1001 - IRANIAN CHURN

Team Nash

- Amirtesh Raghuram
- Nigam Parida
- Shreya Pattanayak
- Harish Sathyanandan

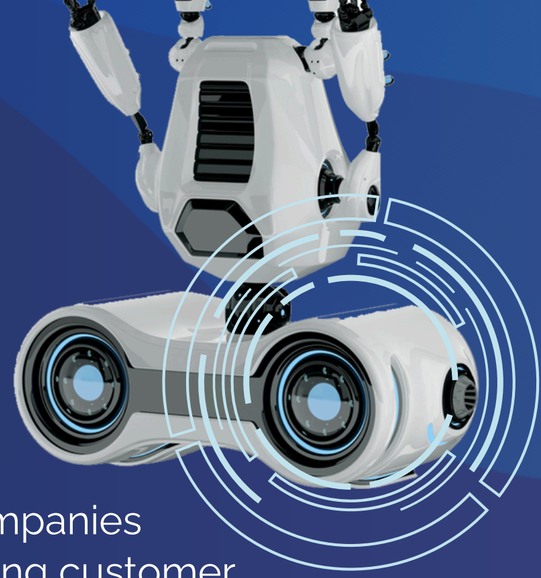
Transforming the Future

Guide’s Name

Nikhil Maurya

02

Goal



1. Customer Retention: Understanding churn patterns helps telecom companies devise strategies to retain customers, reducing revenue loss and enhancing customer loyalty.

2. Data-Driven Insights: Analyzing the dataset can uncover key insights about customer behavior and preferences, enabling better decision-making and targeted marketing strategies.

3. Service Improvement: Identifying common reasons for churn allows companies to address service shortcomings and improve overall customer satisfaction.

4. Competitive Advantage: Utilizing advanced analytics on churn data equips companies with a better understanding of their market, allowing them to potentially outperform competitors who may not be leveraging such insights.

5. Resource Optimization: Investing in churn prediction models can help allocate resources more efficiently, focusing efforts on at-risk customers and improving marketing effectiveness through targeted campaigns.

Exploratory Data Analysis

Basic Information

Statistical Summary:

	Call Failure	Complains	...	Customer Value	Churn
count	3150.000000	3150.000000	...	3150.000000	3150.000000
mean	7.627937	0.076508	...	470.972916	0.157143
std	7.263886	0.265851	...	517.015433	0.363993
min	0.000000	0.000000	...	0.000000	0.000000
25%	1.000000	0.000000	...	113.801250	0.000000
50%	6.000000	0.000000	...	228.480000	0.000000
75%	12.000000	0.000000	...	788.388750	0.000000
max	36.000000	1.000000	...	2165.280000	1.000000

Data Types:

Call Failure	int64
Complains	int64
Subscription Length	int64
Charge Amount	int64
Seconds of Use	int64
Frequency of use	int64
Frequency of SMS	int64
Distinct Called Numbers	int64
Age Group	int64
Tariff Plan	int64
Status	int64
Age	int64
Customer Value	float64
Churn	int64

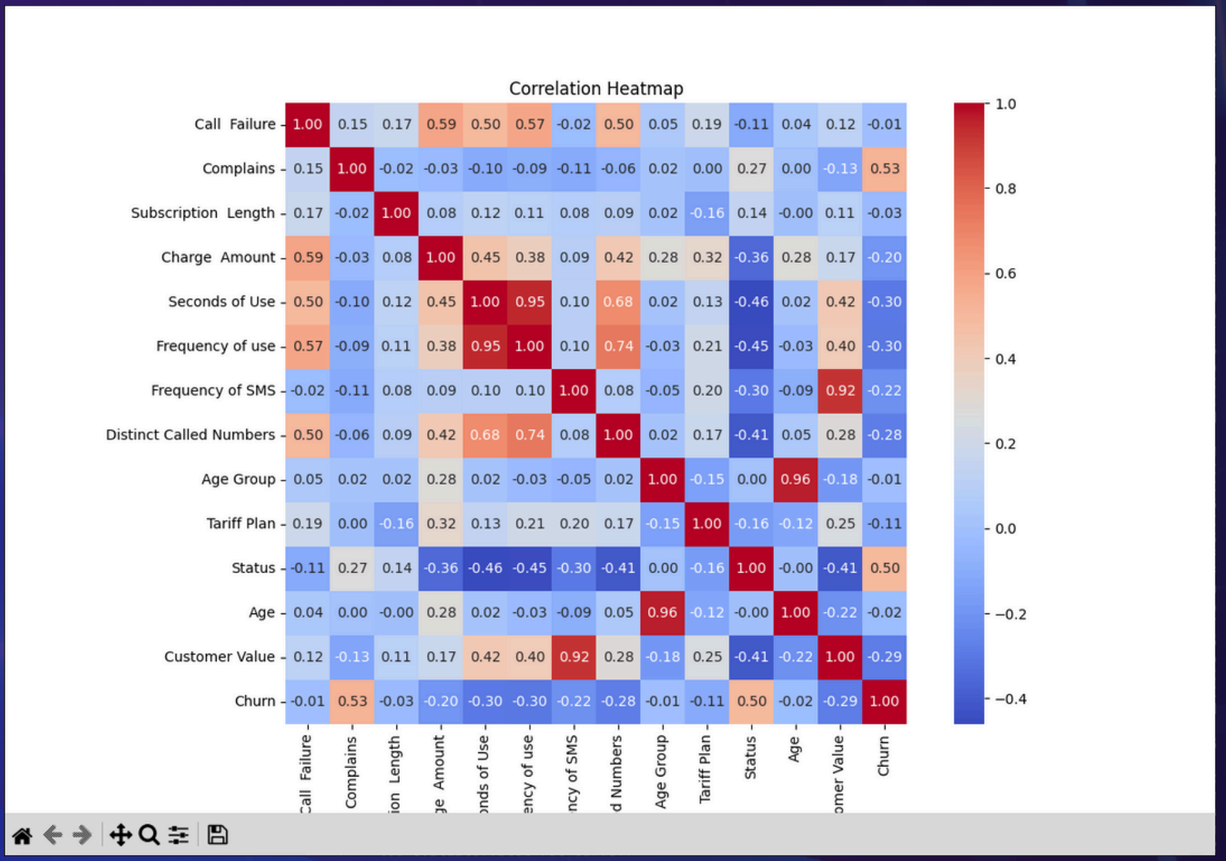
dtype: object

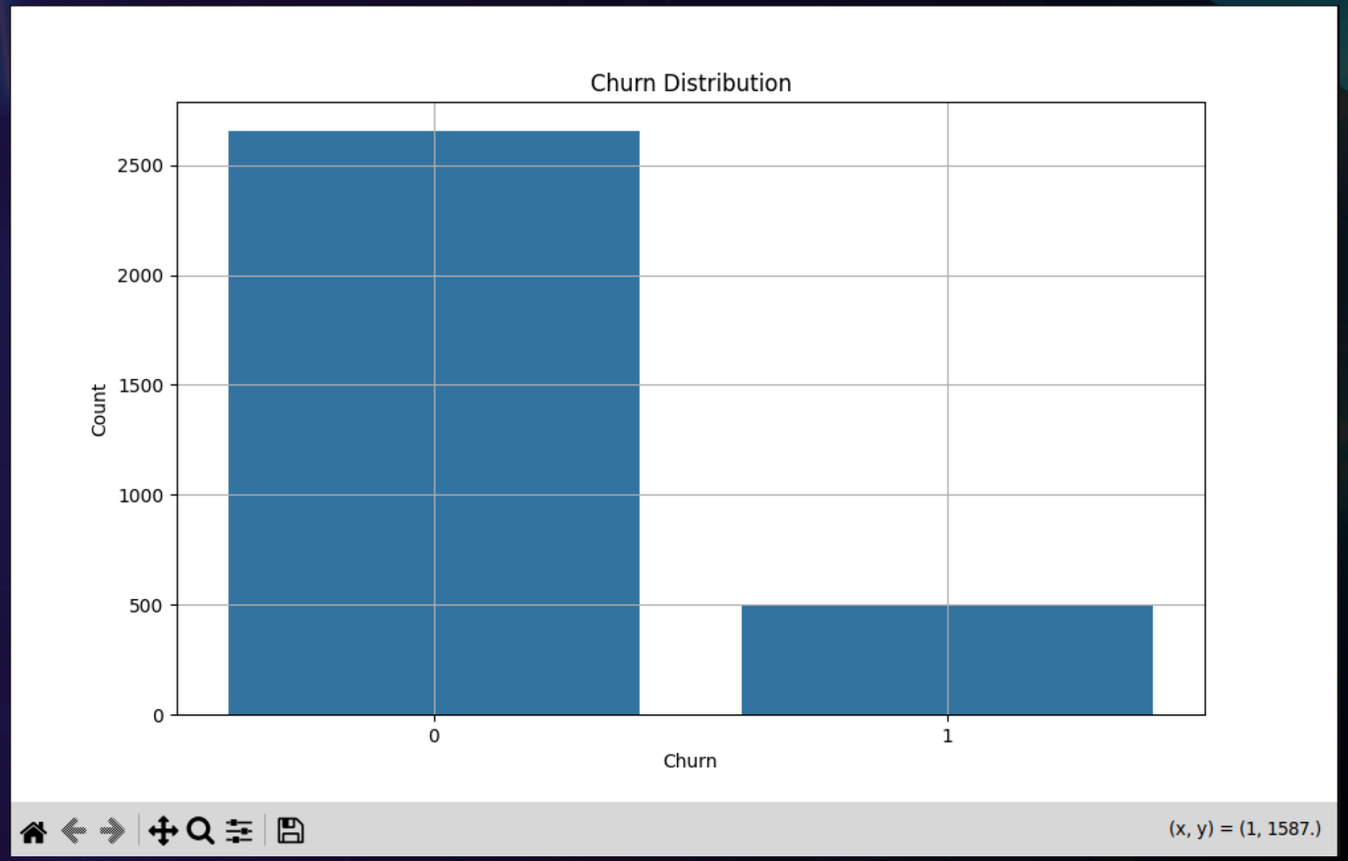
Missing Values:

Call Failure	0
Complains	0
Subscription Length	0
Charge Amount	0
Seconds of Use	0
Frequency of use	0
Frequency of SMS	0
Distinct Called Numbers	0
Age Group	0
Tariff Plan	0
Status	0
Age	0
Customer Value	0
Churn	0

dtype: int64

Exploratory Graphs





The images shows two visualizations. On the left is a correlation heatmap of various features like "Call Failure" and "Customer Value," where colors represent the strength of relationships between variables. On the right is a bar chart showing the distribution of churn (customers who left) and non-churn (customers who stayed), with significantly more non-churners than churners.

The heatmap shows the correlation between different variables, with values ranging from -1 to 1. Red areas indicate a strong positive correlation (variables increase together), while blue areas show a negative or weak correlation (one variable increases while the other decreases or remains unaffected). For example, "Customer Value" and "Status" have a strong positive correlation (0.96), while "Customer Value" and "Complains" show a weak negative correlation (-0.03). This helps identify which features are closely related, useful for feature selection and understanding patterns in the dataset.

Data Understanding

The dataset reveals that approximately 15.7% of the customers have churned.

Key insights include:

- Average call failures are around 7.63, with a minority of customers (7.6%) registering complaints.
- The average subscription length is 32.5 months, and customers use about 4,472 seconds on calls, contacting an average of 23 distinct numbers.
- SMS usage shows significant variability, with an average of 73 messages sent.
- The average customer value is $470, but there’s a wide range, with some customers contributing over $2,100.
- Most customers fall in the 25-30 age range.

Training of the Model

To create a well performing model we have tried to train the model on various classes present in scikit learn module of Python and XGBoost module of Python. The following models were used:

- Logistic regression
- Random forest classifier
- Voting classifier
- XGB classifier
- MLP classifier (Multi layer perceptron, similar to neural networks)

Parameters Used in Every Model

1.Logistic regression: Proceeded with default parameters

2.Random forest classifier: n_estimators=300, max_depth=10,bootstrap=True

3.XGBoost classifier: n_estimators=1000, learning_rate=0.01, gamma=0.01

4.Voting classifier: estimators=[('rf',rf_model),('xgb',xgb_model)],weights=[1,1],voting='soft'

5.MLP classifier: hidden_layer_sizes=[256,128,64],batch_size=32,learning_rate_init=0.001,shuffle=True

Training Progress

```
Logistic regression results on training data:
Accuracy: 0.8940
Precision: 0.8073
Recall: 0.4389
F1 Score: 0.5687
AUC-ROC: 0.9340
Logistic regression results on testing data:
Accuracy: 0.8921
Precision: 0.7407
Recall: 0.4255
F1 Score: 0.5405
AUC-ROC: 0.9349
```

```
MLP classifier results on training data:
Accuracy: 0.9849
Precision: 0.9223
Recall: 0.9845
F1 Score: 0.9524
AUC-ROC: 0.9989
MLP classifier results on testing data:
Accuracy: 0.9571
Precision: 0.8796
Recall: 0.8716
F1 Score: 0.8756
AUC-ROC: 0.9851
```

```
Random forest results on training data:
Accuracy: 0.9833
Precision: 0.9536
Recall: 0.9391
F1 Score: 0.9463
AUC-ROC: 0.9985
Random forest results on testing data:
Accuracy: 0.9508
Precision: 0.8804
Recall: 0.8020
F1 Score: 0.8394
AUC-ROC: 0.9837
```

```
XGBoost results on training data:
Accuracy: 0.9853
Precision: 0.9436
Recall: 0.9649
F1 Score: 0.9542
AUC-ROC: 0.9988
XGBoost results on testing data:
Accuracy: 0.9667
Precision: 0.8788
Recall: 0.9062
F1 Score: 0.8923
AUC-ROC: 0.9910
```

```
Voting classifier results on training data:
Accuracy: 0.9861
Precision: 0.9786
Recall: 0.9311
F1 Score: 0.9542
AUC-ROC: 0.9989
Voting classifier results on testing data:
Accuracy: 0.9587
Precision: 0.9326
Recall: 0.8058
F1 Score: 0.8646
AUC-ROC: 0.9936
```

Process of Evaluation

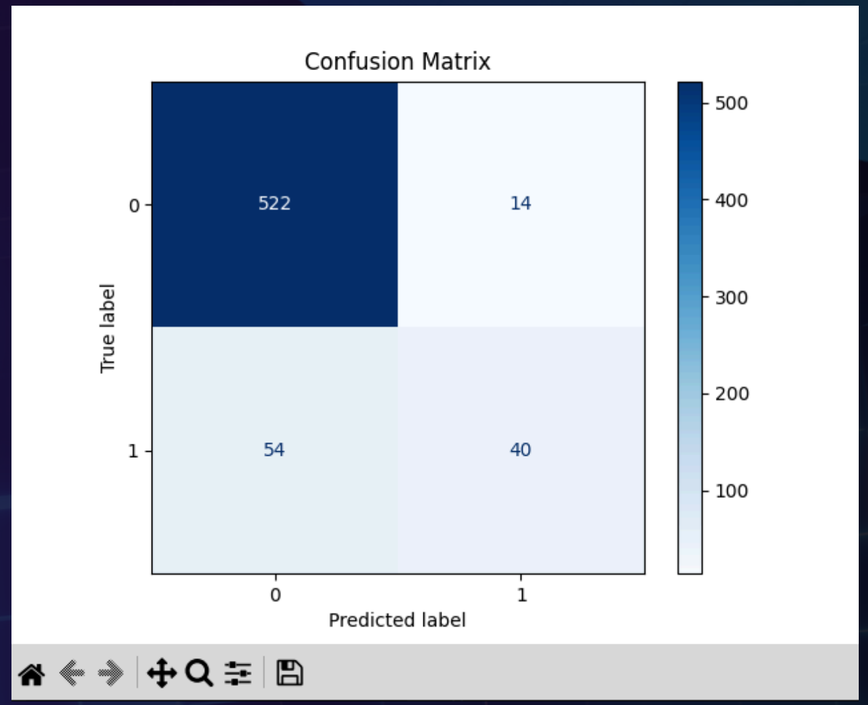
A confusion matrix in Scikit-learn is a tool used to evaluate the performance of classification models. It is structured as an N times N grid, where N represents the number of classes. Each cell in the matrix indicates the count of predictions made by the model, helping to identify true positives, false positives, true negatives, and false negatives. This detailed breakdown allows for a deeper understanding of model performance beyond simple accuracy metrics, enabling the calculation of precision, recall, and F1 score.

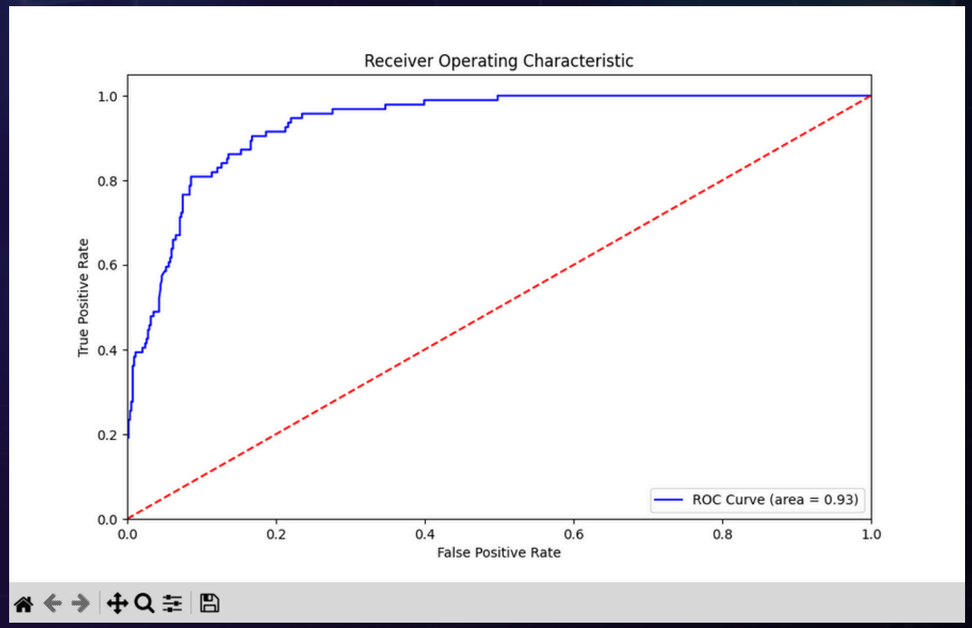
A ROC (Receiver Operating Characteristic) curve is a graphical representation of a classification model's performance across various thresholds. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR), allowing for visual assessment of trade-offs between sensitivity and specificity. The Area Under the Curve (AUC) quantifies the model's ability to distinguish between classes, with values ranging from 0 to 1; higher AUC indicates better performance. An AUC of 0.5 suggests no discriminative power, while values closer to 1 indicate strong class separability.
In our evaluation, the ROC curve has been plotted with the area value displayed at the bottom which can indicate how well our model is performing on unseen data.

Evaluating Models

Logistic Regression Model

```
========================================
Evaluating Model: LogisticRegression
========================================
Accuracy: 0.8921
Precision: 0.7407
Recall: 0.4255
F1 Score: 0.5405
ROC AUC: 0.9349
========================================
```

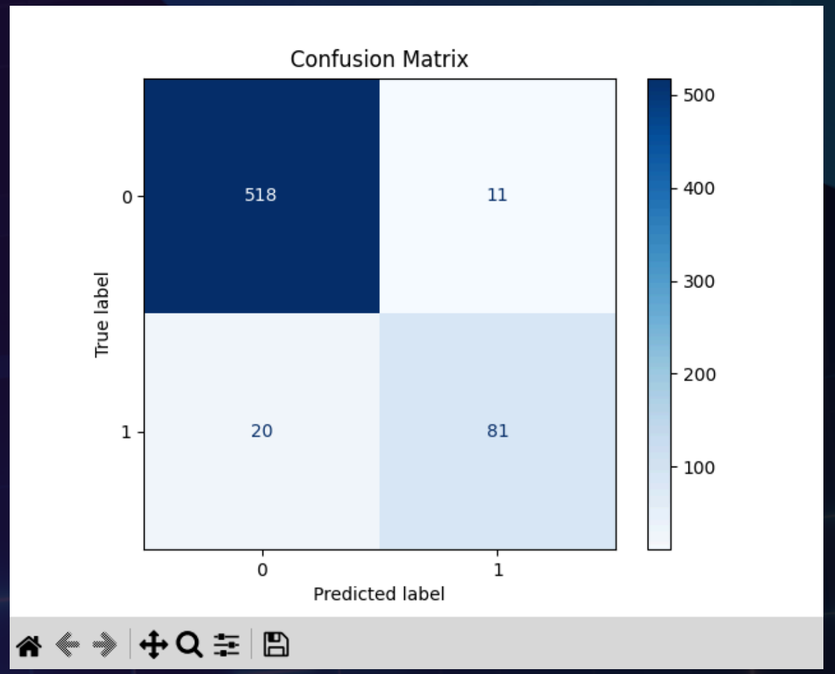


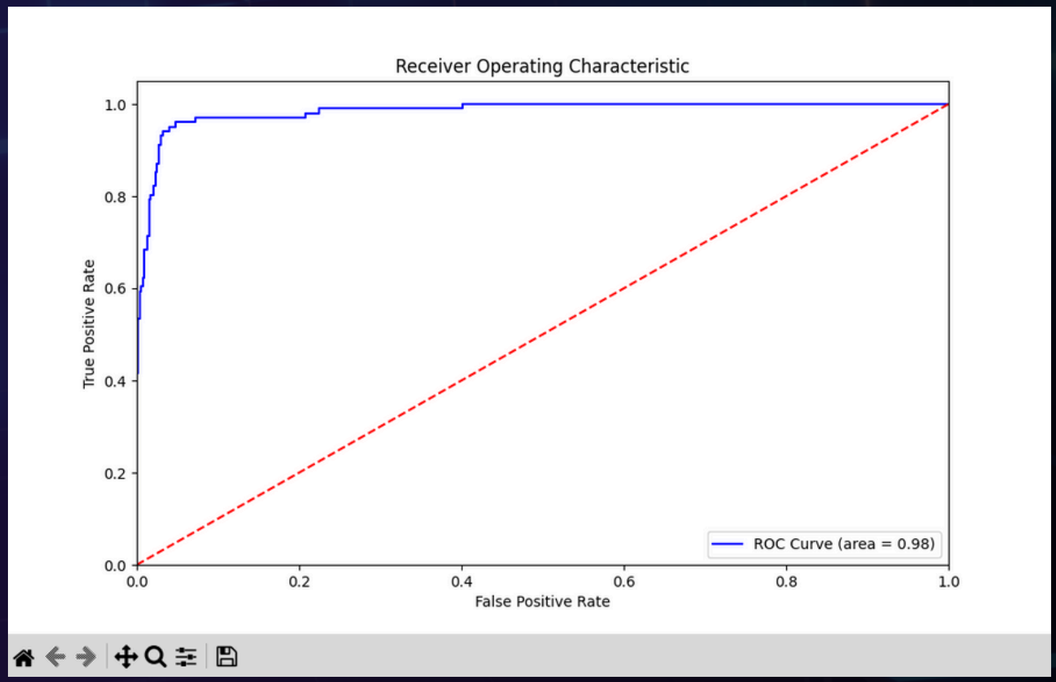


Evaluating Models

Random Forest Classification Model

```
========================================
Evaluating Model: RandomForestClassifier
========================================
Accuracy: 0.9508
Precision: 0.8804
Recall: 0.8020
F1 Score: 0.8394
ROC AUC: 0.9837
========================================
```

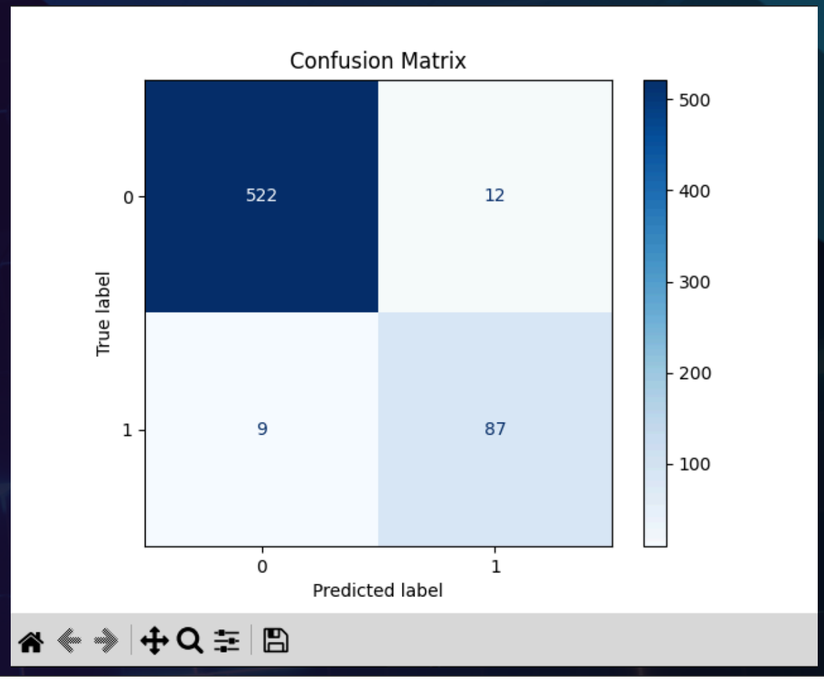


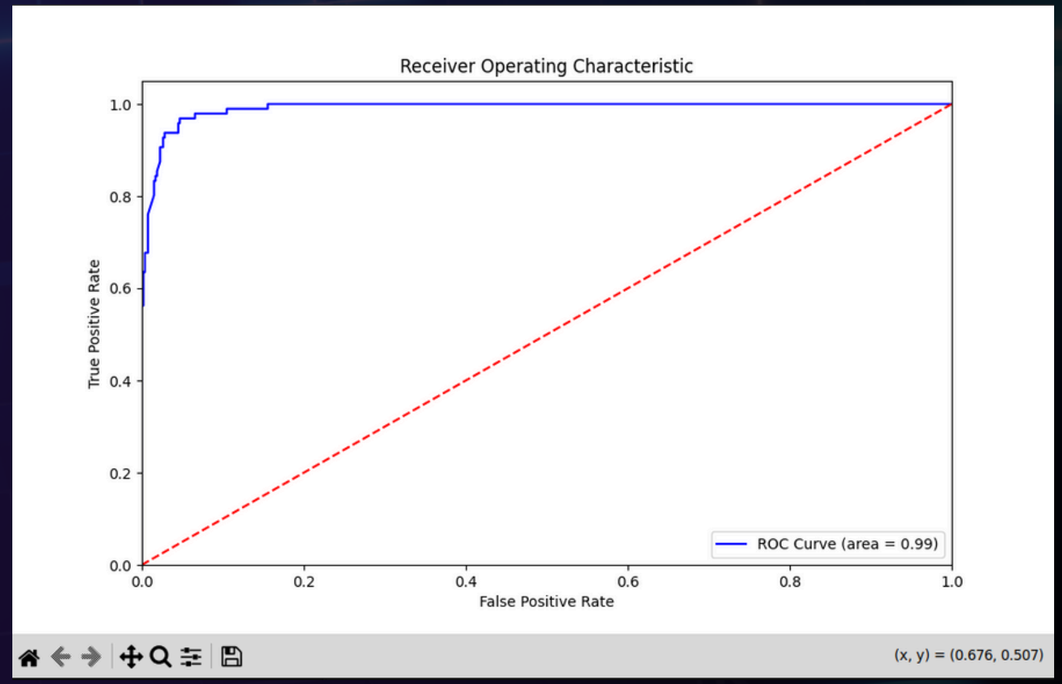


Evaluating Models

XGBoost Classification Model

```
Evaluating Model: XGBClassifier
========================================
Accuracy: 0.9667
Precision: 0.8788
Recall: 0.9062
F1 Score: 0.8923
ROC AUC: 0.9910
========================================
```

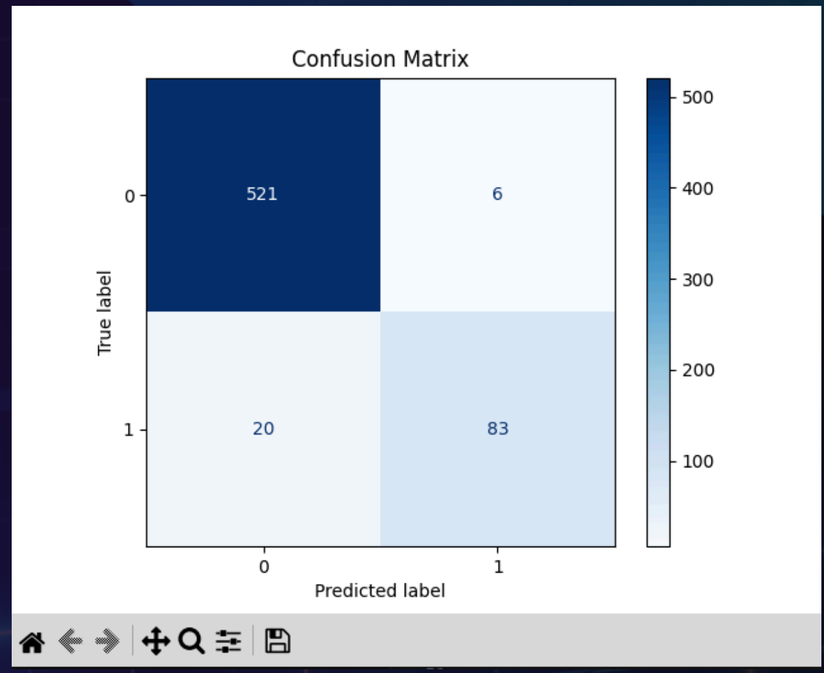


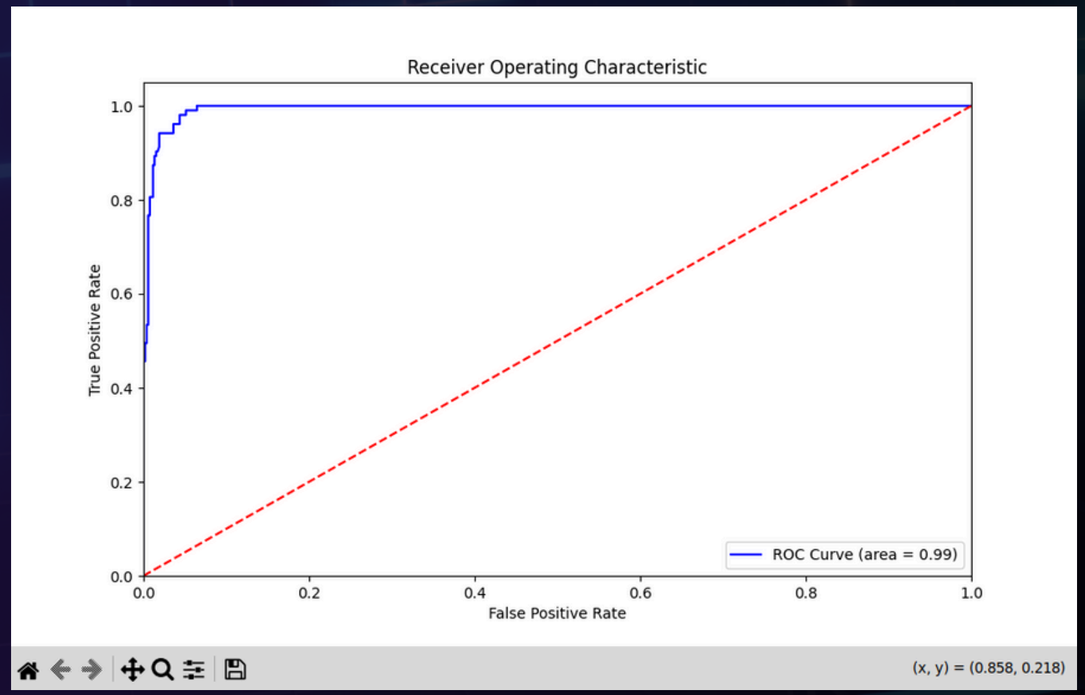


Evaluating Models

Voting Classifier Model

```
========================================
Evaluating Model: VotingClassifier
========================================
Accuracy: 0.9587
Precision: 0.9326
Recall: 0.8058
F1 Score: 0.8646
ROC AUC: 0.9936
========================================
```

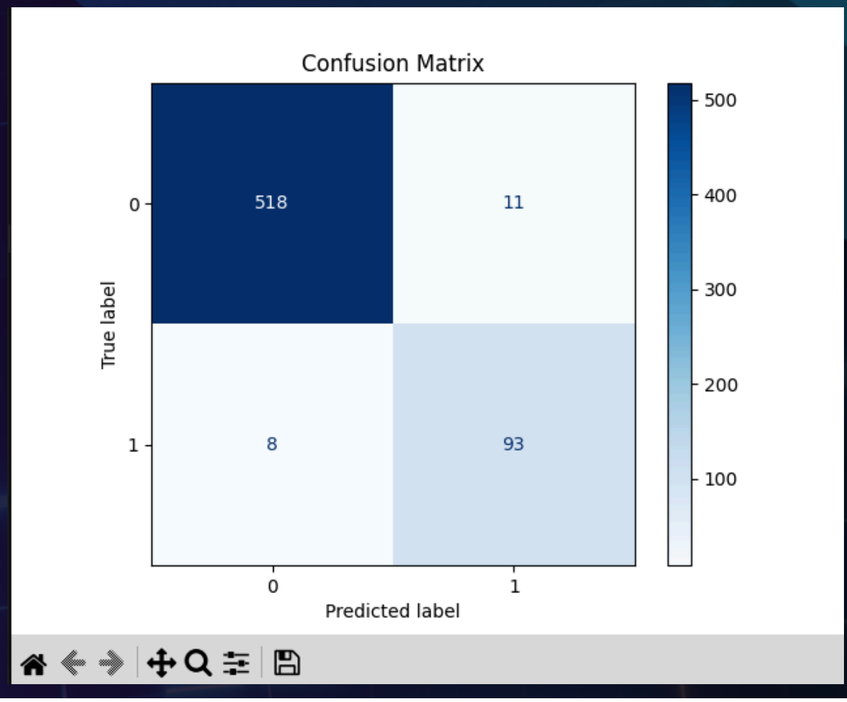


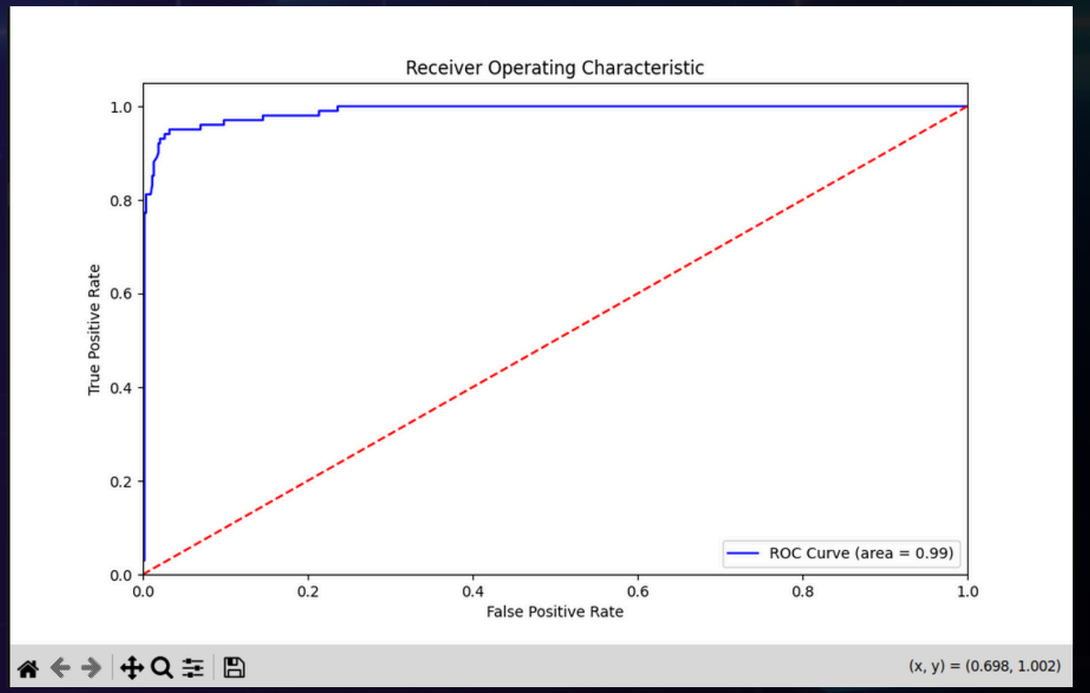


Evaluating Models

MLP Classifier Model

```
Evaluating Model: MLPClassifier
========================================
Accuracy: 0.9571
Precision: 0.8796
Recall: 0.8716
F1 Score: 0.8756
ROC AUC: 0.9851
========================================
```





Comparing all the Models

Model	Accuracy	Precision	Recall	F1 Score	AUC-ROC
MLP Classifier	96.19%	84.11%	92.78%	88.24%	98.84%
XGBoost	95.71%	88.04%	83.51%	85.71%	98.31%
Random Forest	95.08%	85.87%	81.44%	83.60%	98.38%
Voting Classifier	95.71%	88.04%	83.51%	85.71%	98.42%
Logistic Regression	88.57%	73.58%	40.21%	52.00%	91.64%

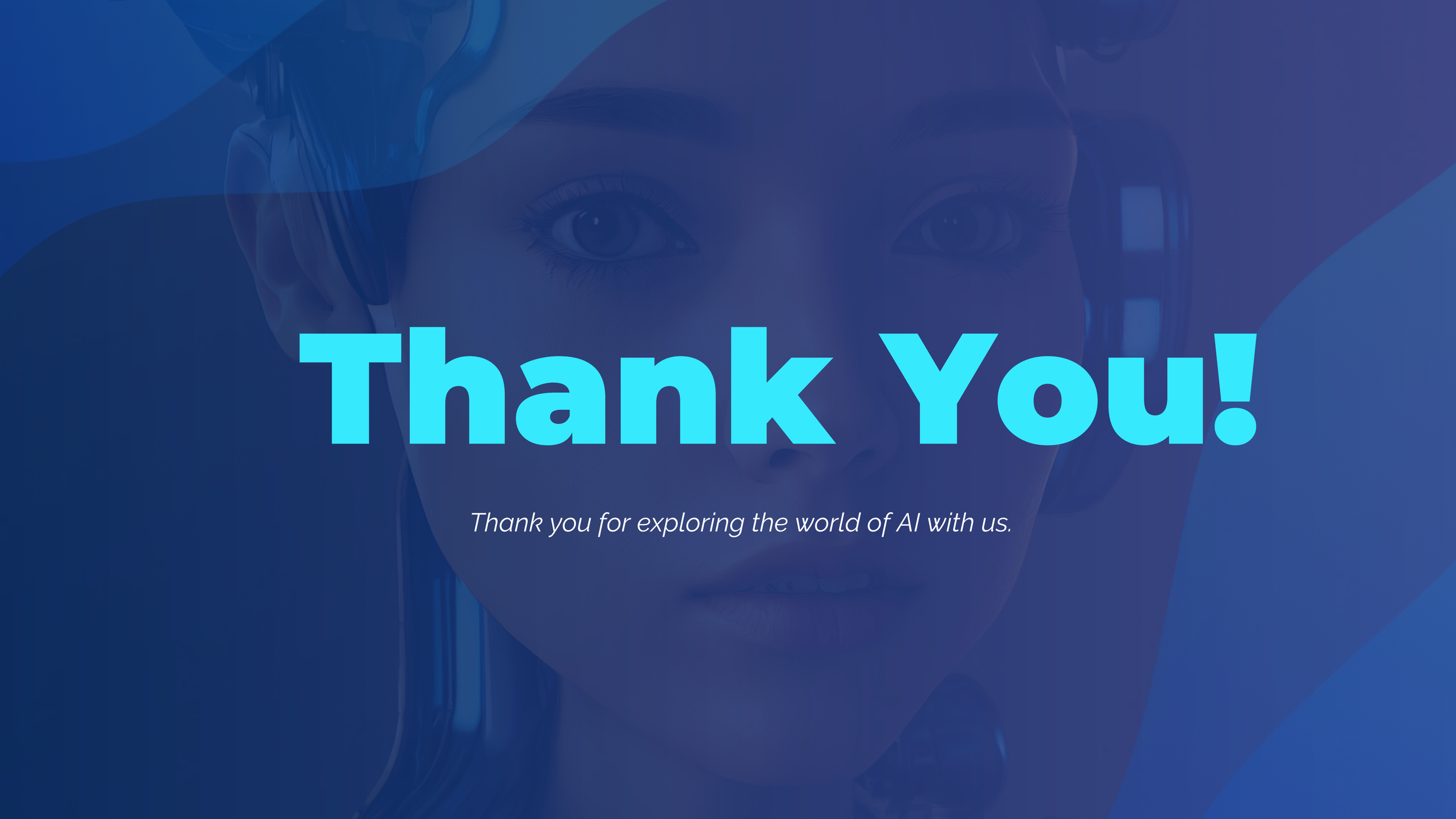
Choosing The Best Model

The MLP Classifier emerges as the top performer in this analysis, excelling both on training and unseen testing data. With the highest accuracy (96.19%) and recall (92.78%), it effectively identifies more true positives than other models, ensuring a strong balance between precision and recall (F1 score: 88.24%). Its impressive AUC-ROC score of 98.84% further highlights its robust ability to distinguish between classes across all thresholds. The MLP’s reliable performance on unseen data demonstrates its generalizability, making it a highly effective model for real-world applications where accurate predictions are essential.

Results of Best Model on Unseen Data

Index	Actual	Predicted
1	1	1
2	0	0
3	0	0
4	0	0
5	1	0
6	0	0
7	0	0
8	0	0
9	0	0
10	1	1
11	0	0
12	1	1
13	0	0
14	1	1
15	0	0
16	0	0
17	1	1
18	0	0
19	0	0
20	0	0

As we can see, the model was used to predict the classes as 0 or 1 on the test data starting from index 1 till 20. Out of 20 prediction, only 1 prediction is wrong which suggests that the model is performing very well on unseen data.



Thank You!

Thank you for exploring the world of AI with us.