HOW TO IMPLEMENT CUSTOMER CHURN PREDICTION

TEAM O: GAYATRI SHANKAR SHINDE

Introduction: When it comes to useful business applications of machine learning, it doesn't get much better than customer churn prediction. It's a problem where you usually have a lot of high-quality, fresh data to work with, it's relatively straightforward, and solving it can be a great way to increase profits. The churn rate is a critical metric of customer satisfaction. Low churn rates mean happy customers; high churn rates mean customers are leaving you. A small rate of monthly/quarterly churn compounds over time. 1% monthly churn quickly translates to almost 12% yearly churn. Thanks to ML, forecasting customer churn with the help of machine learning is possible. Machine learning and data analysis are powerful ways to identify and predict churn. During churn prediction, you're also:

- Identifying at-risk customers,
- Identifying customer pain points,
- Identifying strategy/methods to lower churn and increase customer retention.

1. **Understanding the objective/Defining problem and goal**: In the telecommunications sector encompassing wireless and cable services, satellite television, and internet provision, churn rate stands as a pivotal metric. It serves as a barometer for business quality, indicating customer satisfaction levels and facilitating comparisons with industry rivals to ascertain an acceptable churn threshold.

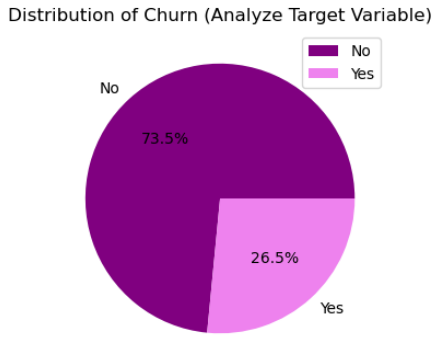
2. **Establishing data source:** The sample data tracks a fictional but classic and well-known telecommunications company, Telco. It's customer churn data sourced by the IBM Developer Platform. It includes a target label indicating whether or not the customer left within the one month, and other dependent features that cover demographics, services that each customer has signed up for, and customer account information. It has data for 7043 clients, with 20 features.

   There's 1 prediction feature:
   Churn: Whether the customer churned or not (Yes or No)
   These features can also be subdivided into:
   - Demographic customer information:
     gender , SeniorCitizen , Partner , Dependents
   - Services that each customer has signed up for:
     PhoneService , MultipleLines , InternetService , OnlineSecurity , OnlineBackup , DeviceProtection , TechSupport , StreamingTV , StreamingMovies,
   - Customer account information:
     tenure , Contract , PaperlessBilling , PaymentMethod , MonthlyCharges , TotalCharges

3. **Exploratory data analysis**:

   We're trying to predict users that left the company in the previous month. It's a binary classification problem with an unbalanced target.

   Churn: No – 73.5%

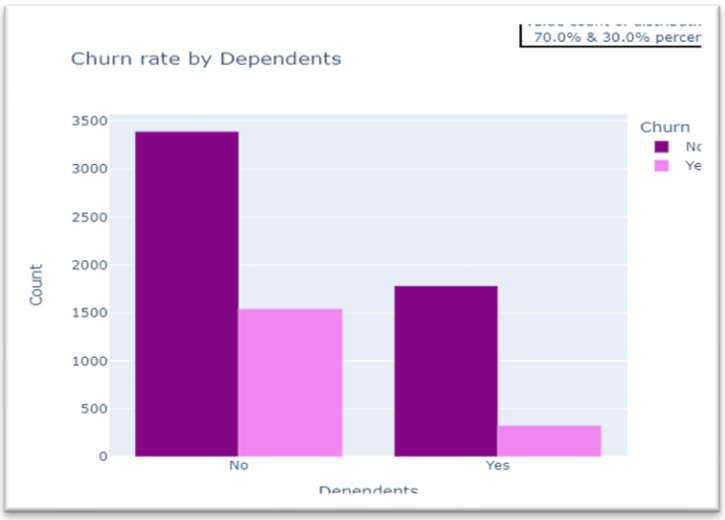


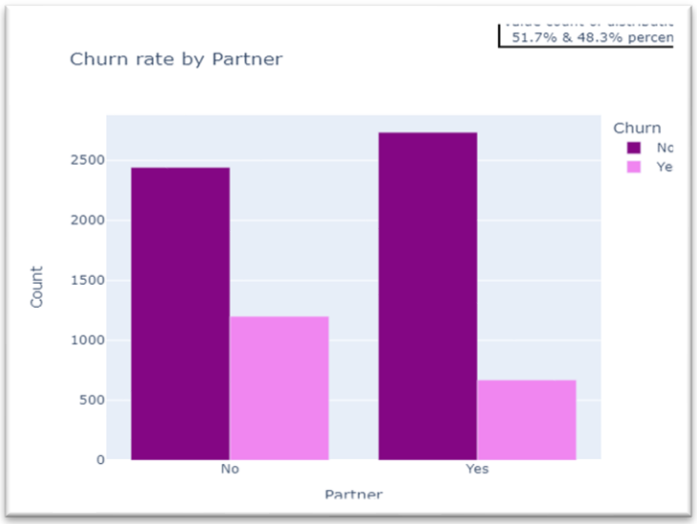
a) <u>Exploring Categorical Variables.</u>

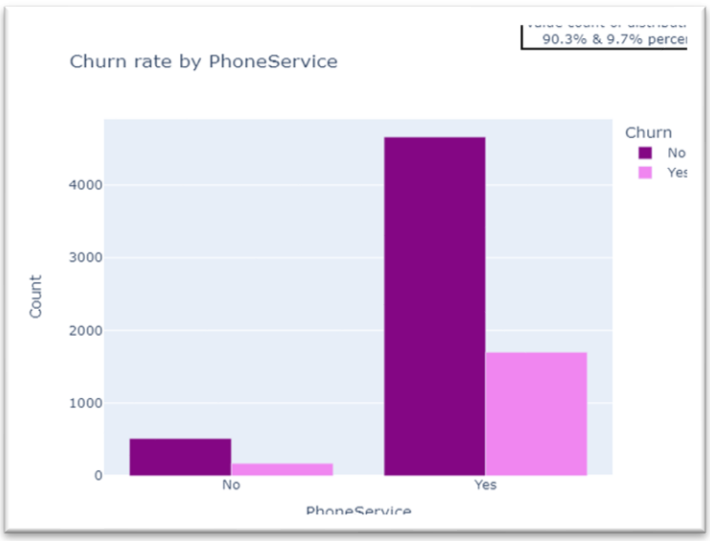
- Demographic features: Analysis reveals gender and partner distributions are nearly equal, with a slightly higher churn rate among females. Younger, unmarried customers without dependents exhibit a higher propensity to churn, emphasizing the significance of targeting this demographic segment.

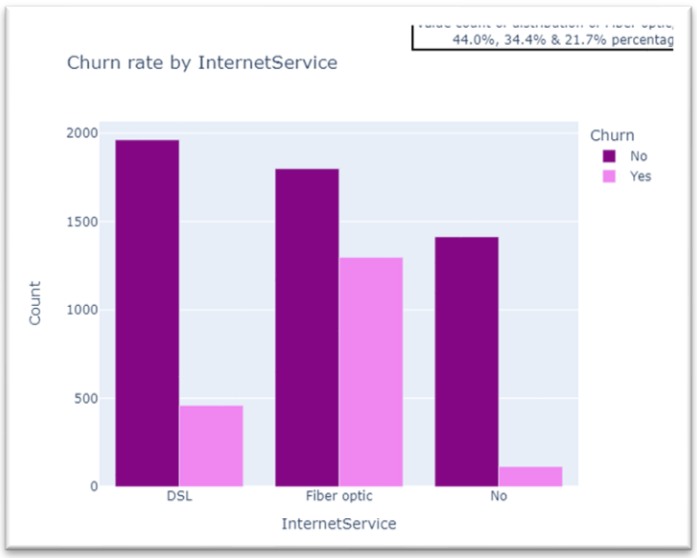
- Service subscriptions: Variations in service types are notable, with phone service being predominant among customers and correlating with a higher churn rate. Fiber optic internet subscribers demonstrate increased churn likelihood, possibly attributed to factors such as pricing disparities and service quality.

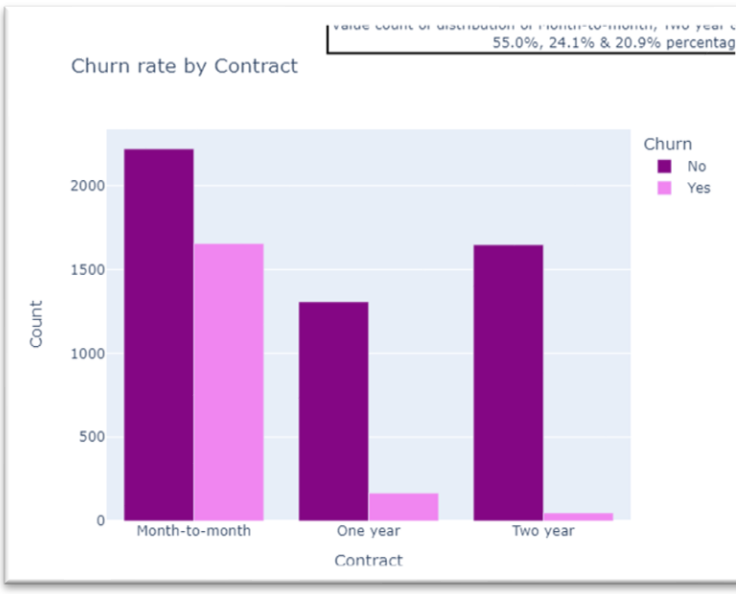
- Payment trends: Customers with shorter contract durations exhibit elevated churn rates, indicating the challenges in early termination. Paperless billing adoption aligns with higher churn rates, while customers paying via electronic checks are more prone to churn, underscoring the importance of fostering long-term customer relationships and diverse payment options

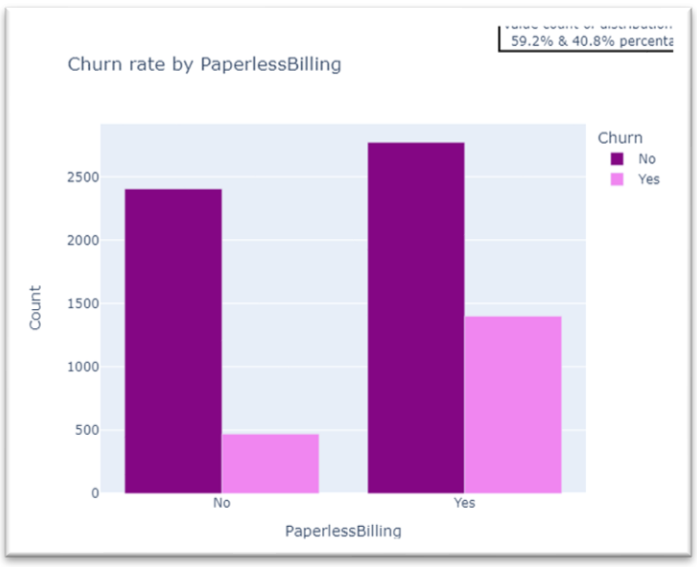






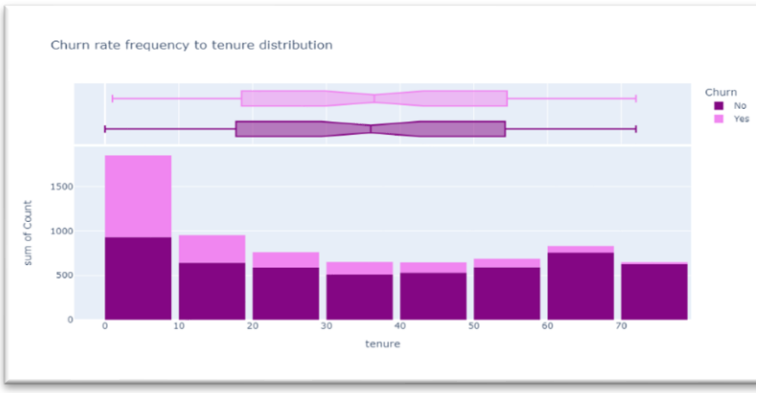


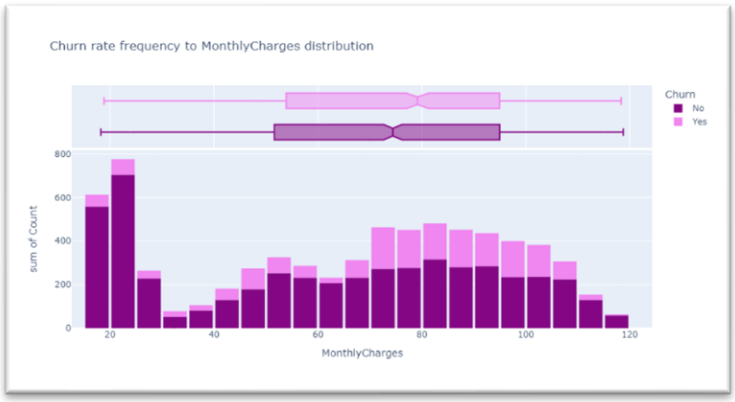




b) <u>Exploring Categorical Variables.</u>

- Customer account information: Insight into customer account information reveals a right-skewed tenure histogram, indicating a majority of customers have relatively short tenure, with the highest churn rate observed within the initial months. Notably, 75% of churn occurs within the first 30 months, underscoring the importance of early retention efforts. Furthermore, a correlation between higher monthly charges and increased churn rate suggests the potential efficacy of discounts and promotions in enhancing customer retention.





Based on binning, the low tenure and high monthly charge bins have higher churn rates, as supported by the previous analysis. At the same time, the low Total charge bin has a higher churn rate.

4.Data preprocessing

a) Dropping 'customerID' column: This action removes the 'customerID' column from the DataFrame, likely because it is considered irrelevant for the subsequent analysis.

b) Encoding target feature ('Churn'): The 'Churn' column is converted into a binary format, where 'Yes' is represented as 1 and 'No' as 0, facilitating binary classification tasks.

c) Encoding 'gender' category: The 'gender' column is transformed into a numeric format, likely to enable mathematical operations or compatibility with certain algorithms.

d) Encoding other binary categories: Similar to the 'gender' column, other binary categorical features such as 'SeniorCitizen', 'Partner', 'Dependents', 'PhoneService', and 'PaperlessBilling' are converted into numeric format, likely for consistency and ease of analysis.

e) Encoding other categorical features with more than two categories: Remaining categorical features with more than two categories are transformed using one-hot encoding, creating new binary columns for each category while dropping one column to prevent multicollinearity issues

Correlation measures the linear relationship between two variables. Features with high correlation are more linearly dependent and have almost the same effect on the dependent variable. So, when two features have a high correlation, we can drop one of them. In our case, we can drop highly correlated features like MultipleLines, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, & StreamingMovies. Churn prediction is a binary classification problem, as customers either churn or are retained in a given period.

Two questions need answering to guide model building:
Which features make customers churn or retain?
What are the most important features to train a model with high performance?

Using the generalized linear model (GLM) to gain some statistics of the respective features with the target. For the first question, I looked at the (P>|z|) column. If the absolute p-value is smaller than 0.05, it means that the feature affects Churn in a statistically significant way.
Examples are:
SeniorCitizen, Tenure, Contract
PaperlessBillings etc.

The second question about feature importances can be answered by looking at the exponential coefficient values. The exponential coefficient estimates the expected change in churn through a given feature by a change of one unit.

Generalized Linear Model Regression Results

Dep. Variable:	Churn	No. Observations:	7043
Model:	GLM	Df Residuals:	7019
Model Family:	Binomial	Df Model:	23
Link Function:	Logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-2914.7
Date:	Tue, 14 May 2024	Deviance:	5829.3
Time:	15:52:48	Pearson chi2:	8.04e+03
No. Iterations:	7	Pseudo R-squ. (CS):	0.2807
Covariance Type:	nonrobust		

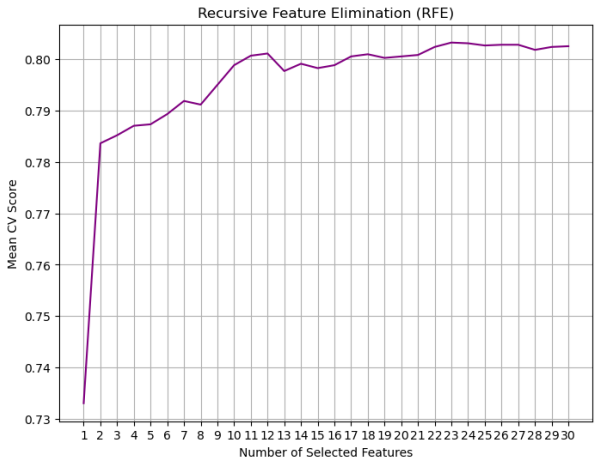
	coef	std err	z	P>|z|
Intercept	0.8274	0.748	1.106	0.269
MultipleLines_No_phone_service[T.True]	0.3238	0.106	3.061	0.002
MultipleLines_Yes[T.True]	0.4469	0.177	2.524	0.012
InternetService_Fiber_optic[T.True]	1.7530	0.798	2.198	0.028
InternetService_No[T.True]	-0.2559	0.115	-2.220	0.026
OnlineSecurity_No_internet_service[T.True]	-0.2559	0.115	-2.220	0.026
OnlineSecurity_Yes[T.True]	-0.2055	0.179	-1.150	0.250
OnlineBackup_No_internet_service[T.True]	-0.2559	0.115	-2.220	0.026
OnlineBackup_Yes[T.True]	0.0258	0.175	0.147	0.883
DeviceProtection_No_internet_service[T.True]	-0.2559	0.115	-2.220	0.026
DeviceProtection_Yes[T.True]	0.1477	0.176	0.838	0.402
TechSupport_No_internet_service[T.True]	-0.2559	0.115	-2.220	0.026
TechSupport_Yes[T.True]	-0.1789	0.180	-0.991	0.322
StreamingTV_No_internet_service[T.True]	-0.2559	0.115	-2.220	0.026
StreamingTV_Yes[T.True]	0.5912	0.326	1.813	0.070
StreamingMovies_No_internet_service[T.True]	-0.2559	0.115	-2.220	0.026
StreamingMovies_Yes[T.True]	0.6038	0.326	1.850	0.064
Contract_One_year[T.True]	-0.6671	0.107	-6.208	0.000
Contract_Two_year[T.True]	-1.3896	0.176	-7.904	0.000
PaymentMethod_Credit_card__automatic_[T.True]	-0.0865	0.114	-0.758	0.448
PaymentMethod_Electronic_check[T.True]	0.3057	0.094	3.236	0.001
PaymentMethod_Mailed_check[T.True]	-0.0567	0.115	-0.493	0.622
gender	-0.0219	0.065	-0.338	0.736
SeniorCitizen	0.2151	0.085	2.545	0.011
Partner	-0.0027	0.078	-0.035	0.972
Dependents	-0.1538	0.090	-1.714	0.087
tenure	-0.0594	0.006	-9.649	0.000
PhoneService	0.5036	0.692	0.728	0.467
PaperlessBilling	0.3418	0.074	4.590	0.000
MonthlyCharges	-0.0404	0.032	-1.272	0.203
TotalCharges	0.0003	7.01e-05	4.543	0.000

This fig below is a output the odd ratios. Values more than 1 indicate increased churn. Values less than 1 indicate that churn is happening less.

Intercept	2.287343
MultipleLines_No_phone_service[T.True]	1.382358
MultipleLines_Yes[T.True]	1.563475
InternetService_Fiber_optic[T.True]	5.771657
InternetService_No[T.True]	0.774257
OnlineSecurity_No_internet_service[T.True]	0.774257
OnlineSecurity_Yes[T.True]	0.814269
OnlineBackup_No_internet_service[T.True]	0.774257
OnlineBackup_Yes[T.True]	1.026127
DeviceProtection_No_internet_service[T.True]	0.774257
DeviceProtection_Yes[T.True]	1.159152
TechSupport_No_internet_service[T.True]	0.774257
TechSupport_Yes[T.True]	0.836193
StreamingTV_No_internet_service[T.True]	0.774257
StreamingTV_Yes[T.True]	1.806134
StreamingMovies_No_internet_service[T.True]	0.774257
StreamingMovies_Yes[T.True]	1.829067
Contract_One_year[T.True]	0.513185
Contract_Two_year[T.True]	0.249179
PaymentMethod_Credit_card__automatic_[T.True]	0.917142
PaymentMethod_Electronic_check[T.True]	1.357617
PaymentMethod_Mailed_check[T.True]	0.944913
gender	0.978355
SeniorCitizen	1.239957
Partner	0.997312
Dependents	0.857471
tenure	0.942322
PhoneService	1.654668
PaperlessBilling	1.407543
MonthlyCharges	0.960432
TotalCharges	1.000318
dtype: float64	

To ensure each feature contributes proportionally to the final distance, it was imperative to normalize their ranges through feature scaling. This process guaranteed equitable representation across all features in distance metrics hence used MinMax Scaler.

5.Feature selection



The optimal number of features: 23
Feature selection was conducted using Recursive Feature Elimination with Cross-Validation (RFECV) to improve model training efficiency, reduce complexity, enhance interpretability, and potentially boost accuracy by identifying optimal subsets of features. Logistic regression was utilized as the estimator, and Stratified K-Fold cross-validation with 10 folds was employed to iteratively select the most relevant features based on accuracy scoring. In the feature selection process, 7 features were removed from the original dataset. These features are: 'gender', 'Partner', 'DeviceProtection_No_internet_service', 'DeviceProtection_Yes', 'OnlineBackup_Yes', 'PaymentMethod_Credit_card__automatic_', & 'PaymentMethod_Mailed_check'.

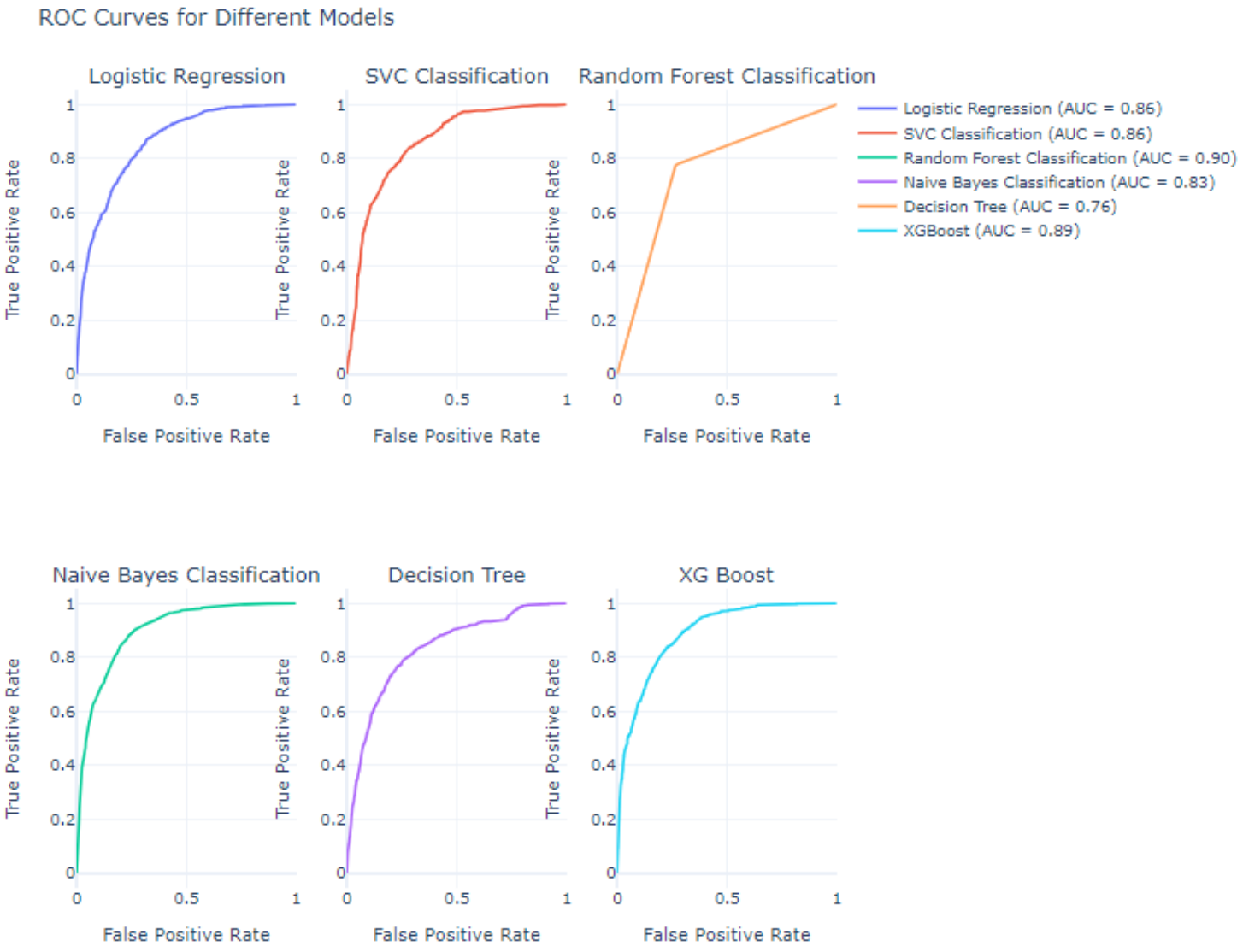
6.SMOTE, Model Performance and Selection

The selected 23 feature were stored in the new dataframe and tested with 6 models .In the context of predicting customer churn, where the occurrence of churn events is typically rare compared to non-churn instances, SMOTE plays a crucial role in rebalancing the dataset. By generating synthetic samples for the minority class (churners), SMOTE addresses the imbalance, ensuring that the machine learning model learns from both churn and non-churn instances effectively.In practical terms, SMOTE created synthetic churn instances by synthesizing new data points based on the existing minority class samples. These new samples are strategically generated along the decision boundary between churn and non-churn instances, allowing the model to better understand the underlying patterns associated with churn behavior.By employing SMOTE, we enhanced the performance of churn prediction model by providing it with a more balanced and representative

dataset. This, in turn, enables the model to make more accurate predictions and identify potential churners with greater reliability, ultimately helping businesses proactively retain their customers and minimize revenue loss.

Initially, a baseline model utilizing Logistic Regression algorithm is established for customer churn prediction using the corrected features and resampled dataset. Subsequently, various other machine learning models including Support Vector Classifier (SVC), Random Forest Classifier, Decision-tree classifier, and Naive-Bayes Classifier are employed to predict churn. Additionally, XGBoost, a powerful gradient boosting algorithm, is utilized to enhance churn prediction accuracy and model performance. From the selected performance metrics, the Random Forest classification algorithm has the highest scores across all chosen metrics

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.7723	0.7526	0.8058	0.7721
SVC Classification	0.7739	0.7383	0.8429	0.7729
Random Forest Classification	**0.8206**	**0.8017**	**0.8481**	**0.8205**
Decision Tree Classification	0.7536	0.7426	0.7701	0.7536
Naive Bayes Classification	0.7169	0.6588	0.8903	0.7085
XGBoost Classification	0.8045	0.7839	0.8364	0.8043



7. HperParameter Tuning.

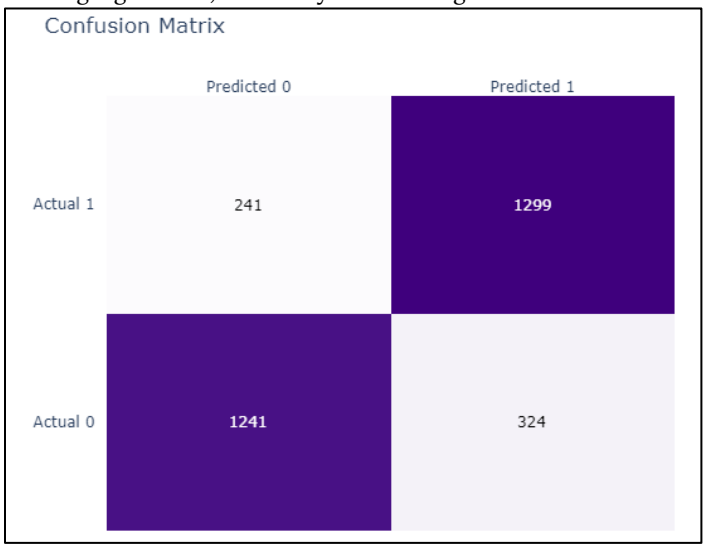
- To optimize the performance of the Random Forest classifier, we utilized hyperparameter tuning employing the Random Search technique. This method aims to find the best combination of hyperparameters for the model by systematically exploring a defined search space.
- Model Selection: The Random Forest classifier was selected due to its effectiveness in handling complex datasets and potential for improved performance with optimized hyperparameters.
- Evaluation Strategy: Repeated Stratified K-Fold cross-validation with 10 splits and 3 repeats was employed to ensure robust evaluation of model performance, accounting for variability in the data.

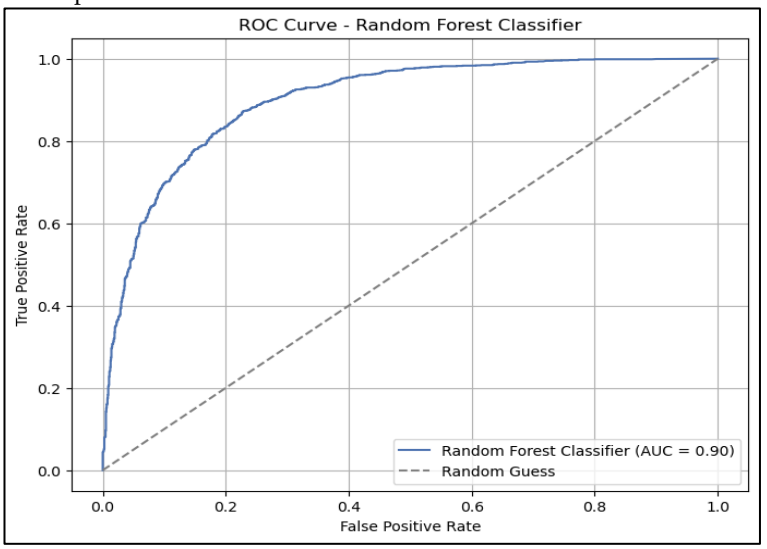
- Search Space Definition: A search space was defined to explore various hyperparameters, including 'n_estimators', 'max_features', 'max_depth', 'min_samples_split', 'min_samples_leaf', and 'bootstrap'. The search space encompasses a range of potential values for each hyperparameter.
- Random Search Execution: RandomizedSearchCV was utilized to execute the hyperparameter search. This involved randomly sampling a specified number of hyperparameter combinations from the defined search space and evaluating each combination using cross-validation.
- Model Improvement: The Random Forest model was improved based on the results of the hyperparameter search. The best combination of hyperparameters achieved a Best Score of 0.8298, indicating improved model performance.
- Best Hyperparameters: The optimal hyperparameters identified through the search were {'bootstrap': False, 'max_depth': 20, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 229}

8. Results

Metric	Original RF	Tuned RF
Accuracy	0.8206	0.8280
Precision	0.8017	0.8034
Recall	0.8481	0.8531
F1 Score	0.8205	0.8280

Based on the results, both the original and tuned Logistic Regression models demonstrate comparable performance across all metrics. The slight improvements observed in the tuned model suggest that hyperparameter tuning effectively fine-tuned the model, albeit marginally. While the differences in performance metrics between the original and tuned models are relatively small, even minor enhancements can be significant in certain applications, especially when dealing with critical tasks such as churn prediction in the telecommunications industry. Therefore, the effort invested in hyperparameter tuning can be considered worthwhile, as it ensures that the model is optimized to achieve the best possible performance.
Overall, these results underscore the importance of fine-tuning model parameters to extract the maximum predictive power from machine learning algorithms, ultimately contributing to more accurate and reliable predictions





9. Deployment

The best-performing model, the tuned Logistic Regression classification model, has been saved to disk using the joblib library. The model is stored under the filename 'model.sav' for future use and deployment in production environments.
The application to be deployed will function via operational use cases:
Online prediction: This use case generates predictions on a one-by-one basis for each data point (in the context of this project, a customer).
Batch prediction: This use is for generating predictions for a set of observations instantaneously.
To deploy the saved model in Streamlit, I structured my project with two main files: 'app.py' and 'preprocessing.py'. In 'app.py', I imported the necessary libraries, loaded the trained model using joblib, and defined the Streamlit app interface. I then called the preprocessing functions from 'preprocessing.py' to prepare incoming data for prediction. This separation of concerns ensures a clean and organized codebase, with 'app.py' focused on the app interface and 'preprocessing.py' handling data preprocessing tasks. With the Streamlit app set up, users can interactively input data, which undergoes preprocessing before being fed into the deployed model for predictions.

9. Conclusion

Churn rate is an important indicator for subscription-based companies. Identifying customers who aren't happy can help managers identify product or pricing plan weak points, operation issues, as well as customer preferences and expectations. When you know all that, it's easier to introduce proactive ways of reducing churn.