Customer Churn Prediction

EXECUTIVE SUMMARY

Customer churn refers to the loss of customers or subscribers for any reason at all. Businesses measure and track churn as a percentage of lost customers compared to the total number of customers over a given time period. This project aims to predict customer churn in a telecommunication business company, by employing machine learning techniques, using data containing customers information and churn status. This project also explores the building of a machine learning web application using the Streamlit framework.

INTRODUCTION

Customer churn is the percentage of customers who stopped purchasing your business's products or services during a certain period of time. Your customer churn rate indicates how many of your existing customers are not likely to make another purchase from your business. A high churn rate indicates that your customers are not satisfied with the products or services you're offering.

The objective of this project are as follows:

- ➢ Explore and analyze dataset to gain insights into factors influencing churn
- ➢ Build a machine learning model to identify potential customer churn.
- ➢ Build a machine learning web application and deploy a model created to identify churn.

This project focuses on predicting customer churn using historical customer data.

METHODOLOGY

DATA COLLECTION AND PREPROCESSING

The dataset used in this project was obtained from the IBM sample data. It contains information about customers churn status, services that each customer signed up for, customer account information and demographic information.

The dataset underwent cleaning to remove missing values, irrelevant columns were dropped and categorical columns one-hot encoded.

FEATURE ENGINEERING

Features such as 'Tenure' were selected and transformed into a bin of 12 months to create a new feature 'Tenure_grp'.

EXPLORATORY DATA ANALYSIS (EDA)

EDA was performed to understand and gain insights from the data set.

DESCRIPTIVE STATISTICS

In [11]:
```
#check the descriptive statistics of numeric variables in the dataset
telco_data.describe()
```

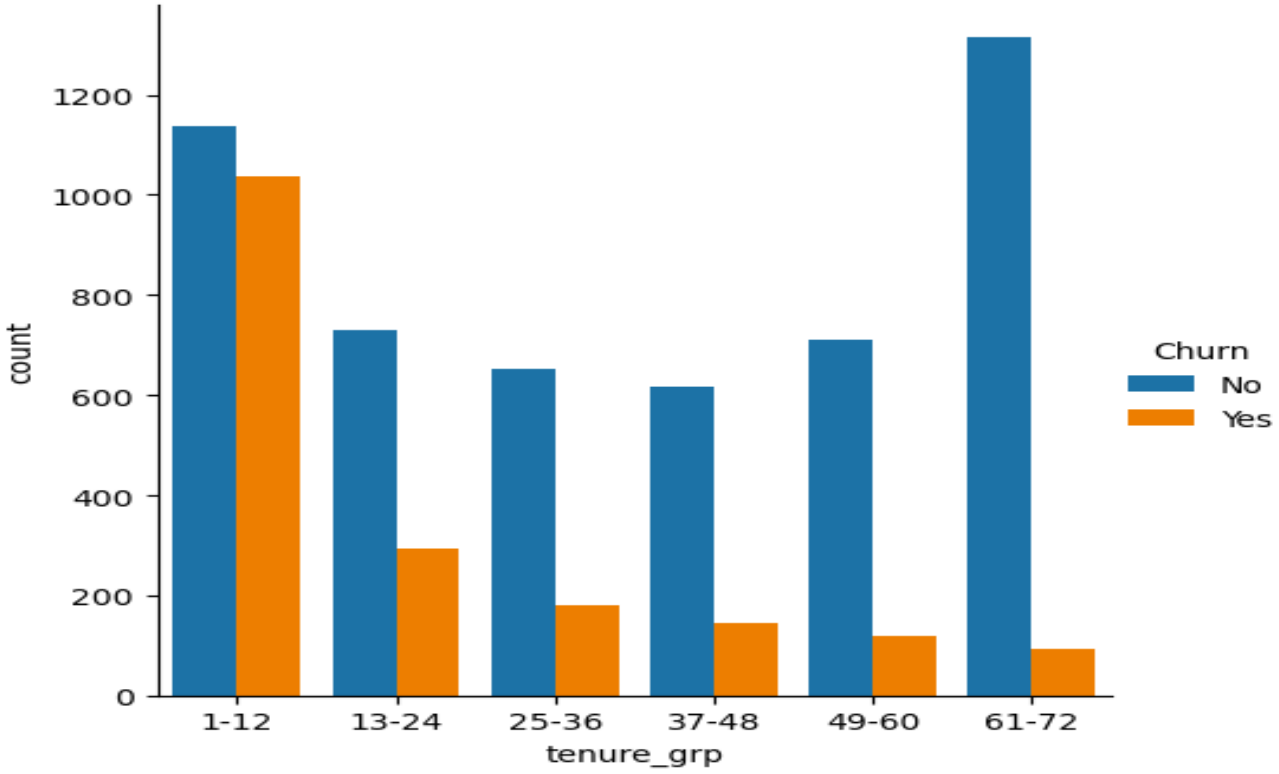
Out[11]:

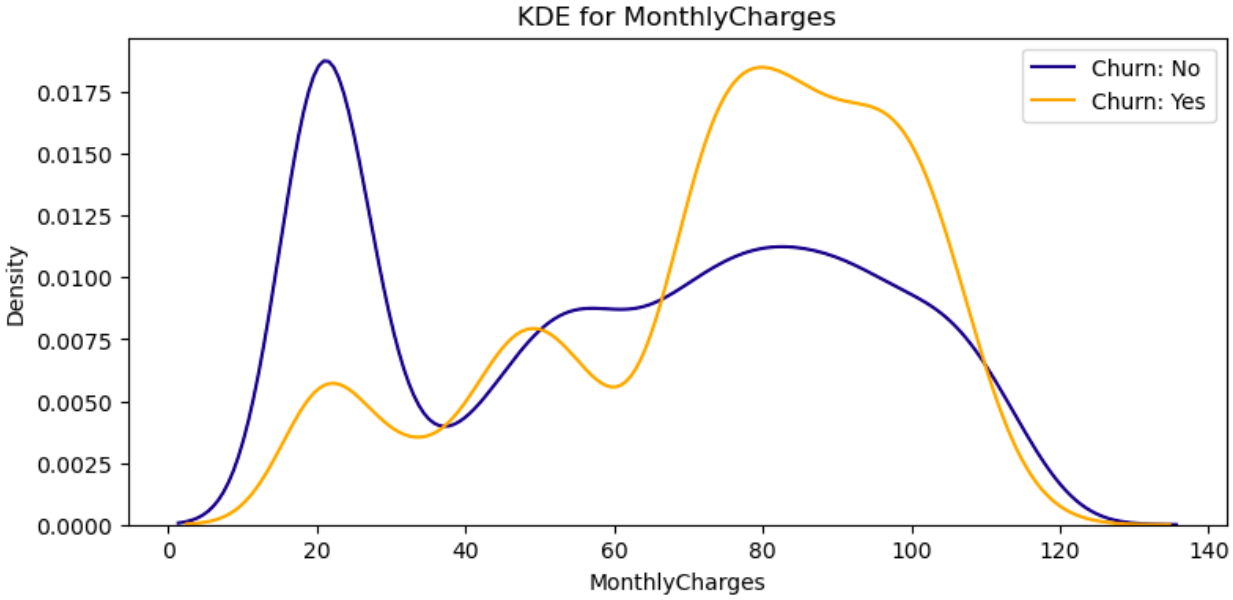
	SeniorCitizen	tenure	MonthlyCharges
count	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692
std	0.368612	24.559481	30.090047
min	0.000000	0.000000	18.250000
25%	0.000000	9.000000	35.500000
50%	0.000000	29.000000	70.350000
75%	0.000000	55.000000	89.850000
max	1.000000	72.000000	118.750000

Insights

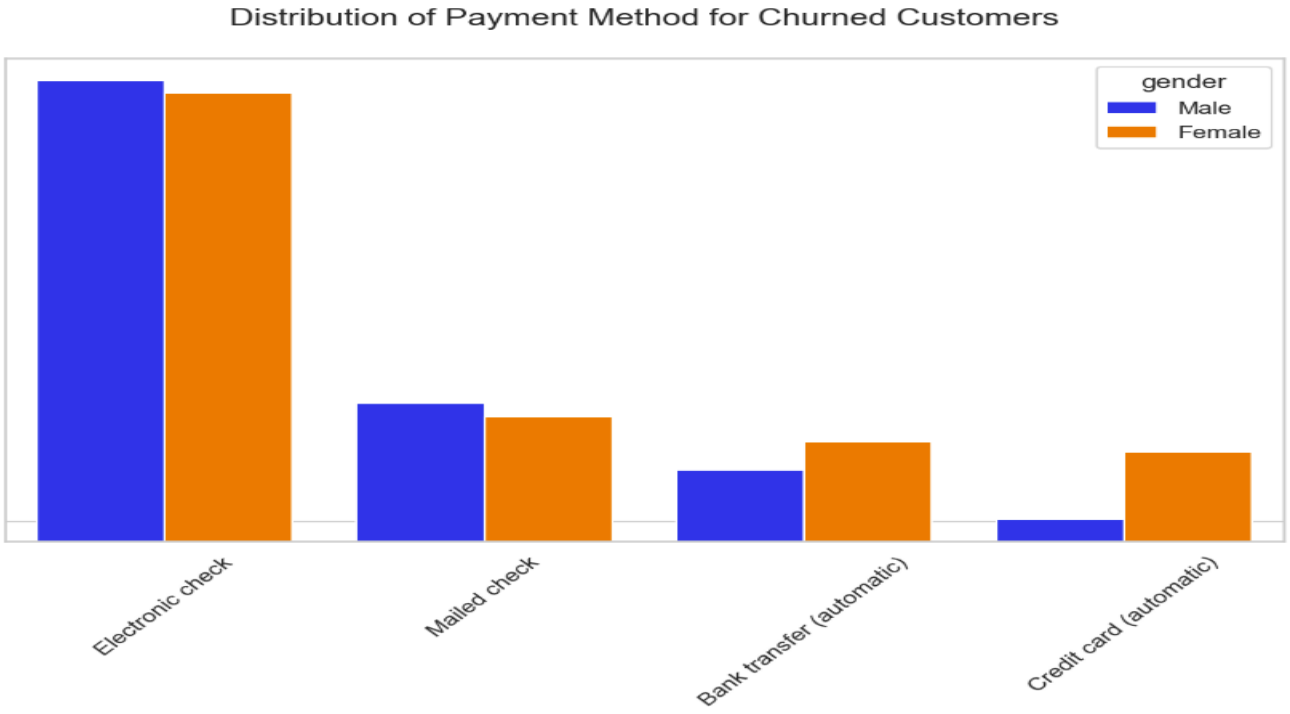
- 'SeniorCitizen' is categorical, hence the 25%-50%-75% distribution is invalid.
- 75% customers have 'tenure' less than 55 months.
- Average monthly charges are USD64 whereas 25% custonmers pay more than USD89.85 per month*

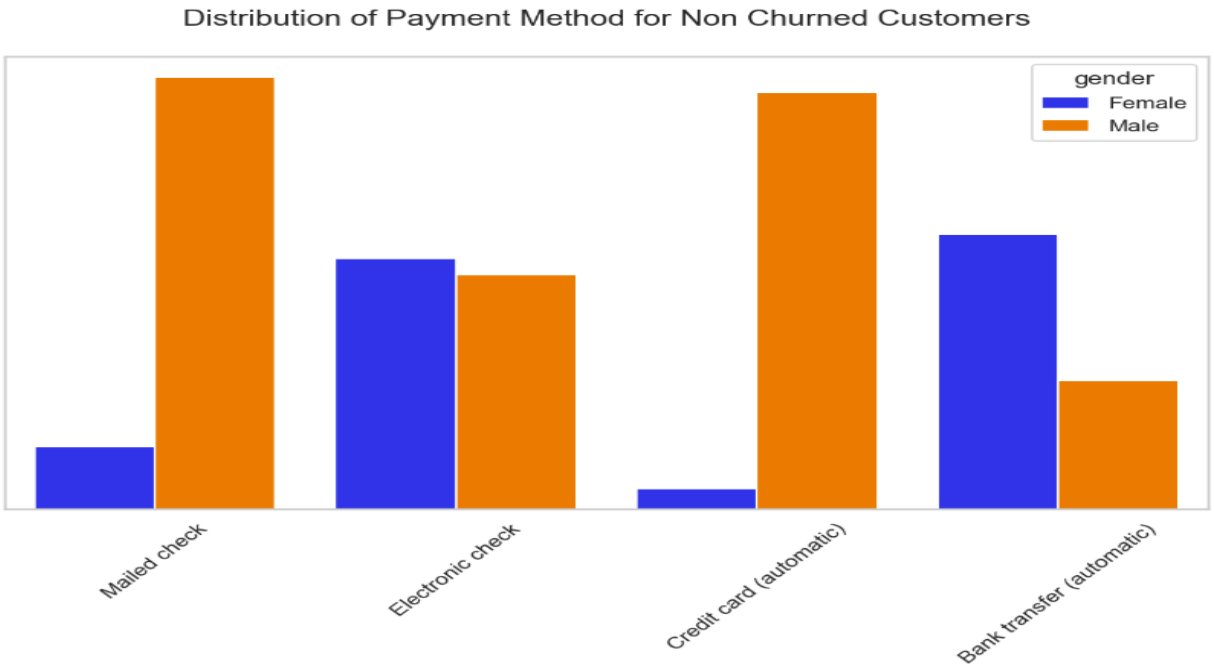
UNIVARIATE ANALYSIS



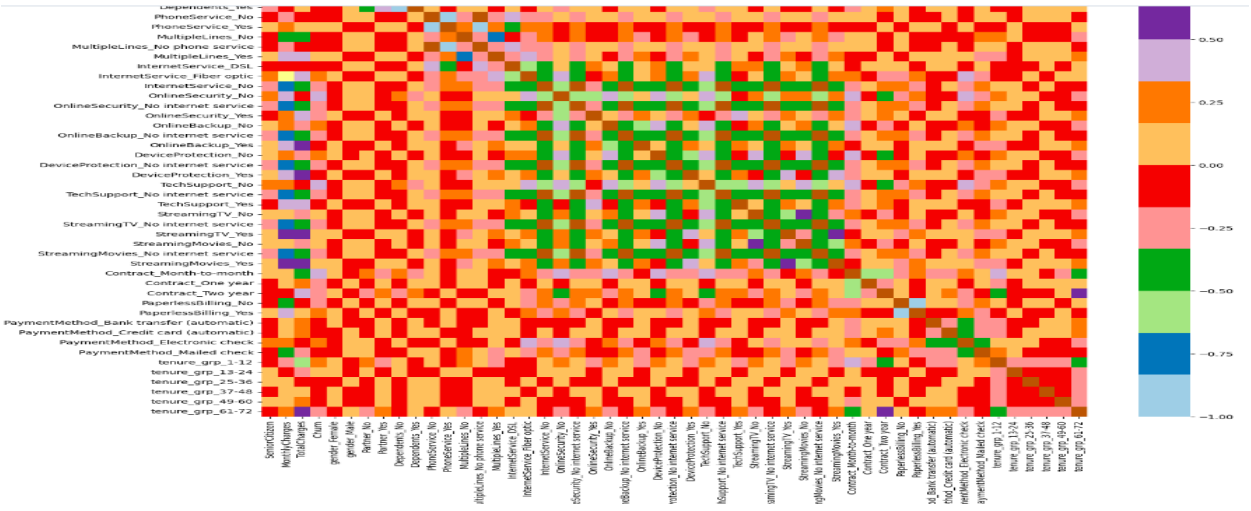


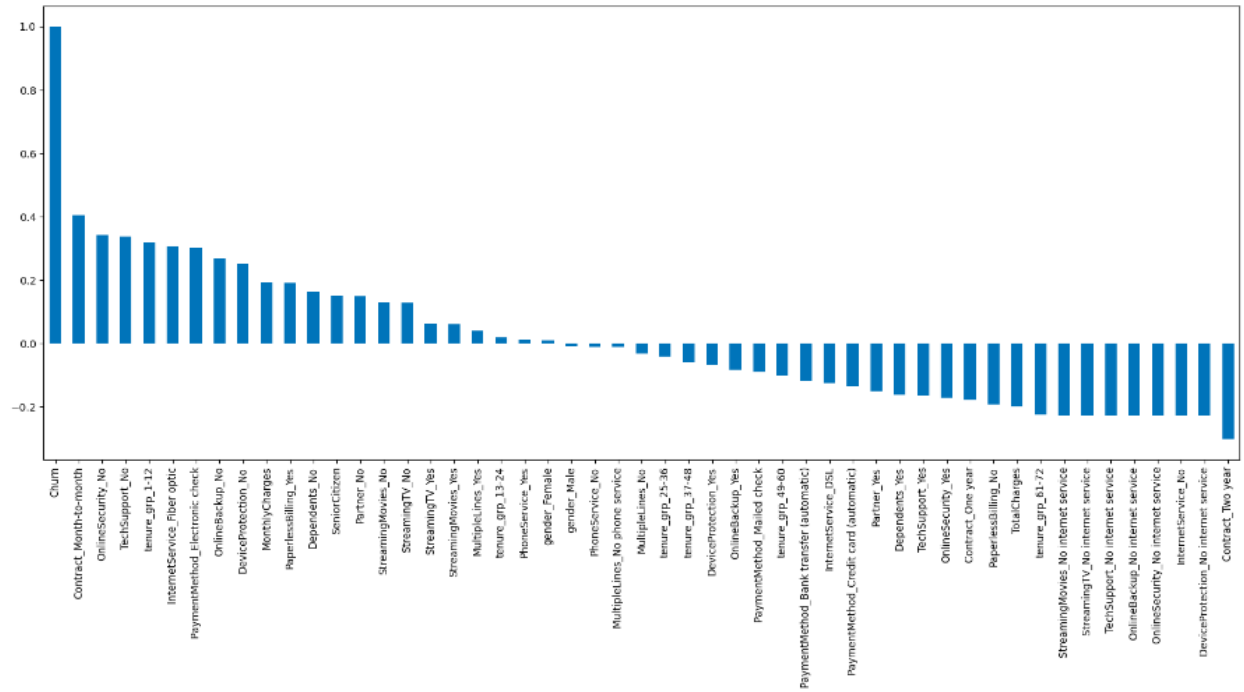
BIVARIATE ANALYSIS





CORRELATION ANALYSIS





Insights

- High Churn when customers are on a **month to month contract, No online security, No tech support, first year subscription, fiber optic internet service, electronic check payment method...**
- Low churn were seen in customers with **Long term contracts, 5 year plan, dependents, partners....**
- **No Phone service, multiple lines and gender** has almost no impact on the churn.

MODEL BUILDING AND EVALUATION

MODEL TRAINING

Dataset was balanced using the **SMOTEENN** technique. The dataset was split into predictive features and target features and further splitted into training set and testing set to train and evaluate the model's performance. Hyperparameter tuning was performed to optimize model accuracy.

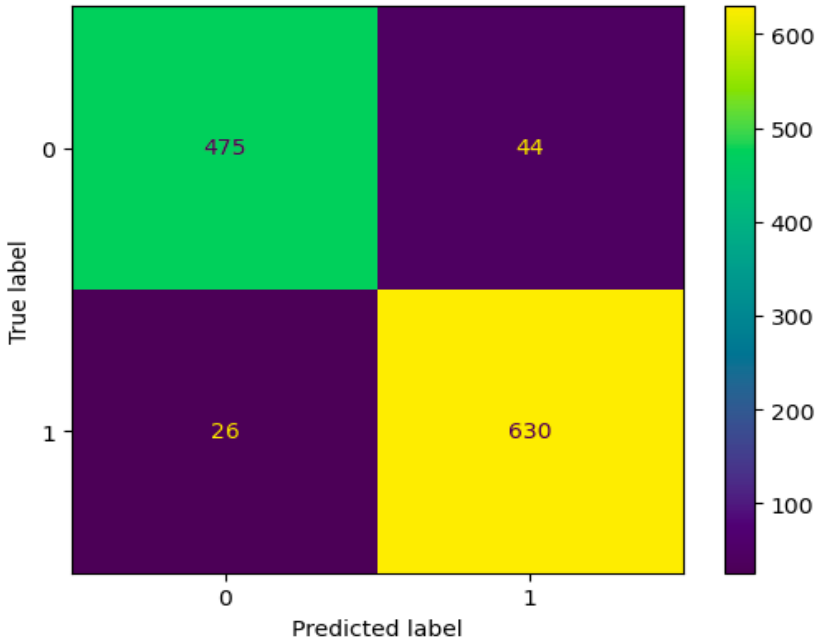
```
# F1-score for class "1":"yes" is poor compared to the other class. This is due to the imbalance dataset with rat
# to solve this problem, Smote-enn is used to balance the dataset by sampling(over/under)

sm=SMOTEENN()
x_resampled, y_resampled = sm.fit_resample(X, Y)
```

```
# split the resampled data

xr_train, xr_test, yr_train, yr_test = train_test_split(x_resampled, y_resampled, test_size=0.2)
```

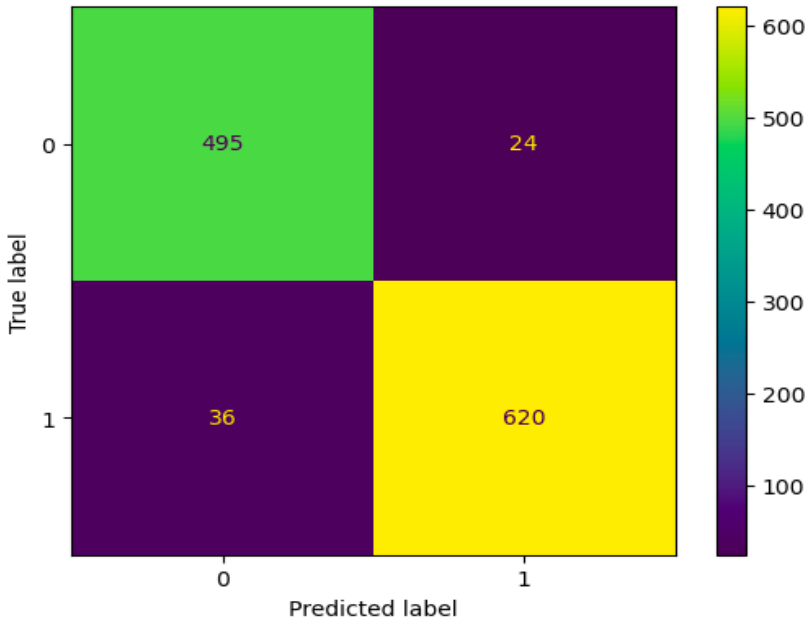
```
clf= RandomForestClassifier(random_state=42)
params={
    "n_estimators": range(50,125,25),
    "max_depth": range(60,81,2)}

rfc_model = GridSearchCV(
    clf,
    param_grid = params,
    cv = 10,
    n_jobs= -1,
    verbose=1)

rfc_model.fit(xr_train,yr_train)
```

```
Fitting 10 folds for each of 33 candidates, totalling 330 fits
GridSearchCV(cv=10, estimator=RandomForestClassifier(random_state=42),
             n_jobs=-1,
```

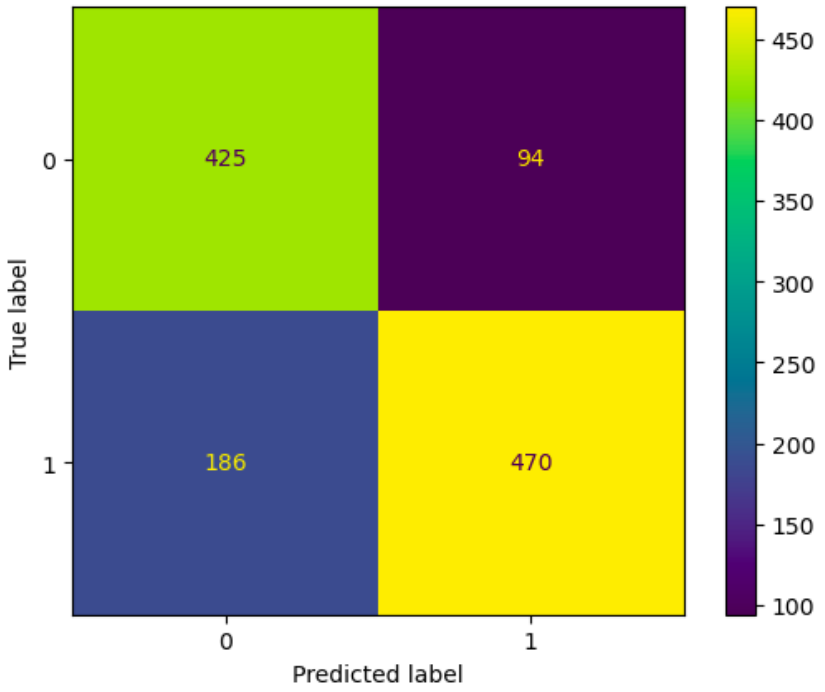
PREDICTIVE ANALYSIS



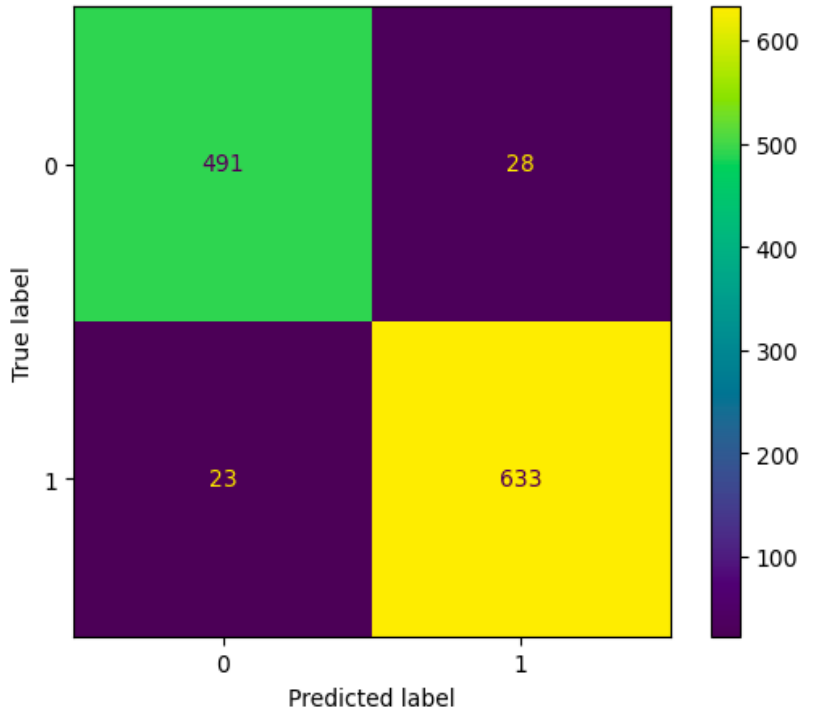
The **Decision Tree Classifier** model had an accuracy of 94.04%



The **Logistic Regression Model** had an accuracy of 94.49%



The **Support Vector Machine** model had an accuracy of 76.17%.



The **Random Forest Classifier** model had an accuracy of 95.66%

MODEL SELECTION

Machine learning algorithms such as: Decision Tree Classifier, Logistic Regression, Support Vector Machine and Random Forest Classifier were considered for the prediction of churn. Final choice of the model was based on model metric performance and interpretability.

In [55]:
```
models = pd.DataFrame({

    "Models": ["Logistic Regression" , "SVM", "DecisionTreeClassifier","RandomForestClassifier"],
    "Score":[lr.score(xr_test,yr_test) ,svm.score(xr_test,yr_test), model_dtc_smote.score(xr_test,yr_test), rfc_m
})
display(models.sort_values(by="Score" , ascending=False))
```

	Models	Score
3	RandomForestClassifier	0.956596
0	Logistic Regression	0.948936
2	DecisionTreeClassifier	0.940426
1	SVM	0.761702

In [60]:
```
print(f"The Random Forest model performed best with a score of {round(rfc_model.score(xr_test,yr_test) * 100, 2)}
```

```
The Random Forest model performed best with a score of 95.66% accuracy.
```

BUILDING THE STREAMLIT APPLICATION

Streamlit is used to create interactive web applications. In this project, streamlit was used to create a web application "apppp.py".

```
1   import streamlit as st
2   from predict_page import show_predict_page
3   from explore_page import show_explore_page
4
5   page = st.sidebar.selectbox("Explore or Predict", ("Explore", "Predict"))
6
7   if page == "Predict":
8       show_predict_page()
9   else:
10      show_explore_page()
```

RESULTS

INSIGHTS FROM EDA

- Clients without internet have a very low churn rate
- Customers with fiber are more probable to churn than those with DSL connection
- Customers with the first 4 additionals (security to tech support) are more unlikely to churn
- Streaming service is not predictive for churn
- Customers with paperless billing are more probable to churn
- The payment method, Electronic check has a very high churn rate
- Short term contracts have higher churn rates
- Longer contracts are more affected by higher monthly charges (for churn rate).
- Non senior citizens are churners

MODEL PERFORMANCE

The **Random Forest Classifier** Model performed best among other models, achieving an accuracy of 95.66%.

In [53]:
```
print(classification_report(
      yr_test,
      rfc_model.predict(xr_test)))
```

```
              precision    recall  f1-score   support

           0       0.96      0.95      0.95       519
           1       0.96      0.96      0.96       656

    accuracy                           0.96      1175
   macro avg       0.96      0.96      0.96      1175
weighted avg       0.96      0.96      0.96      1175
```

In [54]:
```
print(accuracy_score(yr_test, y_pred_rfc))
```

```
0.9565957446808511
```

CONCLUSION

This project successfully developed a predictive web application model for customer churn using historical data. The model's performance metrics indicate its effectiveness in identifying potential churners.