Model Building & Evaluation

In our project, we employed two fundamental machine learning techniques: classification and boosting, each offering distinct modeling approaches.

Classification: Within the classification framework, we embraced both **linear and non-linear models** to capture various aspects of our data.

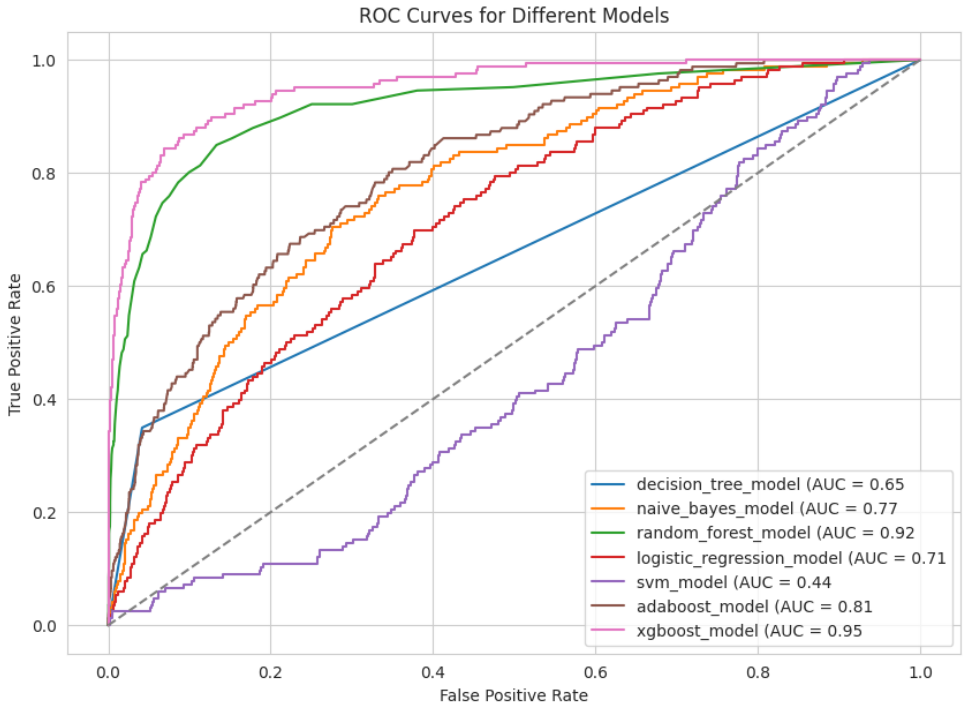
Linear models, such as **Logistic Regression** and **Support Vector Machines (SVM)**, excel in discerning linear relationships within the data. They are adept at identifying linear decision boundaries between classes.

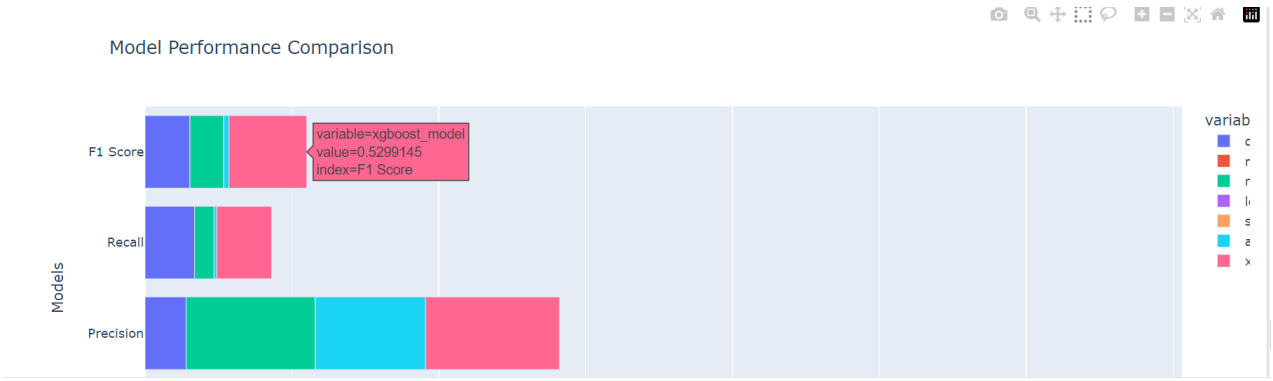
For **non-linear** complexities in the data, we leveraged **Decision Trees**, **Naive Bayes**, and **Random Forest**. Decision Trees hierarchically partition the feature space, allowing for the capture of non-linear decision boundaries. Naive Bayes applies probabilistic reasoning and is particularly effective when features exhibit strong conditional independence. Random Forest, a powerful ensemble of decision trees, excels in capturing complex non-linear relationships.

Boosting: In addition to classification, we ventured into boosting techniques, specifically **AdaBoost and XGBoost**. Boosting is an **ensemble method that combines multiple weak learners to create a strong learner**. It iteratively focuses on misclassified instances, correcting the model's mistakes, and enhancing overall predictive performance. AdaBoost, with its weighted emphasis on misclassified samples, and XGBoost, a scalable and efficient gradient boosting algorithm, serve as potent tools to improve model accuracy and generalization.

Model Evaluation: To gauge the performance of our models, we employed two key evaluation tools: **ROC (Receiver Operating Characteristic)** curves and **confusion matrices**. ROC curves graphically illustrate a model's ability to distinguish between classes across various threshold settings. This is crucial in understanding the trade-offs between true positive and false positive rates.

Concurrently, confusion matrices offer a comprehensive summary of a model's performance. They allow us to assess the correctness of class predictions, considering true positives, true negatives, false positives, and false negatives. These metrics are instrumental in determining the model's predictive accuracy and its capacity to minimize classification errors.





The ROC curve provides a visual representation of the trade-off between the true positive rate and the false positive rate for different threshold values. In this case, a higher AUC indicates better model performance, but the model seems to struggle with detecting positive instances, as seen in the confusion matrix.

The table presents the **AUC (Area Under the Curve) scores**, which measure the performance of different models in predicting customer purchase intent. **A higher AUC score indicates better predictive accuracy, and it serves as a critical metric for assessing the quality of a model.**

Model Name	AUC
Decisoon Tree	0.65
Naïve Bayes	0.77
Random Forest	0.92
Logistic regression	0.71
Support vector machine	0.44
AdaBoost	0.81
XGBoost	**0.95**

Among the models evaluated, the **XGBoost model has demonstrated the highest AUC score of 0.95**. This exceptional performance suggests that XGBoost excels in distinguishing between customers who are "Willing to Buy" and those who are "Not Willing to Buy." Let's delve into why this is noteworthy:

XGBoost leverages a sophisticated ensemble technique, combining the predictive power of multiple decision trees to create a robust and accurate model. This ensemble approach allows it to capture intricate patterns within the data that might be challenging for individual decision trees to discern.

Furthermore, XGBoost's scalability and computational efficiency make it suitable for handling complex datasets with a large number of features and observations. This capability enables the model to exploit data nuances effectively.

In addition to its formidable predictive capabilities, the exceptional AUC score of 0.95 reflects XGBoost's ability to strike a balance between true positive and false positive rates, optimizing the model's ability to make accurate predictions while minimizing errors. This balance is crucial in real-world applications where misclassifications can have significant consequences.

The outstanding performance of XGBoost in this context is a testament to its versatility and adaptability in solving complex classification problems. Its ability to harness the predictive power of multiple decision trees while maintaining a balanced trade-off between true positives and false positives positions XGBoost as an excellent choice for predictive modeling in scenarios where precision and accuracy are paramount.