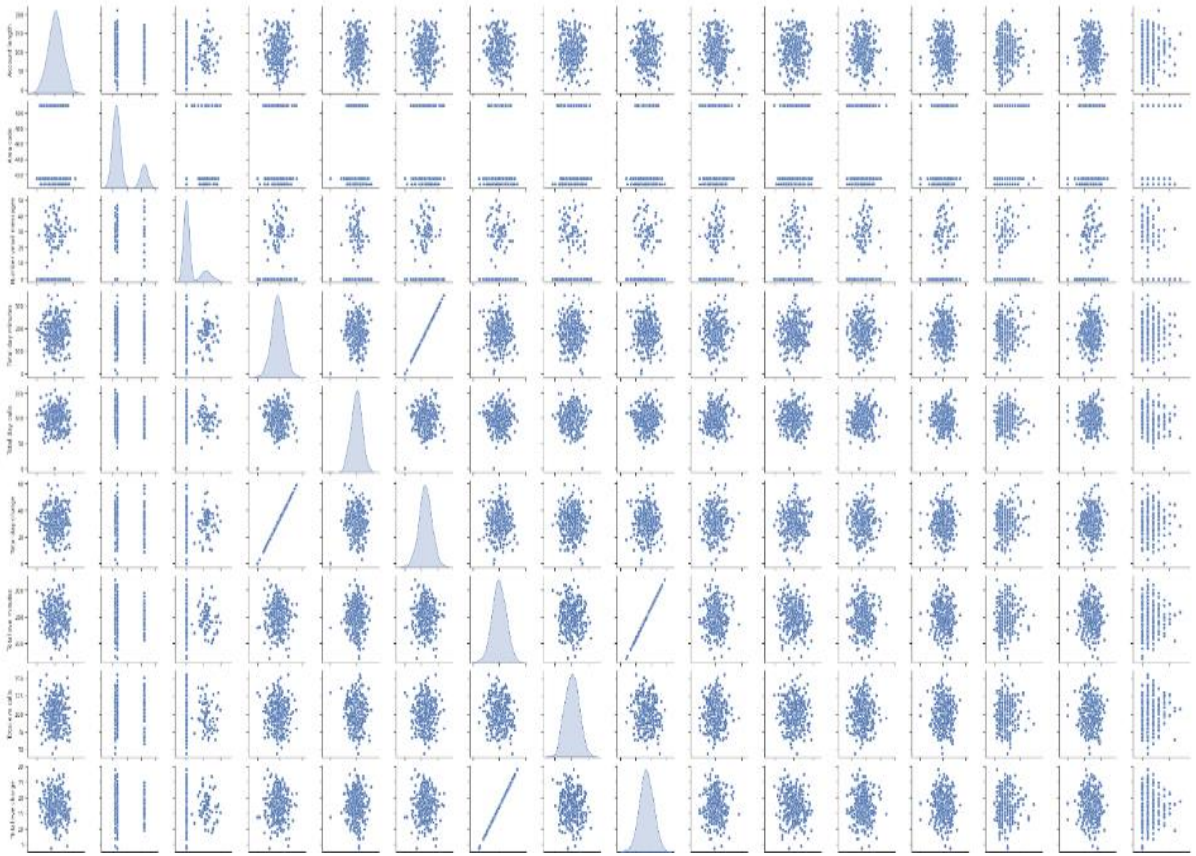
```
|-- State: string (nullable = true)
|-- Account length: integer (nullable = true)
|-- Area code: integer (nullable = true)
|-- International plan: string (nullable = true)
|-- Voice mail plan: string (nullable = true)
|-- Number vmail messages: integer (nullable = true)
|-- Total day minutes: double (nullable = true)
|-- Total day calls: integer (nullable = true)
|-- Total day charge: double (nullable = true)
|-- Total eve minutes: double (nullable = true)
|-- Total eve calls: integer (nullable = true)
|-- Total eve charge: double (nullable = true)
|-- Total night minutes: double (nullable = true)
|-- Total night calls: integer (nullable = true)
|-- Total night charge: double (nullable = true)
|-- Total intl minutes: double (nullable = true)
|-- Total intl calls: integer (nullable = true)
|-- Total intl charge: double (nullable = true)
|-- Customer service calls: integer (nullable = true)
|-- Churn: boolean (nullable = true)
```

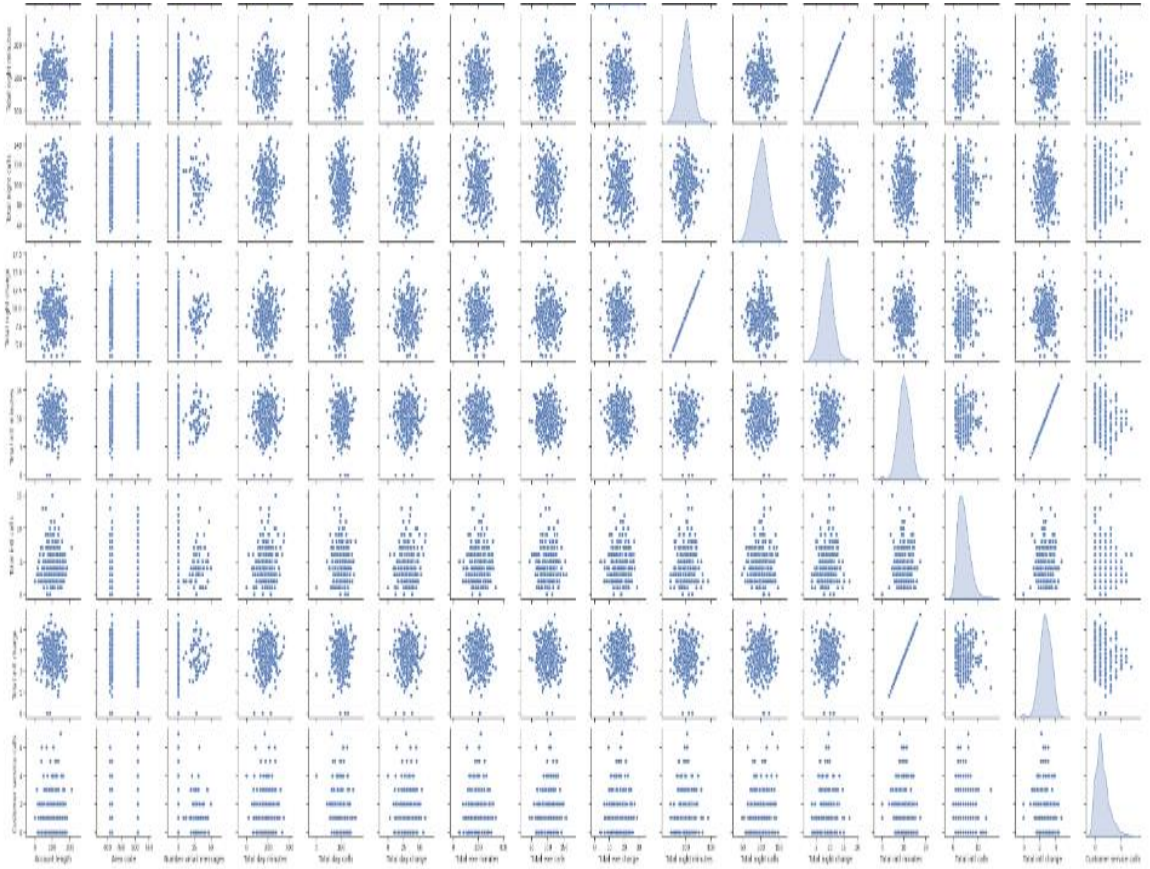
Training dataset contains:
2666 samples

Testing dataset contains:
667 samples

CORRELATION ANALYSIS

The statistical analysis is performed using the seaborn package to examine correlations between the numeric columns by generating scatter plots of them.



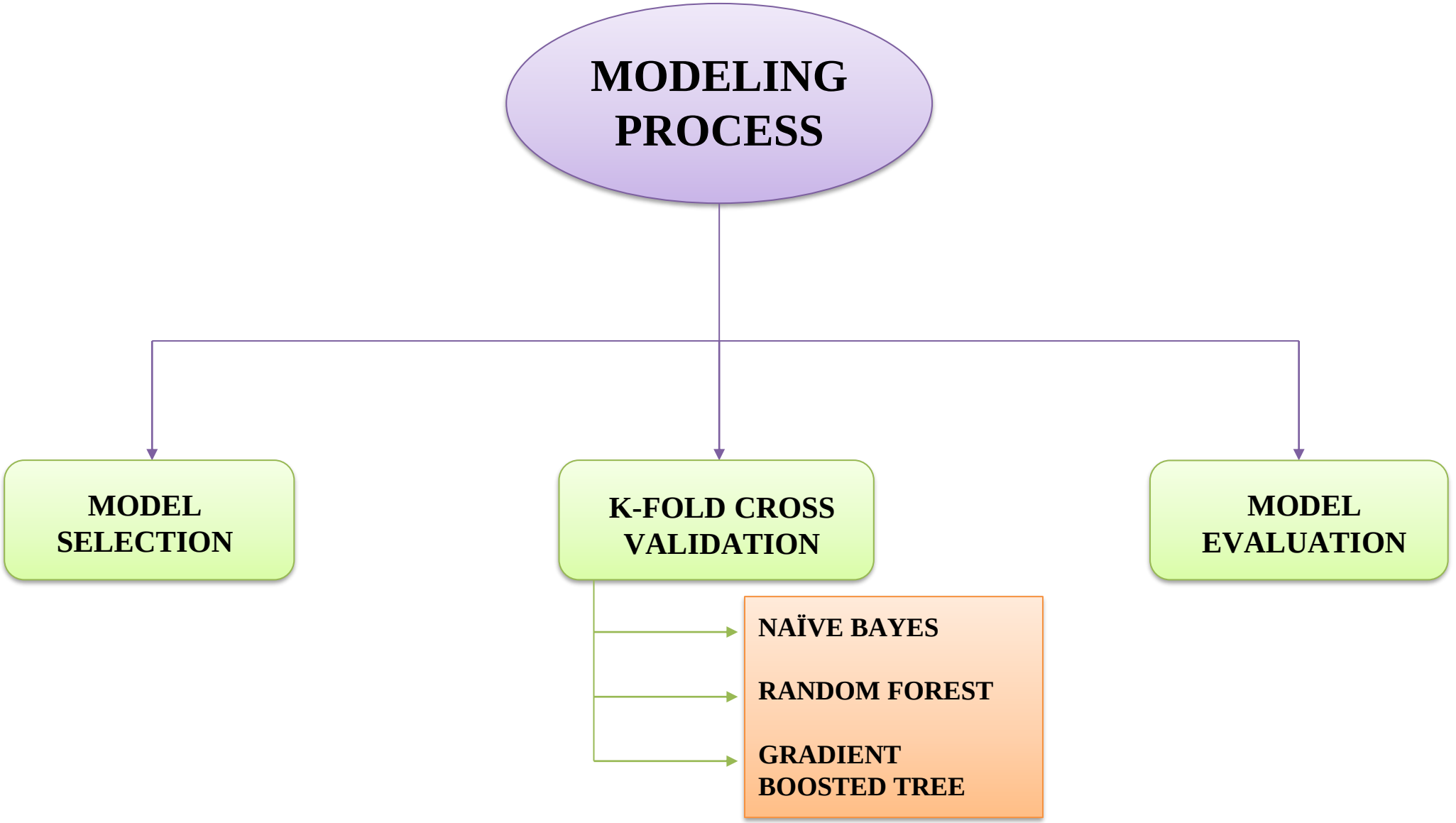


DATA PREPROCESSING

After statistical analysis, we have converted categorical columns into numerical ones by assigning a value of 1 for "true/yes" and 0 for "false/no".

	0	1	2	3	4
State	KS	OH	NJ	OH	OK
Account length	128	107	137	84	75
Area code	415	415	415	408	415
International plan	No	No	No	Yes	Yes
Voice mail plan	Yes	Yes	No	No	No
Number vmail messages	25	26	0	0	0
Total day minutes	265.1	161.6	243.4	299.4	166.7
Total day calls	110	123	114	71	113
Total day charge	45.07	27.47	41.38	50.9	28.34
Total eve minutes	197.4	195.5	121.2	61.9	148.3
Total eve calls	99	103	110	88	122
Total eve charge	16.78	16.62	10.3	5.26	12.61
Total night minutes	244.7	254.4	162.6	196.9	186.9
Total night calls	91	103	104	89	121
Total night charge	11.01	11.45	7.32	8.86	8.41
Total intl minutes	10.0	13.7	12.2	6.6	10.1
Total intl calls	3	3	5	7	3
Total intl charge	2.7	3.7	3.29	1.78	2.73
Customer service calls	1	1	0	2	3
Churn	False	False	False	False	False

	0	1	2	3	4
State	KS	OH	NJ	OH	OK
Account length	128	107	137	84	75
Area code	415	415	415	408	415
International plan	0.0	0.0	0.0	1.0	1.0
Voice mail plan	1.0	1.0	0.0	0.0	0.0
Number vmail messages	25	26	0	0	0
Total day minutes	265.1	161.6	243.4	299.4	166.7
Total day calls	110	123	114	71	113
Total day charge	45.07	27.47	41.38	50.9	28.34
Total eve minutes	197.4	195.5	121.2	61.9	148.3
Total eve calls	99	103	110	88	122
Total eve charge	16.78	16.62	10.3	5.26	12.61
Total night minutes	244.7	254.4	162.6	196.9	186.9
Total night calls	91	103	104	89	121
Total night charge	11.01	11.45	7.32	8.86	8.41
Total intl minutes	10.0	13.7	12.2	6.6	10.1
Total intl calls	3	3	5	7	3
Total intl charge	2.7	3.7	3.29	1.78	2.73
Customer service calls	1	1	0	2	3
Churn	0.0	0.0	0.0	0.0	0.0


MODELING PROCESS
MODEL SELECTION
K-FOLD CROSS VALIDATION
MODEL EVALUATION
NAÏVE BAYES
RANDOM FOREST
GRADIENT BOOSTED TREE

MODEL SELECTION

- Model selection, coupled with pipelining techniques, is a crucial aspect of machine learning workflows.
- Pipelining allows for the seamless chaining of data transformers, estimators, and model selectors, streamlining the process of preparing and evaluating models.
- In the context of the ML package, data is formatted into a DataFrame using the VectorAssembler function, and passed through a pipeline of transformers like StringIndexer and VectorIndexer for indexing label and feature fields, respectively.
- This formatted data is then subjected to k-fold cross-validation, where the dataset is split into k partitions for training and testing.
- Through iterative training and evaluation cycles using varied model parameters, the optimal model configuration is determined based on performance metrics such as the F1 score.
- This systematic approach ensures reliable model selection, ultimately leading to models that generalize effectively to new data while leveraging the efficiency of pipelining techniques.

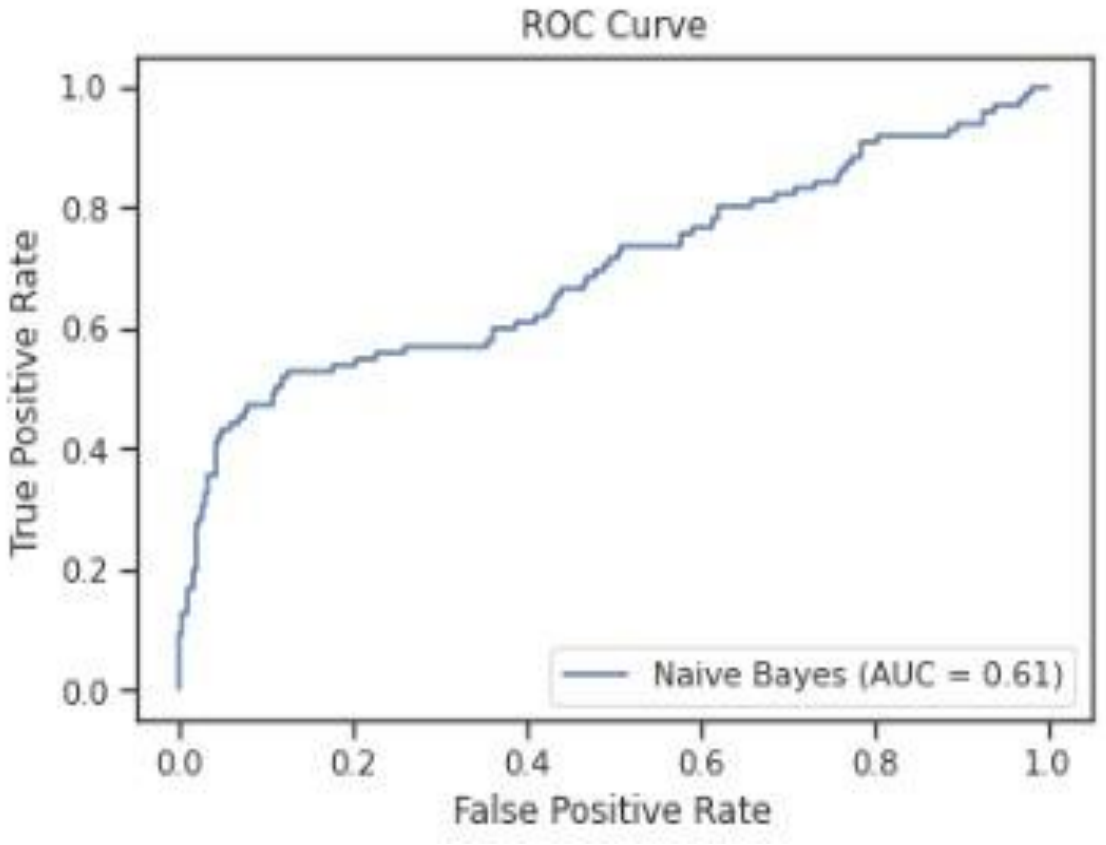
K-FOLD CROSS VALIDATION

- K-fold cross-validation is a technique used to assess the performance and generalization ability of a machine learning model. It's particularly useful when you have a limited amount of data and want to make the most out of it by partitioning it into subsets for both training and testing.
- The data is partitioned into k equal-sized folds, where each fold serves as both training and validation data iteratively. Following each training iteration, model performance is assessed using the validation data, resulting in k performance scores. These scores are then averaged to derive a final performance metric, offering a more reliable estimate of the model's performance compared to a single train-test split.
- K-fold cross-validation helps to reduce the variance in the performance estimate, provides a more accurate estimate of the model's performance, and allows for better assessment of how well the model generalizes to new, unseen data.
- This method is employed to evaluate the effectiveness of machine learning algorithms, including Naive Bayes, Random Forest Classifier, and Gradient-boosted tree classifier.

NAÏVE BAYES

- Naive Bayes is a probabilistic classifier that assumes independence among features.
- Naive Bayes classifiers are often a good choice for certain situations due to their simplicity, efficiency, low bias nature and effectiveness in certain types of data
- The ROC curve for Naive Bayes depicts the relationship between False Positive Rate (FPR) and True Positive Rate (TPR), occupying an area of 0.61 under the curve.
- Naive Bayes classifier achieved an accuracy of 53.67%.

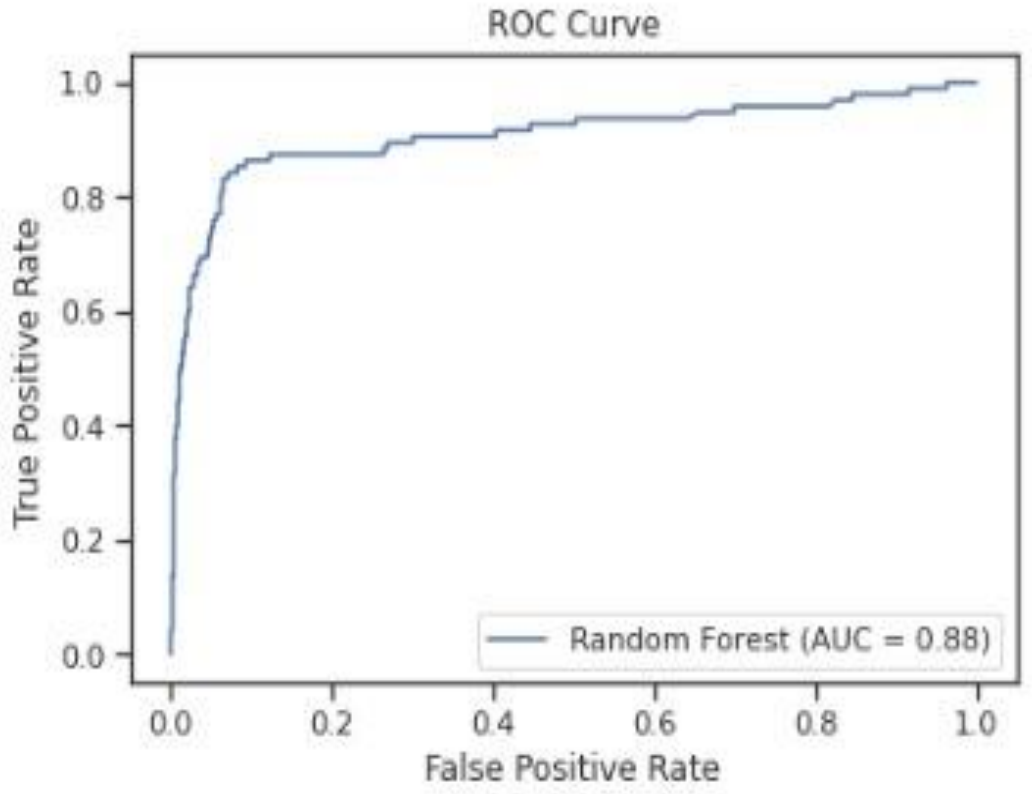
```
Naive Bayes ROC AUC: 0.6070022083179978
```



RANDOM FOREST

- Random Forest is an ensemble learning method that builds multiple decision trees and combines their predictions.
- It's robust to overfitting, handles categorical variables well, and provides feature importance scores, making it suitable for customer churn prediction tasks.
- The ROC curve for Random forest depicts the relationship between False Positive Rate (FPR) and True Positive Rate (TPR), occupying an area of 0.88 under the curve.
- Random Forest classifier achieved an accuracy of 88.91%.

Random Forest ROC AUC: 0.8782572690467427



GRADIENT BOOSTED TREE

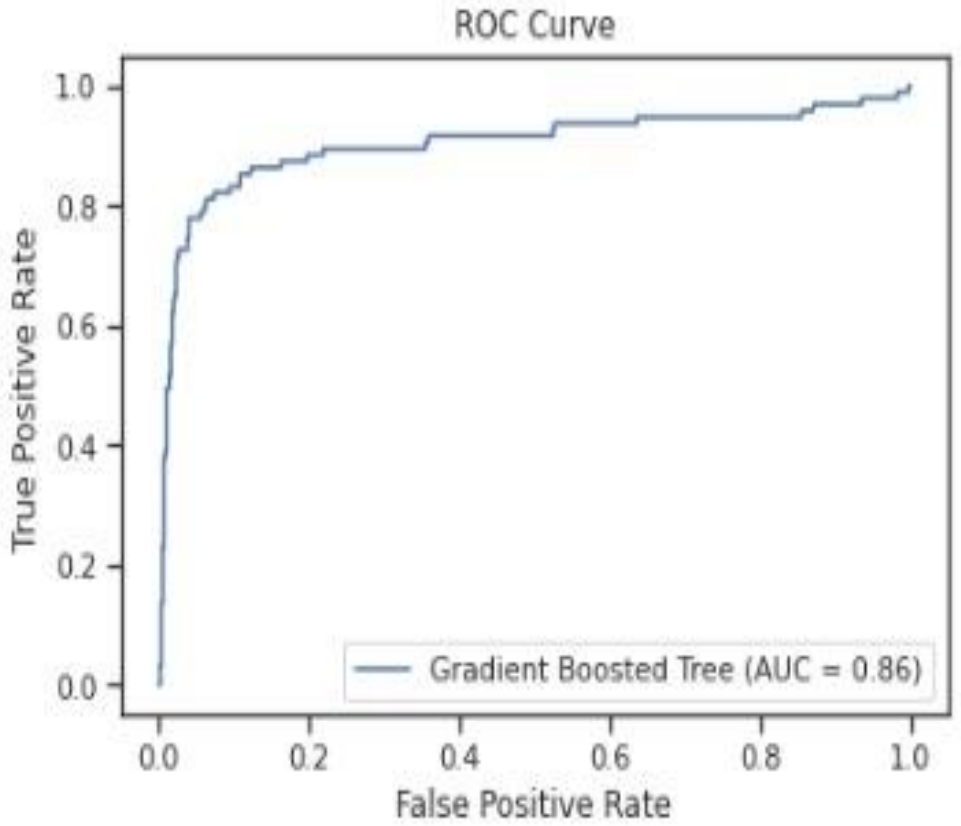
- Gradient Boosting is another ensemble learning method that builds decision trees sequentially, where each tree corrects the errors of the previous one.

- It often provides higher accuracy than Random Forest but may require more computational resources and tuning. However, its ability to capture complex patterns and interactions makes it well-suited for customer churn prediction.

- The ROC curve for Gradient Boosted tree depicts the relationship between False Positive Rate (FPR) and True Positive Rate (TPR), occupying an area of 0.86 under the curve.

- Gradient Boosted Tree classifier achieved an accuracy of 92.05%.

```
Gradient Boosted Tree ROC AUC: 0.8615016562384984
```

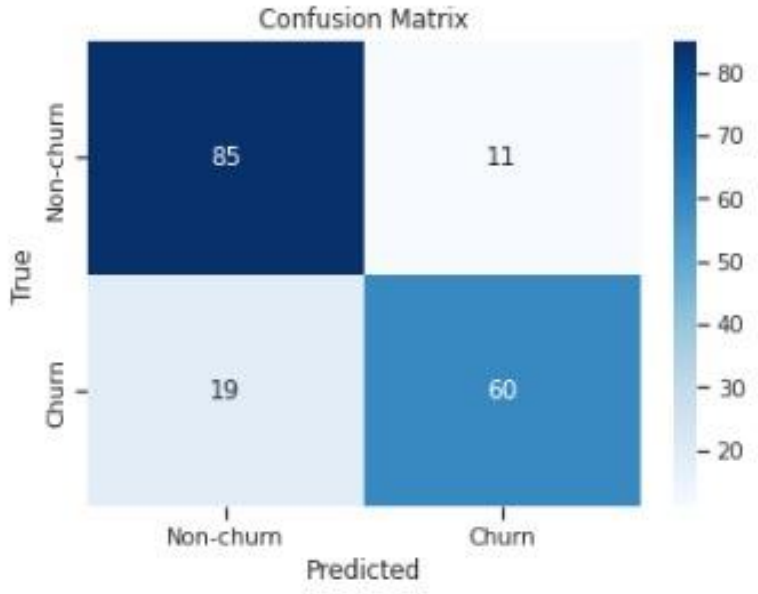


MODEL EVALUATION

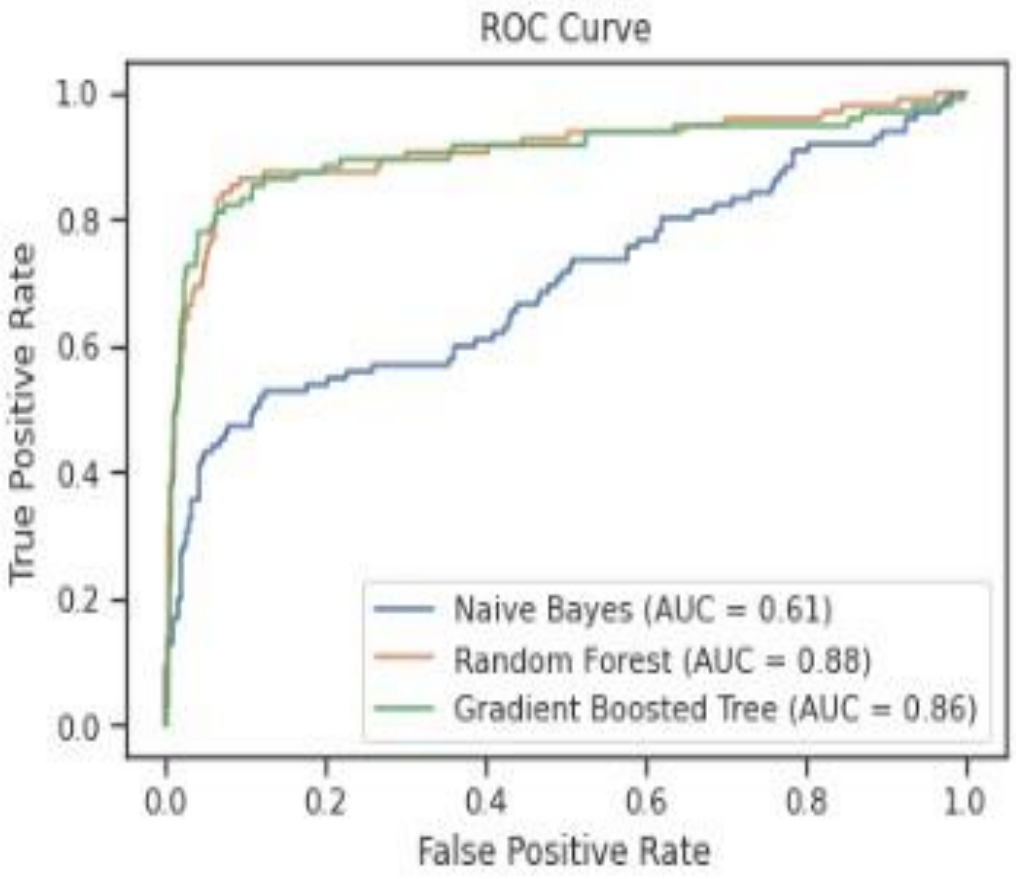
Let's start by exposing the current state of the art for the Spark MLlib and ML functions that were utilized to calculate the metrics:

- **MulticlassMetrics:** This function provides a range of metrics for evaluating multi-class classification models.
- **BinaryClassificationEvaluator:** This function is designed for binary classification tasks. It allows you to calculate two key metrics: the area under the ROC curve (AUC-ROC).
- **MulticlassClassificationEvaluator:** This function evaluates multi-class classification models by computing metrics such as the F1 score, weighted precision, weighted recall, and accuracy.

We have used Confusion matrix to classify the churn and non-churn customers :



RESULTS



Classifier name	underROC_train	underROC_test	Accuracy_train	Accuracy_test	f1_train	f1_test	wPrecision_train	wPrecision_test	wRecall_train	wRecall_test
NB	0.602	0.607	0.602	0.537	0.602	0.537	0.603	0.810	0.602	0.537
RF	0.889	0.878	0.889	0.889	0.889	0.889	0.892	0.918	0.889	0.889
GBT	0.947	0.862	0.947	0.921	0.947	0.921	0.952	0.925	0.947	0.921

The above table displays performance metrics for three classifiers: Naive Bayes (NB), Random Forest (RF), and Gradient Boosted Trees (GBT). It includes metrics such as area under the ROC curve (underROC) for both training and testing sets, accuracy, F1-score, weighted precision, and weighted recall for each classifier on both training and testing data.

CONCLUSION

- In conclusion, customer churn prediction with PySpark has offered a powerful and scalable approach to understanding and predicting customer attrition. By leveraging PySpark's distributed computing capabilities, complex datasets can be processed and analyzed efficiently, facilitating the development of robust churn prediction models.
- Through a combination of feature engineering, data transformation, and advanced machine learning algorithms like Random Forest, Gradient Boosted Trees, and others, PySpark allows for comprehensive churn analysis.
- With cross-validation techniques, the reliability and accuracy of these models can be ensured, guiding businesses to proactively identify at-risk customers and develop targeted retention strategies.
- From the 3 models that we have used in conducting the evaluation, Gradient boost tree achieved a higher accuracy of 92.05%.
- Ultimately, PySpark enables organizations to make data-driven decisions that can significantly reduce churn and improve customer retention.

FUTURE SCOPE

- In the future, we can construct an ETL pipeline that extracts the data from an SQL server, transforms it using an Apache Spark environment like Databricks and loading the data into PostgreSQL.

- Also, the future scope of our project involves leveraging advanced analytics techniques, real-time monitoring capabilities, and integration with external data sources to enhance customer churn prediction accuracy, optimize retention strategies, and drive sustainable business growth.

- We can also deploy various other machine learning models such as Decision tree and SVC in churn prediction.

stevens.edu

THANK YOU