Data interpretation and findings for Telco

Brief Structure of the dataset

The dataset is obtained from Kaggle and has a Usability value of 8.82. It contains 7043 rows and 21 columns. Each row represents a customer, represented by its unique customer ID and the target column is "Churn"- which tells if the customer has left the telco company's service in the past month or not. Value "1" means left and "0" means stayed. I have used Kaggle's API to fetch the dataset. Alternatively, you can download the dataset from "https://www.kaggle.com/datasets/blastchar/telco-customer-churn"

The downloaded dataset's columns include the following parameters:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
 15  Contract          7043 non-null   object
 16  PaperlessBilling  7043 non-null   object
 17  PaymentMethod     7043 non-null   object
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object
 20  Churn             7043 non-null   object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

Cleaning the dataset

The dataset has been cleaned in two stages:

1. Stage 1: - standard cleaning procedures have been followed which include:
   a) Dropping duplicates, if any
   b) Checking for any empty rows in the columns (found 11 in 'TotalCharges')
   c) Analyzing the rows where empty rows were found.
   d) Checking and correcting the datatypes of the columns.
   e) Converting the options ('Yes'- 1, 'No'- 0 , 'No phone service'-0, 'No internet service'-0) to do modelling on the dataset efficiently in a later stage.
   f) Finding the skewness (0.96, right skewness) of the dataset so as to decide whether to fill the median or mean in the empty rows. In our case, we inputted median.
   g) Checked again for any missing values. Saved the result in "cleaned_stage1.csv"

2. In stage 2: identification of outliers took place by plotting box-plots.

```
''' analysis for tenure:
    q1=9, q3=55, so IQR= q3-q1=46, median =29.
    lower fence= Q1-1.5XIQR=-60 which is effectively 0
    upper fence= q3+1.5XIQR= 55+1.5X46=124
    so we have no values outside this range so NO OUTLIER FOR TENURE '''
'''ANALYSIS FOR MONTHLY CHARGES
    q1=35.5, q3=89.85,so IQR=q3-q1=89.85-39.5=50.35, median=70.35
    lower fence= q1- 1.5XIQR= 35.5-1.5X50.35=-40
    upper fence= q3+1.5XIQR=89.85+1.5X50.35=165.375
    so we have no values outside this range so NO OUTLIER for MONTHLY CHARGES'''
''' ANALYSIS FOR TOTAL CHARGES
    q1=402, q3=3787.9, IQR=q3-q1=3385.9, median=1397.4
    lower fence= q1-1.5xIQR= 402-1.5x3385.9=-4676.85
    upper fence= q3+1.5xIQR= 3787.9+1.5x3385.9=8866.75'''
```

No outliers were found in the dataset.

Exploratory Data Analysis

Steps done in EDA are as follows :-

1. Line chart for tenure bins
2. Interactive Histogram (need to hover to see the values) for 6 month and 3 month intervals.

	tenure_bins	count
0	0-6	1481
1	7-12	705
2	13-18	548
3	19-24	476
4	25-30	431
5	31-36	401
6	37-42	379
7	43-48	383
8	49-54	420
9	55-60	412
10	61-66	463
11	67-72	944

	tenure_bins	count
0	0-2	862
1	3-5	509
2	6-8	364
3	9-11	334
4	12-14	302
5	15-17	266
6	18-20	241
7	21-23	238
8	24-26	252
9	27-29	201
10	30-32	206
11	33-35	217
12	36-38	174
13	39-41	190
14	42-44	181
15	45-47	203
16	48-50	198
17	51-53	218
18	54-56	212
19	57-59	192
20	60-62	222
21	63-65	228
22	66-68	287
23	69-72	746

This data will help in finding which customers to target for retention strategies.

3. For 3 month intervals, grouped and calculated data which shows mean and median monthly charges paid by the customers to telco, along with the minium and maximum paid by the customers.

	tenure_bins_3_months	mean_monthly_charges	median_monthly_charges	min_monthly_charges	max_monthly_charges
0	0-2	52.225638	50.450	18.75	104.40
1	3-5	58.594597	64.350	18.80	107.95
2	6-8	57.909615	61.150	18.95	109.90
3	9-11	60.052844	66.075	18.85	111.40
4	12-14	59.470695	61.725	18.80	112.95
5	15-17	61.854511	69.300	18.80	112.95
6	18-20	59.456224	61.500	18.80	108.20
7	21-23	62.903361	68.700	19.05	111.20
8	24-26	61.848810	66.175	18.70	108.90
9	27-29	63.062687	68.850	18.25	110.85
10	30-32	70.688350	79.375	18.95	110.45
11	33-35	65.570046	70.300	19.15	116.25
12	36-38	65.313793	73.150	18.55	110.70
13	39-41	65.852368	70.225	19.10	114.50
14	42-44	67.132597	73.850	19.05	115.05
15	45-47	67.303202	74.800	18.85	115.65
16	48-50	69.679545	75.350	19.00	117.45
17	51-53	67.492202	75.900	18.70	111.80
18	54-56	72.347406	78.600	18.95	116.50
19	57-59	69.454687	74.475	18.40	113.75
20	60-62	73.182432	81.025	19.10	118.60
21	63-65	76.741886	85.375	19.15	115.10
22	66-68	73.834495	80.450	19.25	118.60
23	69-72	77.163874	85.450	19.10	118.75

This data will help in later stage for business report and marketing strategies.

4. Next, I found the churn percentage of every 3 month interval to find relation between the charges paid by the customer and their decision to leave the service, i.e. Churn. We found that the churn rate was highest amongst the early customers, which had stayed for 0-6 months with the company. So focusing on the early customers which have joined the company in less than a years time will be profitable.

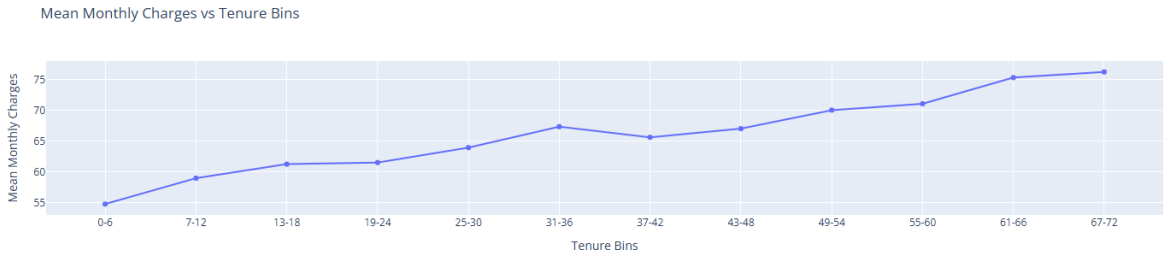
	tenure_bins_3_months	mean_monthly_charges	median_monthly_charges	min_monthly_charges	max_monthly_charges	customer_count	churn_percentage
0	0-2	52.225638	50.450	18.75	104.40	862	58.352668
1	3-5	58.594597	64.350	18.80	107.95	509	47.347741
2	6-8	57.909615	61.150	18.95	109.90	364	36.538462
3	9-11	60.052844	66.075	18.85	111.40	334	36.526946
4	12-14	59.470695	61.725	18.80	112.95	302	33.112583
5	15-17	61.854511	69.300	18.80	112.95	266	34.210526
6	18-20	59.456224	61.500	18.80	108.20	241	25.311203
7	21-23	62.903361	68.700	19.05	111.20	238	23.949580
8	24-26	61.848810	66.175	18.70	108.90	252	24.206349
9	27-29	63.062687	68.850	18.25	110.85	201	19.900498
10	30-32	70.688350	79.375	18.95	110.45	206	24.757282
11	33-35	65.570046	70.300	19.15	116.25	217	18.894009
12	36-38	65.313793	73.150	18.55	110.70	174	21.839080
13	39-41	65.852368	70.225	19.10	114.50	190	21.578947
14	42-44	67.132597	73.850	19.05	115.05	181	19.337017
15	45-47	67.303202	74.800	18.85	115.65	203	15.763547
16	48-50	69.679545	75.350	19.00	117.45	198	17.171717
17	51-53	67.492202	75.900	18.70	111.80	218	13.761468
18	54-56	72.347406	78.600	18.95	116.50	212	15.094340
19	57-59	69.454687	74.475	18.40	113.75	192	14.062500
20	60-62	73.182432	81.025	19.10	118.60	222	8.558559
21	63-65	76.741886	85.375	19.15	115.10	228	7.456140
22	66-68	73.834495	80.450	19.25	118.60	287	11.149826
23	69-72	77.163874	85.450	19.10	118.75	746	4.155496

Similarly made for 6 months

	tenure_bins	count	mean_monthly_charges	median_monthly_charges	min_monthly_charges	max_monthly_charges	customer_count	churn_percentage
0	0-6	1481	54.738656	54.700	18.75	109.90	1481	52.937205
1	7-12	705	58.952908	64.950	18.85	112.95	705	35.886525
2	13-18	548	61.236131	68.475	18.80	112.95	548	32.299270
3	19-24	476	61.496744	65.150	18.80	111.20	476	24.579832
4	25-30	431	63.929698	69.750	18.25	110.85	431	21.809745
5	31-36	401	67.344389	75.200	18.55	116.25	401	21.446384
6	37-42	379	65.610158	70.600	19.05	114.50	379	21.899736
7	43-48	383	67.018930	73.850	18.85	117.45	383	16.187990
8	49-54	420	70.039286	79.175	18.70	115.60	420	16.190476
9	55-60	412	71.072209	76.800	18.40	116.60	412	12.621359
10	61-66	463	75.335529	84.500	19.10	118.60	463	9.287257
11	67-72	944	76.255403	84.800	19.10	118.75	944	5.296610

5. Finding correlation between the different columns to tenure.
   a) Between monthly charges and tenure

```
Correlation between MonthlyCharges and Tenure:
               MonthlyCharges  tenure
MonthlyCharges          1.0000  0.2479
tenure                  0.2479  1.0000
```



Check for anomolies - Anomalies with 0 tenure but non-zero TotalCharges.. Found out that there was just a system issue of telco that they had not updated the tenure on their database. These customers had `TotalCharges` as NaN, which was imputed with the median value during data cleaning. Since all these customers have `Churn = 0`, they are likely valid, active customers who were billed upfront before completing a full month of service.

6. Generated bar charts, and heat maps, line graphs for "Churn Rate vs. Time Active" and "Monthly Charges vs. Time Active "for 3-month intervals and 6 month intervals.
7. Churn analysis by factors present in the cleaned dataset.

gender	Male (3555)	Female(3488)
number	930	939
percentage	26.160338	26.920872

Senior Citizen	0	1	Churned percentage
Total customers	5901	1142	23.606168
churned	1393	476	41.681261

Contract	Month-to-month	1 year	2 years
Total customers	3875	1473	1695
churned	1655	166	48
Churned %	42.709677	11.269518	2.831858

Payment type	Bank transfer	Credit card	Electronic check	Mailed check

	(automatic)	(automatic)		
total_customers	1544	1522	2365	1612
churned_customers	258	232	1071	308
churn_rate	16.7%	15.24	45.28%	19.10%

\

Internet service	fiber	DSL	NO internet service
Total customers	3096	2421	1526
Churned percentage	41.892765	18.959108	7.404980
Churned customer count	1297	459	113

OnlineSecurity	0	1	
Total customers	5024	2019	
churned	1574	295	
Churned percenatge	31.329618	14.611194	

TechSupport	0	1
Total customers	4999	2044
Churned customers	1559	310
percentage	31.186237	15.166341

StreamingTV	0	1
total_customers	4336	2707
Churned customers	1055	814
percentage	24.331181	30.070188

StreamingMovies	0	1
Total customers	4311	2732
Churned customers	1051	818
percentage	24.379494	29.941435

8. Next I analyzed data between senior citizens (taking them whole and then later analyzing through gender) and non senior citizens (same approach) so see which category is more prone to leave so we have a targetted customer segment.

```
senior_citizen_male_analysis
```

]:

	category	Value	Count	churned_customers	Churn Rate (%)
0	PaymentMethod	Bank transfer (automatic)	121	27	22.314050
1	PaymentMethod	Credit card (automatic)	111	27	24.324324
2	PaymentMethod	Electronic check	298	159	53.355705
3	PaymentMethod	Mailed check	44	23	52.272727
4	Contract	Month-to-month	408	220	53.921569
5	Contract	One year	89	13	14.606742
6	Contract	Two year	77	3	3.896104
7	InternetService	DSL	136	40	29.411765
8	InternetService	Fiber optic	407	194	47.665848
9	InternetService	No	31	2	6.451613
10	OnlineSecurity	0	441	208	47.165533
11	OnlineSecurity	1	133	28	21.052632
12	TechSupport	0	455	217	47.692308
13	TechSupport	1	119	19	15.966387
14	StreamingTV	0	294	115	39.115646
15	StreamingTV	1	280	121	43.214286
16	StreamingMovies	0	271	103	38.007380
17	StreamingMovies	1	303	133	43.894389

```
senior_citizen_female_analysis
```

	category	Value	Count	churned_customers	Churn Rate (%)
0	PaymentMethod	Bank transfer (automatic)	112	26	23.214286
1	PaymentMethod	Credit card (automatic)	110	35	31.818182
2	PaymentMethod	Electronic check	296	158	53.378378
3	PaymentMethod	Mailed check	50	21	42.000000
4	Contract	Month-to-month	399	221	55.388471
5	Contract	One year	101	16	15.841584
6	Contract	Two year	68	3	4.411765
7	InternetService	DSL	123	38	30.894309
8	InternetService	Fiber optic	424	199	46.933962
9	InternetService	No	21	3	14.285714
10	OnlineSecurity	0	419	204	48.687351
11	OnlineSecurity	1	149	36	24.161074
12	TechSupport	0	427	208	48.711944
13	TechSupport	1	141	32	22.695035
14	StreamingTV	0	276	132	47.826087
15	StreamingTV	1	292	108	36.986301
16	StreamingMovies	0	276	129	46.739130
17	StreamingMovies	1	292	111	38.013699

```
senior_citizens_whole
```

	category	Value	Count	churned_customers	Churn Rate (%)
0	PaymentMethod	Bank transfer (automatic)	233	53	22.746781
1	PaymentMethod	Credit card (automatic)	221	62	28.054299
2	PaymentMethod	Electronic check	594	317	53.367003
3	PaymentMethod	Mailed check	94	44	46.808511
4	Contract	Month-to-month	807	441	54.646840
5	Contract	One year	190	29	15.263158
6	Contract	Two year	145	6	4.137931
7	InternetService	DSL	259	78	30.115830
8	InternetService	Fiber optic	831	393	47.292419
9	InternetService	No	52	5	9.615385
10	OnlineSecurity	0	860	412	47.906977
11	OnlineSecurity	1	282	64	22.695035
12	TechSupport	0	882	425	48.185941
13	TechSupport	1	260	51	19.615385
14	StreamingTV	0	570	247	43.333333
15	StreamingTV	1	572	229	40.034965
16	StreamingMovies	0	547	232	42.413163
17	StreamingMovies	1	595	244	41.008403

For non_senior_citizens

```
non_senior_citizens_whole
```

]:

	category	Value	Count	churned_customers	Churn Rate (%)
0	PaymentMethod	Bank transfer (automatic)	1311	205	15.636918
1	PaymentMethod	Credit card (automatic)	1301	170	13.066872
2	PaymentMethod	Electronic check	1771	754	42.574816
3	PaymentMethod	Mailed check	1518	264	17.391304
4	Contract	Month-to-month	3068	1214	39.569752
5	Contract	One year	1283	137	10.678098
6	Contract	Two year	1550	42	2.709677
7	InternetService	DSL	2162	381	17.622572
8	InternetService	Fiber optic	2265	904	39.911700
9	InternetService	No	1474	108	7.327001
10	OnlineSecurity	0	4164	1162	27.905860
11	OnlineSecurity	1	1737	231	13.298791
12	TechSupport	0	4117	1134	27.544328
13	TechSupport	1	1784	259	14.517937
14	StreamingTV	0	3766	808	21.455125
15	StreamingTV	1	2135	585	27.400468
16	StreamingMovies	0	3764	819	21.758767
17	StreamingMovies	1	2137	574	26.860084

```
non_senior_citizen_female_analysis
```

	category	Value	Count	churned_customers	Churn Rate (%)
0	PaymentMethod	Bank transfer (automatic)	676	110	16.272189
1	PaymentMethod	Credit card (automatic)	642	96	14.953271
2	PaymentMethod	Electronic check	874	364	41.647597
3	PaymentMethod	Mailed check	728	129	17.719780
4	Contract	Month-to-month	1526	621	40.694626
5	Contract	One year	617	59	9.562399
6	Contract	Two year	777	19	2.445302
7	InternetService	DSL	1065	181	16.995305
8	InternetService	Fiber optic	1129	465	41.186891
9	InternetService	No	726	53	7.300275
10	OnlineSecurity	0	2042	573	28.060725
11	OnlineSecurity	1	878	126	14.350797
12	TechSupport	0	2034	566	27.826942
13	TechSupport	1	886	133	15.011287
14	StreamingTV	0	1857	405	21.809370
15	StreamingTV	1	1063	294	27.657573
16	StreamingMovies	0	1841	403	21.890277
17	StreamingMovies	1	1079	296	27.432808

9. Analysis on the data obtained from point 8. The results will be pointed out in the business report for this project.

Modelling

I have used supervised learning algorithms for this dataset as the dataset of a CLEAR TARGET VARIABLE- CHURN and we have binary data, which is not good for unsupervised learning models. Therefore, i have used logistic regression, decision tree classifier, random forest, gradient boosting, KNN and SVM algorithms to model the data and find which model can give me the best results.

Model Development and Evaluation Steps:

1. Data Preprocessing:
    a. Split the dataset into features (X) and target (y), encode categorical variables, and standardize numeric features.
2. Train-Test Split:
    a. The data is split into 80% for training and 20% for testing.
3. Initial Model Training:
    a. We trained a logistic regression model with a default threshold of 0.5 for predicting churn.
4. Model Evaluation:
    a. Evaluated performance using accuracy, classification report, and confusion matrix.
5. Cross-Validation:
    a. Used 5-fold cross-validation to assess the model's consistency and calculate the mean cross-validation accuracy.
6. Feature Importance:
    a. Identified key features influencing customer churn based on the model's coefficients.
7. Hyperparameter Tuning:
    a. Tuned the model using **GridSearchCV**, testing various regularization strengths (C) and penalties (l1 and l2).
8. ROC Curve:
    a. Plotted the ROC curve and calculated the AUC to measure the model's ability to distinguish between churners and non-churners.
9. Precision-Recall Curve:
    a. Plotted the precision-recall curve, calculated the best threshold, and selected the optimal decision boundary.
10. Custom Threshold:
    a. Applied the best threshold to adjust the decision-making process and improved model performance.
11. Final Evaluation:
    a. After retraining with the custom threshold, we re-evaluated performance with accuracy and cross-validation, ensuring the model's robustness.

Decision Tree modifier

1. Model Initialization:

a. Initialized a Decision Tree classifier with a fixed random state for reproducibility.

2. Model Training:
   a. Trained the Decision Tree model on the training data with default parameters with 80-20 sets.
3. Model Evaluation:
   a. Evaluated the model's performance on the training and test data using:
      i. Accuracy - `Training Accuracy: 0.9982250621228257, Testing Accuracy: 0.7246273953158269`
      ii. Classification Report

```
Classification Report (Test Data):
              precision    recall  f1-score   support

           0       0.81      0.81      0.81      1023
           1       0.50      0.51      0.50       386

    accuracy                           0.72      1409
   macro avg       0.66      0.66      0.66      1409
weighted avg       0.73      0.72      0.73      1409
```

      iii. Confusion Matrix : `[[824 199]`
      `[189 197]]`
   b. Identified overfitting due to the significant difference between training and testing accuracies.

- **Observation**: The model shows signs of **overfitting**, as indicated by a significant difference between **training accuracy** and **testing accuracy**. This suggests that the model is performing well on the training data but not generalizing well to unseen data (the test set). To combat this: pruning, cross-validation and regularization is done.

4. Pruning the Decision Tree:
   a. Applied pruning to the tree by setting:
      i. max_depth = 5
      ii. min_samples_split = 10
      iii. min_samples_leaf = 5

   Results: `Training Accuracy (Pruned Tree): 0.8020944266950657`

```
Testing Accuracy (Pruned Tree): 0.7771469127040455
```

```
Classification Report (Pruned Tree):
              precision    recall  f1-score   support

           0       0.81      0.91      0.86      1023
           1       0.64      0.43      0.51       386

    accuracy                           0.78      1409
   macro avg       0.72      0.67      0.68      1409
weighted avg       0.76      0.78      0.76      1409

Confusion matrix- [[930  93]
                   [221 165]]
```

b. Retrained the model with these hyperparameters and reassessed performance.

5. Feature Importance:
   a. Analyzed the top features contributing to churn predictions using the model's feature importances.
   b. Plotted a bar chart to visualize the importance of the top 10 features.

```
Top 10 Important Features:
                            Feature  Importance
3                            tenure    0.458658
16     InternetService_Fiber optic    0.344321
14                     TotalCharges    0.049238
17               InternetService_No    0.040273
19                Contract_Two year    0.024828
21  PaymentMethod_Electronic check    0.017986
18                Contract_One year    0.013288
13                   MonthlyCharges    0.012366
6                    OnlineSecurity    0.009501
10                      StreamingTV    0.007009
```

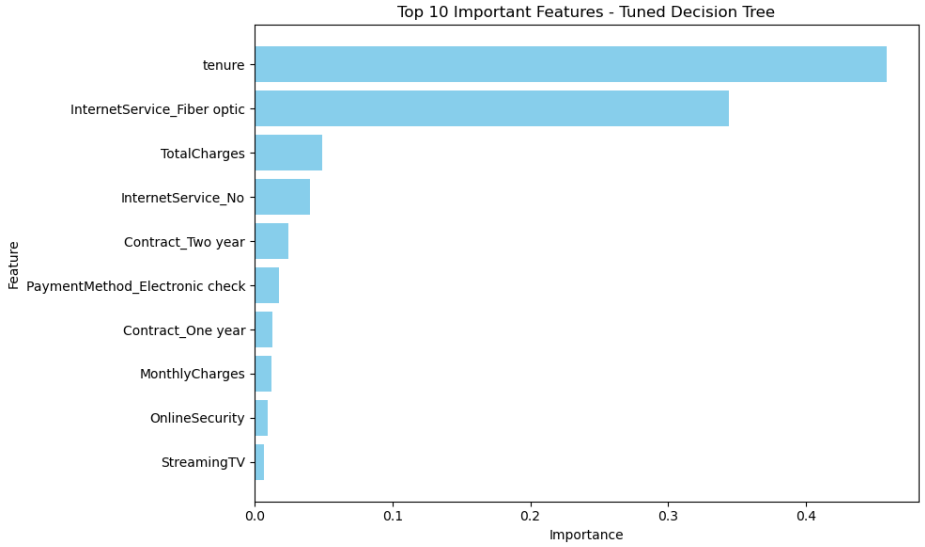
6. Partial Dependence Plot for Key Features. We created **Partial Dependence Plots (PDPs)** to visualize the relationship between the most important features and the predicted probability of churn
7. Cross-Validation of Pruned Decision Tree -

```
Cross-Validation Scores: [0.78704525 0.79414374 0.78615794 0.77905945 0.80195382]
Mean CV Accuracy: 0.7896720414940244
```

8. **Hyperparameter Tuning with GridSearchCV**:
9. Used **GridSearchCV** to find the optimal hyperparameters for the Decision Tree, focusing on:
   a. **max_depth**
   b. **min_samples_split**
   c. **min_samples_leaf**

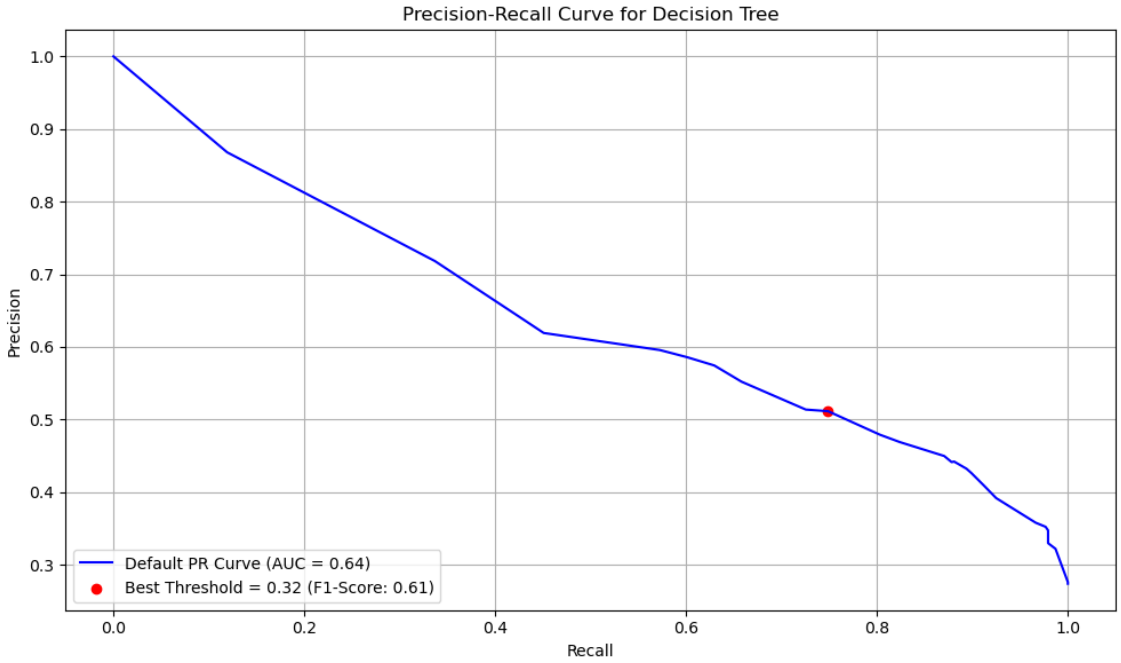
```
Best Parameters: {'max_depth': 3, 'min_samples_leaf': 1, 'min_samples_split': 2}
Best CV Accuracy: 0.7921573015645365

Classification Report (Best Decision Tree):
              precision    recall  f1-score   support

           0       0.80      0.92      0.86      1023
           1       0.66      0.40      0.49       386

    accuracy                           0.78      1409
   macro avg       0.73      0.66      0.68      1409
weighted avg       0.76      0.78      0.76      1409


Confusion Matrix (Best Decision Tree):
[[943  80]
 [233 153]]
```

10. Top 10 features of pruned decision tree.



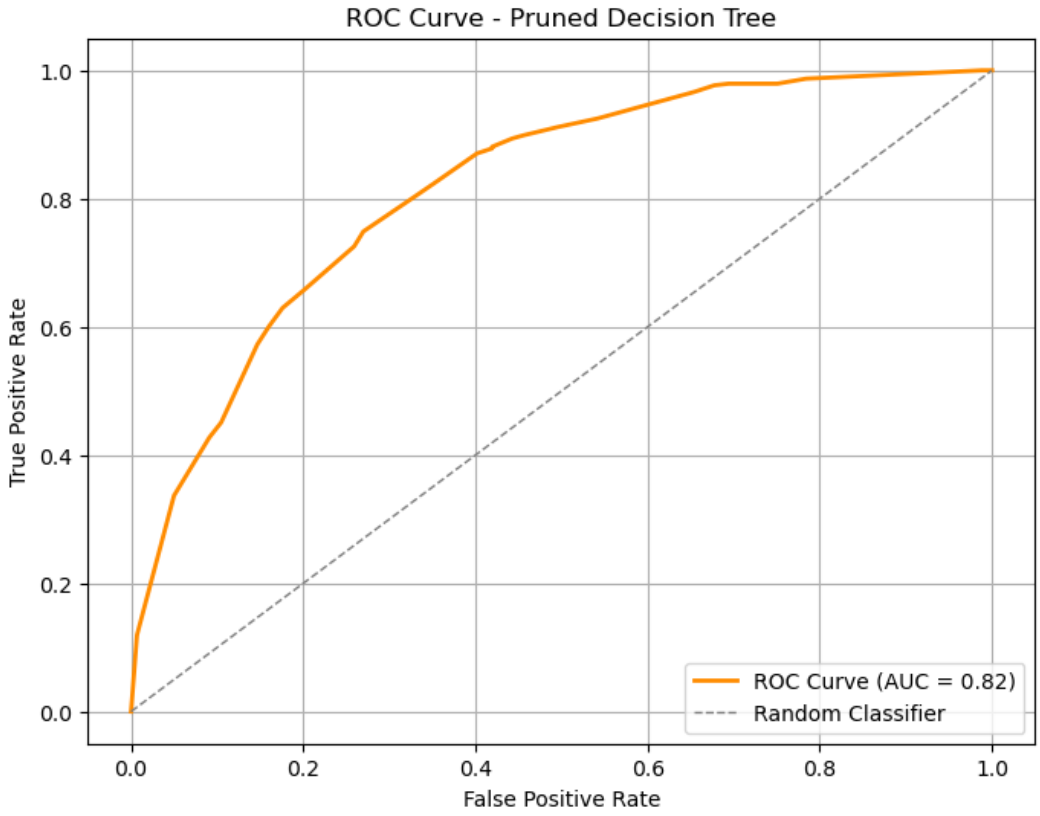
11. **Precision-Recall Curve and Best Threshold**:
    I) Plotted the **Precision-Recall Curve** to find the **best threshold** that maximizes the **F1 score**.
    II) Applied the **best threshold** to make final predictions and evaluated the model's performance.

{'Best Threshold': 0.31645569620253167,
 'Training Accuracy': 0.8020944266950657,
 'Testing Accuracy': 0.7352732434350603,
 'Classification Report': '   precision   recall f1-score   support\n\n   0   0.89   0.73   0.80   1023\n   1   0.51   0.75   0.61
386\n\n   accuracy   0.74   1409\n   macro avg   0.70   0.74   0.70   1409\nweighted avg   0.78   0.74   0.75   1409\n',
 'Confusion Matrix': array([[747, 276],
        [97, 289]], dtype=int64)}



12. ROC Curve and AUC:

- Plotted the **ROC curve** and calculated the **AUC** to assess the model's ability to distinguish between churners and non-churners.



8]: 0.8163407432168923

Random Forest Model

1. Model Initialization and Training : initialized and trained a **Random Forest classifier** using the default parameters.
2. Model Evaluation for Random Forest:
   Evaluated the Random Forest model's performance using:
   - **Accuracy** on both **training** and **testing** datasets.
   - **Classification report** for detailed metrics like precision, recall, and F1 score.
   - **Confusion matrix** to understand the breakdown of predictions (True Positives, True Negatives, False Positives, False Negatives).

```
Training Accuracy (RF): 0.9982250621228257
Testing Accuracy (RF): 0.7806955287437899

Classification Report (RF):
              precision    recall  f1-score   support

           0       0.82      0.90      0.86      1023
           1       0.63      0.47      0.54       386

    accuracy                           0.78      1409
   macro avg       0.73      0.68      0.70      1409
weighted avg       0.77      0.78      0.77      1409


Confusion Matrix (RF):
[[918 105]
 [204 182]]
```

3. Extracted **feature importance** from the trained **Random Forest** model and visualized the top contributing features.

```
Top 10 Important Features (Random Forest):
                          Feature  Importance
14                    TotalCharges    0.198466
13                  MonthlyCharges    0.175261
3                           tenure    0.173524
16     InternetService_Fiber optic    0.050116
21  PaymentMethod_Electronic check    0.038327
19               Contract_Two year    0.030816
15                     gender_Male    0.027507
12                PaperlessBilling    0.026722
6                   OnlineSecurity    0.024943
1                           Partner    0.024318
```

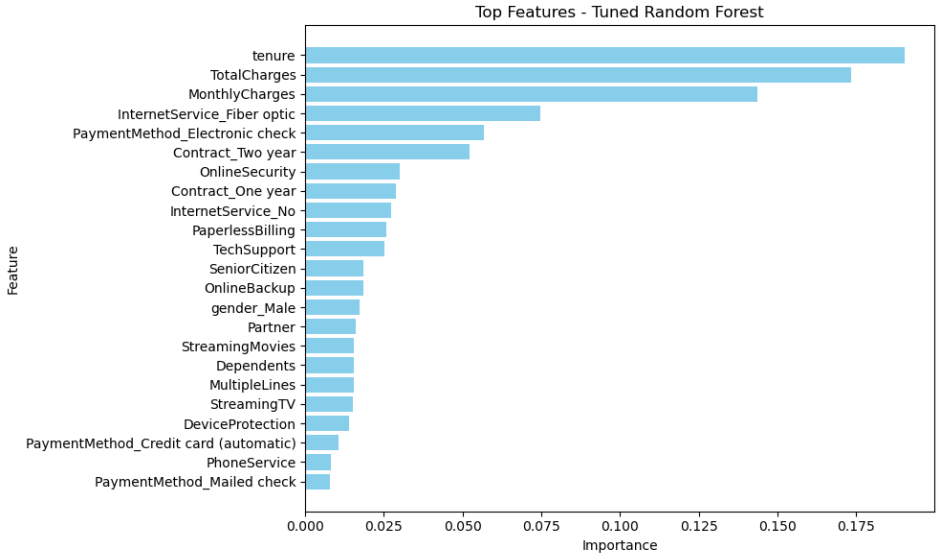
4. Cross-Validation for Random Forest - Performed **5-fold cross-validation** on the Random Forest model to evaluate its performance across multiple subsets of the training data. This ensures a more reliable estimate of the model's generalization ability.

```
Cross-Validation Scores (RF): [0.80567879 0.79680568 0.78527063 0.7826087  0.8277087 ]
Mean CV Accuracy (RF): 0.7996145002135536
```

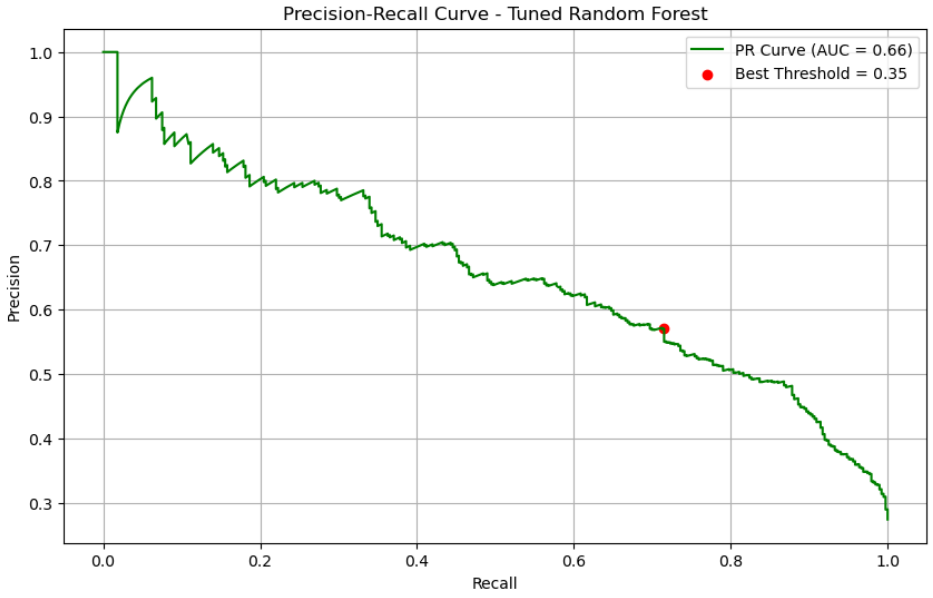
5. We performed **hyperparameter tuning** for the **Random Forest** model using **GridSearchCV** to find the optimal combination of hyperparameters that maximizes model performance. **GridSearchCV** helped identify the best hyperparameters for the **Random Forest** model, improving **cross-validation performance**.

```
Fitting 5 folds for each of 216 candidates, totalling 1080 fits
Best Parameters (RF): {'bootstrap': True, 'max_depth': 20, 'min_samples_leaf': 2, 'min_samples_split': 10, 'n_estimators': 50}
Best CV Accuracy (RF): 0.8060031426270408

Classification Report (Best RF Model):
              precision    recall  f1-score   support

           0       0.83      0.89      0.86      1023
           1       0.64      0.51      0.57       386

    accuracy                           0.79      1409
   macro avg       0.73      0.70      0.71      1409
weighted avg       0.78      0.79      0.78      1409


Confusion Matrix (Best RF Model):
[[914 109]
 [191 195]]
```

6. Feature Importance for Tuned Random Forest : Extracted **feature importance** from the **best-tuned Random Forest model** and visualized the top features using a **horizontal bar chart**.



7. We computed the **Precision-Recall curve** for the **best-tuned Random Forest model** and identified the **best threshold** based on the highest **F1 score**.



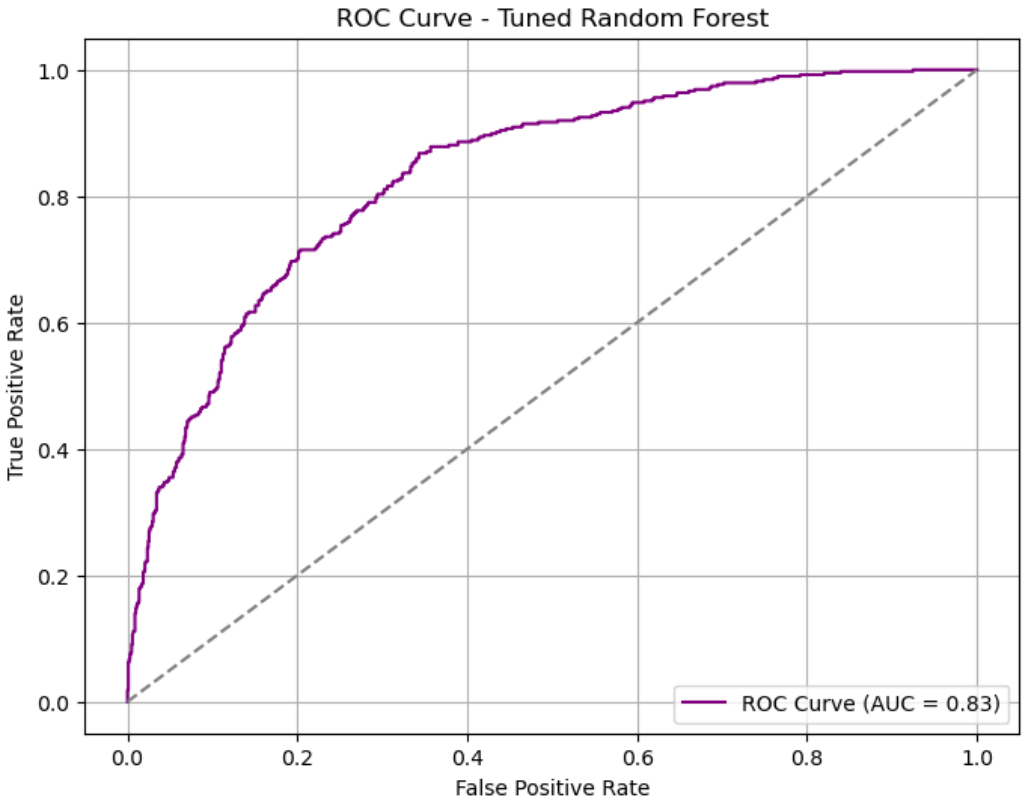
[576]: 0.3518417693417693

8. ROC AUC and Precision-Recall AUC for Tuned Random Forest- Calculated the **ROC AUC** and **Precision-Recall AUC (PR AUC)** for the **best-tuned Random Forest model**.
   - **ROC AUC**: Measures the model's ability to distinguish between churners and non-churners. A higher AUC indicates a better model.
   - **Precision-Recall AUC (PR AUC)**: Focuses on the performance for the positive class (churn) and provides an aggregate measure of the model's ability to identify churners across thresholds.

```
0.8333713197493909
0.6591469576720563
```

9 . Best threshold- `0.3518417693417693`

10. `ROC Curve and AUC for Tuned Random Forest`



12. Model Evaluation with Custom Threshold for Tuned Random Forest

```
Training Accuracy with Custom Threshold: 0.8933262335818246
Testing Accuracy with Custom Threshold: 0.7743080198722498

Classification Report (Test Data with Custom Threshold):
              precision    recall  f1-score   support

           0       0.88      0.80      0.84      1023
           1       0.57      0.72      0.63       386

    accuracy                           0.77      1409
   macro avg       0.73      0.76      0.74      1409
weighted avg       0.80      0.77      0.78      1409


Confusion Matrix (Test Data with Custom Threshold):
[[815 208]
 [110 276]]
```

Gradient Boosting Model

1. Model Initialization and Training for Gradient Boosting: Initialized the **Gradient Boosting classifier** with a fixed **random state** for reproducibility. The **Gradient Boosting model** achieved solid performance, with **accuracies** calculated on both

the training and test datasets. The **classification report** and **confusion matrix** provide insights into the model's performance in predicting churn.

2. Feature Importance for Gradient Boosting: Extracted **feature importance** from the trained **Gradient Boosting model** and visualized the top contributing features.

```
Top 10 Important Features (Gradient Boosting):
                            Feature  Importance
3                            tenure    0.332981
16     InternetService_Fiber optic    0.207831
14                     TotalCharges    0.085559
21  PaymentMethod_Electronic check    0.066072
19                Contract_Two year    0.064727
13                   MonthlyCharges    0.058338
17               InternetService_No    0.052402
18                Contract_One year    0.046117
12                 PaperlessBilling    0.017327
6                    OnlineSecurity    0.016868
```

3. **Cross-Validation for Gradient Boosting :** Performed **5-fold cross-validation** on the **Gradient Boosting model** to evaluate its performance across multiple subsets of the training data. This ensures more reliable results by testing the model's performance on different splits.

```
Cross-Validation Scores (Gradient Boosting): [0.81011535 0.7985803  0.78615794 0.78527063 0.82238011]
Mean CV Accuracy (Gradient Boosting): 0.8005008660348842
```

4. **Hyperparameter Tuning with GridSearchCV for Gradient Boosting:** Performed **hyperparameter tuning** for the **Gradient Boosting model** using **GridSearchCV** to find the best combination of hyperparameters. The grid includes variations of **n_estimators**, **learning_rate**, **max_depth**, **min_samples_split**, and **min_samples_leaf**.

```
Fitting 5 folds for each of 243 candidates, totalling 1215 fits
Best Parameters (Gradient Boosting): {'learning_rate': 0.1, 'max_depth': 3, 'min_samples_leaf': 5, 'min_samples_split': 2, 'n_estimators': 50}
Best CV Accuracy (Gradient Boosting): 0.8067131493882596

Classification Report (Best Gradient Boosting Model):
              precision    recall  f1-score   support

           0       0.83      0.90      0.86      1023
           1       0.65      0.50      0.57       386

    accuracy                           0.79      1409
   macro avg       0.74      0.70      0.71      1409
weighted avg       0.78      0.79      0.78      1409


Confusion Matrix (Best Gradient Boosting Model):
[[918 105]
 [192 194]]
```

5. Evaluation of Tuned Gradient Boosting Model:

- Evaluated the **best-tuned Gradient Boosting model** on both the **training** and **test** datasets.

- **Predictions**: Made predictions on both training and testing data using the **best model**.

- **Accuracy**: Calculated the **training accuracy** and **testing accuracy**.
- **Classification Report**: Generated a **classification report** for detailed performance metrics such as precision, recall, and F1-score.
- **Confusion Matrix**: Displayed the **confusion matrix** to show the breakdown of true positives, false positives, true negatives, and false negatives.

```
Training Accuracy (Tuned GB): 0.8168264110756124
Testing Accuracy (Tuned GB): 0.7892122072391767

Classification Report (Tuned GB):
              precision    recall  f1-score   support

           0       0.83      0.90      0.86      1023
           1       0.65      0.50      0.57       386

    accuracy                           0.79      1409
   macro avg       0.74      0.70      0.71      1409
weighted avg       0.78      0.79      0.78      1409


Confusion Matrix (Tuned GB):
[[918 105]
 [192 194]]
```

6. Feature Importance for Tuned Gradient Boosting : Extracted **feature importance** from the **best-tuned Gradient Boosting model** and visualized the top features.

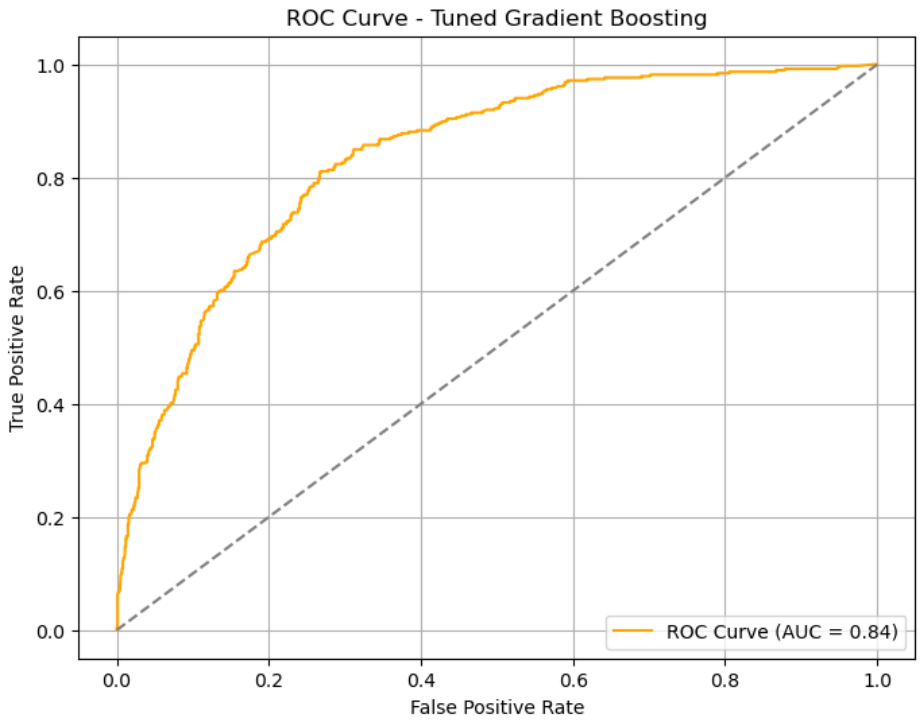
```
Top 10 Important Features (Tuned Gradient Boosting):
                          Feature  Importance
3                          tenure    0.358916
16     InternetService_Fiber optic    0.224279
21  PaymentMethod_Electronic check    0.070877
19               Contract_Two year    0.069646
14                    TotalCharges    0.056834
17              InternetService_No    0.055592
18               Contract_One year    0.049114
13                  MonthlyCharges    0.031761
6                   OnlineSecurity    0.019170
12                PaperlessBilling    0.017610
```

7. **Finding the Best Threshold for Tuned Gradient Boosting Based on F1 Score:** We computed the **Precision-Recall curve** for the **best-tuned Gradient Boosting model** and identified the **best threshold** that maximizes the **F1 score**.
   - The **best threshold** was chosen to optimize the balance between **precision** and **recall**, ensuring the best performance in predicting churners.
8. **Evaluation of Tuned Gradient Boosting with Custom Threshold :** We applied the **best threshold** to the **tuned Gradient Boosting model** to make predictions and evaluated its performance.
   - **Custom Threshold**: The threshold chosen maximizes the **F1 score** for optimal prediction of churn.

- We evaluated the model's performance using **accuracy**, **classification report**, and **confusion matrix**.

```
Training Accuracy with Custom Threshold: 0.8168264110756124
Testing Accuracy with Custom Threshold: 0.7537260468417317

Classification Report (Test Data with Custom Threshold):
              precision    recall  f1-score   support

           0       0.91      0.73      0.81      1023
           1       0.53      0.81      0.64       386

    accuracy                           0.75      1409
   macro avg       0.72      0.77      0.73      1409
weighted avg       0.81      0.75      0.77      1409


Confusion Matrix (Test Data with Custom Threshold):
[[749 274]
 [ 73 313]]
```

9. ROC Curve and AUC for Tuned Gradient Boosting : We computed the **ROC curve** for the **best-tuned Gradient Boosting model** and calculated the **AUC** (Area Under the Curve) to evaluate the model's ability to distinguish between churners and non-churners.
    - **ROC Curve**: Visualizes the trade-off between the **False Positive Rate (FPR)** and **True Positive Rate (TPR)** at various thresholds.
    - **AUC**: The **Area Under the Curve** summarizes the model's overall performance. A higher AUC indicates better discriminatory ability.




```
25]:  0.8357049519091972
```

Hybrid Model

1. Model Initialization for Hybrid Model: Initialized the individual models that will be part of the hybrid model
2. Initializing Base Models for Hybrid Model
3. Hyperparameter Tuning for Logistic Regression in Hybrid Model. We performed **GridSearchCV** for **Logistic Regression** to find the optimal hyperparameters. **Best Model**: The **best Logistic Regression model** is selected based on **accuracy** from **5-fold cross-validation**.

```
Best Parameters (Logistic Regression): {'C': 100, 'penalty': 'l2', 'solver': 'liblinear'}
Best CV Accuracy (Logistic Regression): 0.8068912420941811
```

4. Hyperparameter Tuning for Random Forest in Hybrid Model
5. We performed **GridSearchCV** for the **Random Forest** model to identify the best hyperparameters. The **parameter grid** includes various values for:
    a. **`n_estimators`** (number of trees in the forest)
    b. **`max_depth`** (maximum depth of the trees)
    c. **`min_samples_split`** (minimum samples required to split an internal node)
    d. **`min_samples_leaf`** (minimum samples required to be at a leaf node)

- **Best Model**: The **best Random Forest model** is selected based on **accuracy** from **5-fold cross-validation**.

```
Fitting 5 folds for each of 81 candidates, totalling 405 fits
Best Parameters (Random Forest): {'max_depth': 20, 'min_samples_leaf': 2, 'min_samples_split': 10, 'n_estimators': 50}
Best CV Accuracy (Random Forest): 0.8060031426270408
```

6. Hyperparameter Tuning for Gradient Boosting in Hybrid Model- We performed **GridSearchCV** for the **Gradient Boosting** model to identify the best hyperparameters.

- **`n_estimators`** (number of boosting stages or trees),
- **`learning_rate`** (step size for each tree),
- **`max_depth`** (maximum depth of the individual trees),
- **`min_samples_split`** (minimum number of samples required to split an internal node),
- **`min_samples_leaf`** (minimum number of samples required to be at a leaf node).
- **Best Model**: The **best Gradient Boosting model** is selected based on **accuracy** from **5-fold cross-validation**.

```
Fitting 5 folds for each of 243 candidates, totalling 1215 fits
Best Parameters (Gradient Boosting): {'learning_rate': 0.1, 'max_depth': 3, 'min_samples_leaf': 5, 'min_samples_split': 2, 'n_estimators': 50}
Best CV Accuracy (Gradient Boosting): 0.8067131493882596
```

7. Hyperparameter Tuning for SVM in Hybrid Model. We performed **GridSearchCV** for the **Support Vector Machine (SVM)** model to find the best hyperparameters. The **parameter grid** includes:

- **`C`** (regularization parameter),
- **`kernel`** (type of kernel function used in SVM),
- **`gamma`** (kernel coefficient).
- **Best Model**: The **best SVM model** is selected based on **accuracy** from **5-fold cross-validation**.

```
Fitting 5 folds for each of 24 candidates, totalling 120 fits
Best Parameters (SVM): {'C': 0.1, 'gamma': 'scale', 'kernel': 'linear'}
Best CV Accuracy (SVM): 0.8019202491406634
```

8. Hyperparameter Tuning for KNN in Hybrid Model We performed **GridSearchCV** for the **K-Nearest Neighbors (KNN)** model to find the best hyperparameters. he **parameter grid** includes:

- **`n_neighbors`** (number of neighbors to use),
- **`weights`** (weighting function used in prediction, either 'uniform' or 'distance'),

- **p** (distance metric used, where p=1 for Manhattan distance and p=2 for Euclidean distance).
- **Best Model**: The **best KNN model** is selected based on **accuracy** from **5-fold cross-validation**.

```
Fitting 5 folds for each of 16 candidates, totalling 80 fits
Best Parameters (KNN): {'n_neighbors': 9, 'p': 1, 'weights': 'uniform'}
Best CV Accuracy (KNN): 0.7813308410861449
```

9. Training and Evaluating Stacked Model : We created a **Stacking Classifier** using the previously tuned base models:

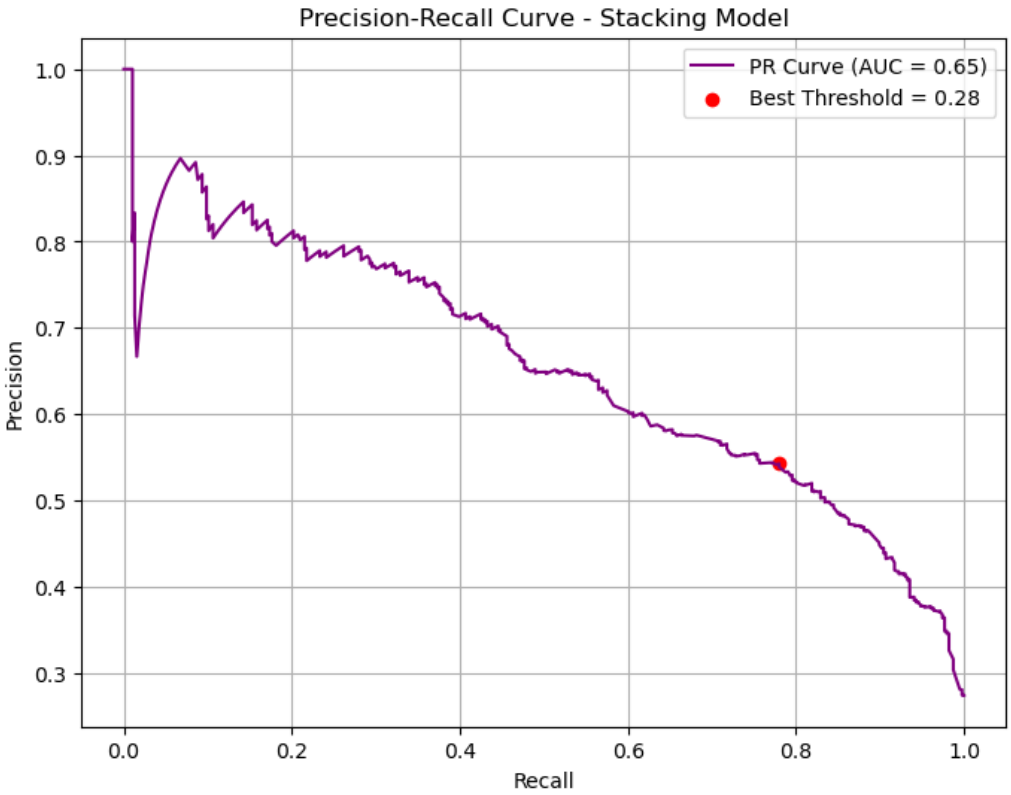
- **Base Models**: Logistic Regression, Random Forest, Gradient Boosting, SVM, and KNN.
- **Final Estimator**: Gradient Boosting Classifier (used to combine the predictions from base models).
- **Cross-Validation**: We used 5-fold cross-validation to train and evaluate the stacked model.

```
Training Accuracy (Stacking Model): 0.8446929357472488
Testing Accuracy (Stacking Model): 0.7892122072391767

Classification Report (Stacking Model):
              precision    recall  f1-score   support

           0       0.83      0.90      0.86      1023
           1       0.65      0.51      0.57       386

    accuracy                           0.79      1409
   macro avg       0.74      0.70      0.71      1409
weighted avg       0.78      0.79      0.78      1409


Confusion Matrix (Stacking Model):
[[917 106]
 [191 195]]
```

10. Evaluating Stacked Model with Custom Threshold. After training the **stacked model**, we:
    - Calculated the **Precision-Recall curve** for the stacked model.
    - Identified the **best threshold** based on the **F1 score**.
    - Applied the **best threshold** to the model's predicted probabilities to make final predictions.
    - Evaluated the model's performance using **accuracy**, **classification report**, and **confusion matrix**.

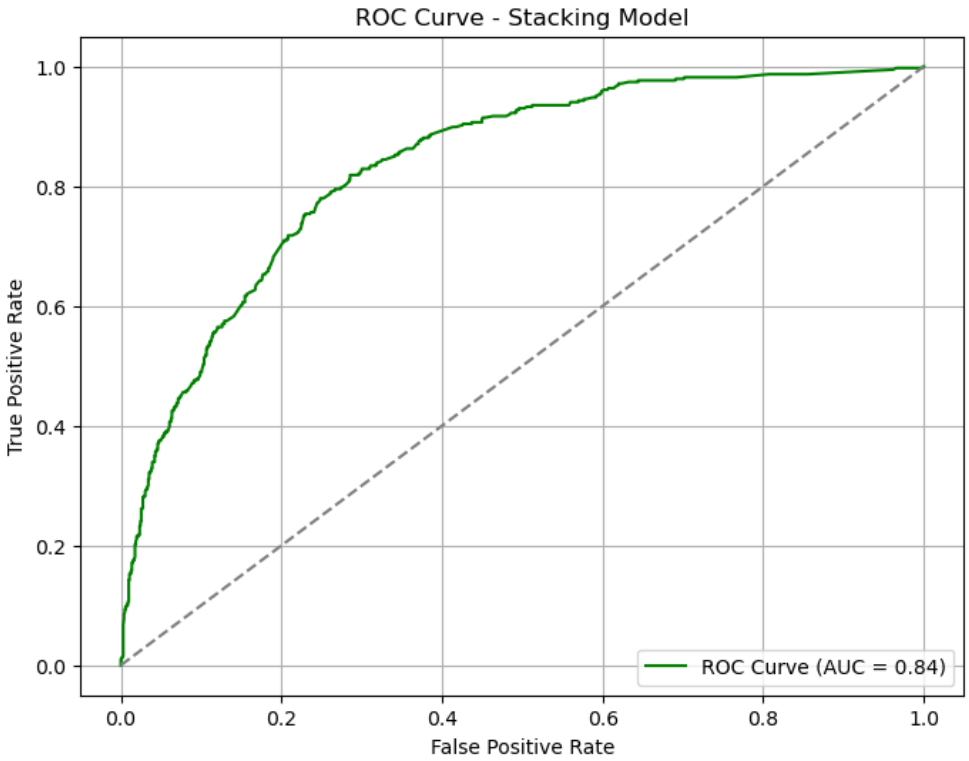
```
Best Threshold for Stacking Model: 0.28408626468119763
Training Accuracy with Custom Threshold: 0.8446929357472488
Testing Accuracy with Custom Threshold: 0.759403832505323

Classification Report (Test Data with Custom Threshold):
              precision    recall  f1-score   support

           0       0.90      0.75      0.82      1023
           1       0.54      0.78      0.64       386

    accuracy                           0.76      1409
   macro avg       0.72      0.77      0.73      1409
weighted avg       0.80      0.76      0.77      1409


Confusion Matrix (Test Data with Custom Threshold):
[[769 254]
 [ 85 301]]
```

11. Precision-Recall Curve for Stacked Model. We plotted the **Precision-Recall curve** for the **stacking model** to evaluate its performance.
    - The **AUC (Area Under the Curve)** was calculated to assess the overall performance of the model in distinguishing churners from non-churners.
    - The **best threshold** is marked on the curve to show the point that optimizes the **F1 score**.



12. ROC Curve for Stacked Model - We plotted the **ROC curve** for the **stacking model** to assess its ability to distinguish between churners and non-churners.

- The **AUC (Area Under the Curve)** was calculated to summarize the model's overall ability to classify churners correctly.
- The **ROC curve** visualizes the trade-off between the **False Positive Rate (FPR)** and **True Positive Rate (TPR)**.



Model Interpretability using SHAP and LIME

1. Initializing SHAP Explainer for Gradient Boosting Model
2. Computing SHAP Values for Test Data
3. Generating Feature Importance Summary Plot (Bar Plot)- The **summary plot** provides a global view of feature importance. This bar plot shows how much each feature influences the model's output across all predictions.

SHAP for Random Forest-

1. Initialize SHAP Explainer for Random Forest.
2. Compute SHAP Values for Test Data
3. summary Plot for Global Feature Importance