Bank Churn Rate Final Project Report

Introduction & Problem Statement:

Determining or predicting churn rate is key in increasing business effectiveness and allowing for targeted marketing and resources to increase customer retention. This is especially the case for a bank who released their user data on Kaggle for assistance on predicting churn in order to reduce it. The goal of this project is to provide a churn prediction model to mitigate the bank's churn rate and increase overall bank revenue by 20% for the next financial quarter.

Data Wrangling:

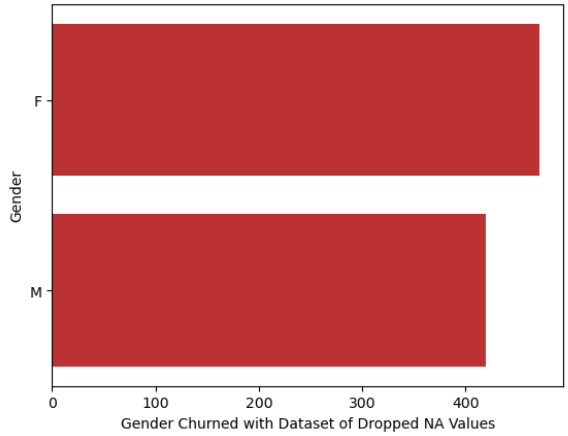
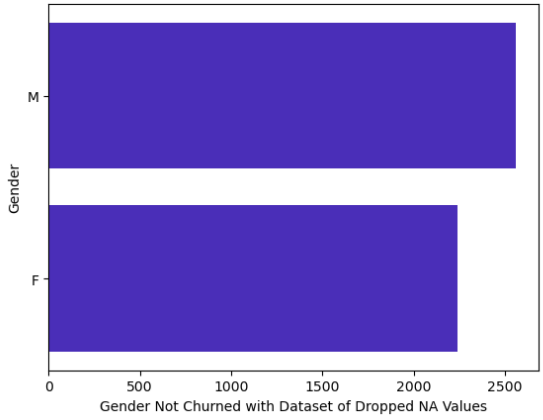
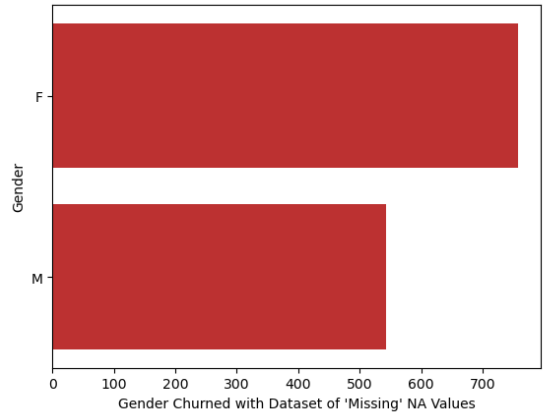
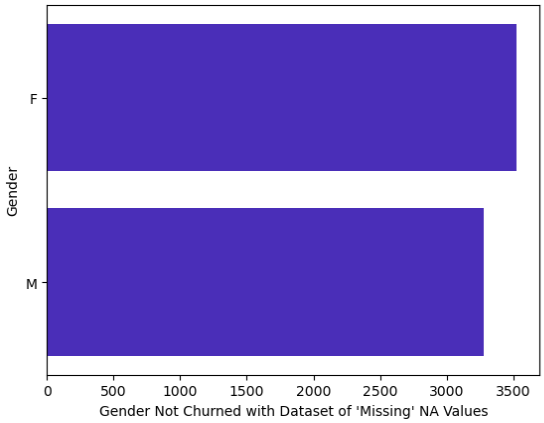
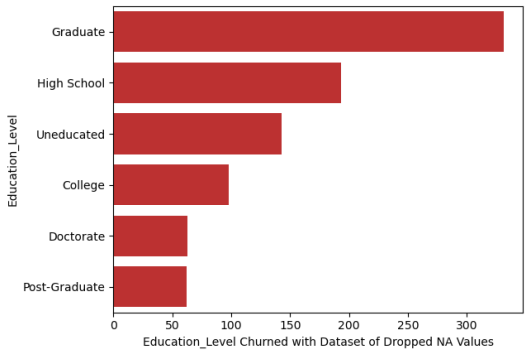
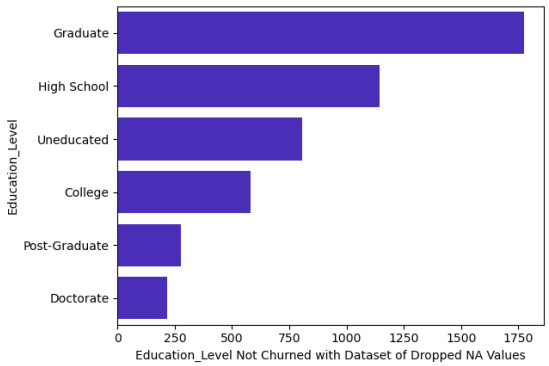
In order to begin steps of model development, the dataset was first assessed and evaluated. It seemed that applying the standard df.isnull() did not seem at first to locate any missing values for the dataset, however upon closer inspection the missing values were actually imputed with the value of 'Unknown.' Therefore, replacing all the 'Unknown' values with NaN retrieved the below results for the columns that had missing or null values and their respective percent of missingness from the dataset.

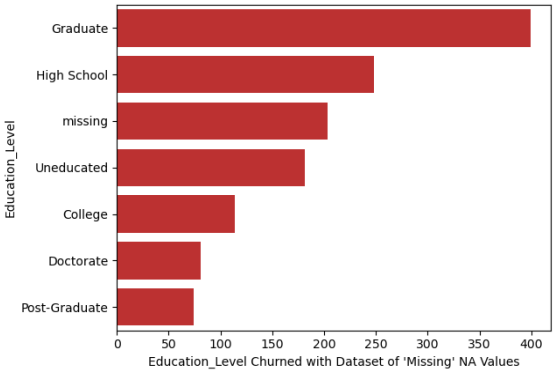
	count	% missing
Education_Level	1205	14.874707
Income_Category	889	10.973954
Marital_Status	579	7.147266

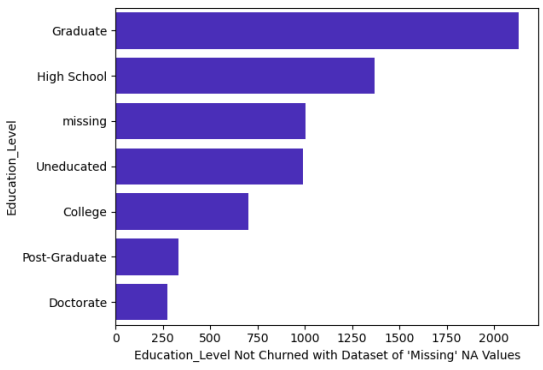
In order to deal with the 'missingness' for the above values, two datasets were created and then assessed throughout the project to assess future model performance. One where the above columns' null data were replaced with the categorical value or string of 'missing' and one where any observations in the above three columns that had null values were dropped. These two datasets as reference to subsequent steps are referred to respectively as 'missing' dataset and 'dropped' dataset respectively.

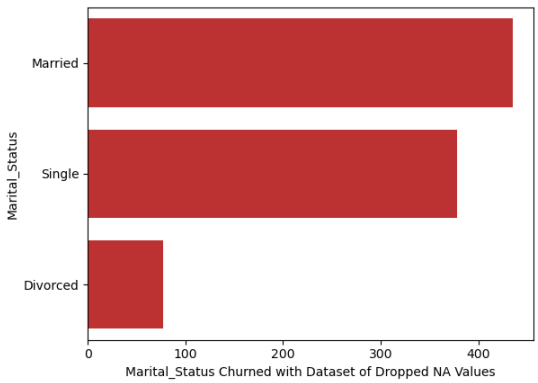
Exploratory Data Analysis:

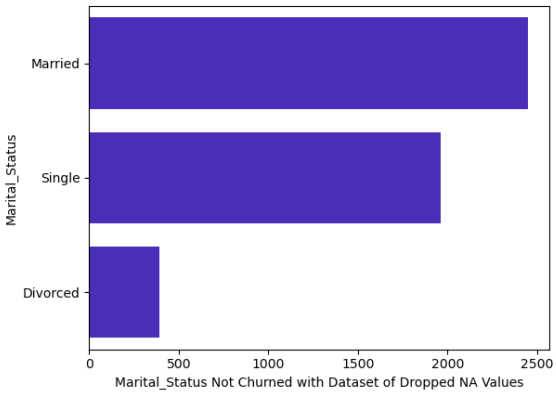
The two datasets were assessed based on categorical features to determine churn not churn user base:

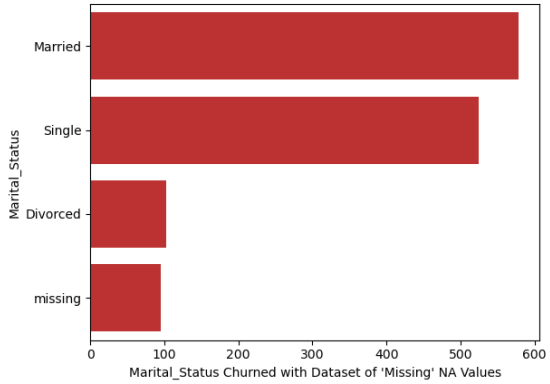
User Base Churn	User Base Not Churn
	
	
	

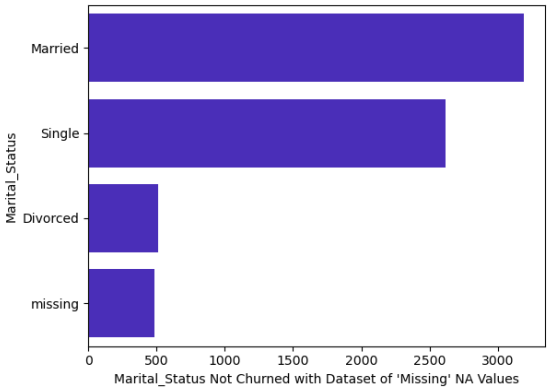

Graduate
High School
missing
Uneducated
College
Doctorate
Post-Graduate
Education_Level
0
50
100
150
200
250
300
350
400
Education_Level Churned with Dataset of 'Missing' NA Values

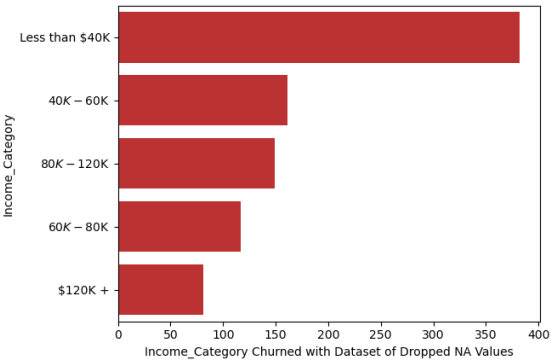

Graduate
High School
missing
Uneducated
College
Post-Graduate
Doctorate
Education_Level
0
250
500
750
1000
1250
1500
1750
2000
Education_Level Not Churned with Dataset of 'Missing' NA Values

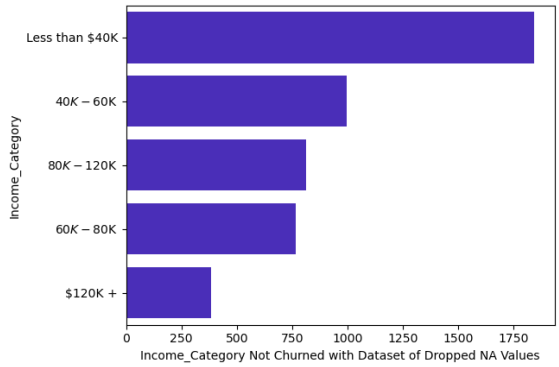

Married
Single
Divorced
Marital_Status
0
100
200
300
400
Marital_Status Churned with Dataset of Dropped NA Values

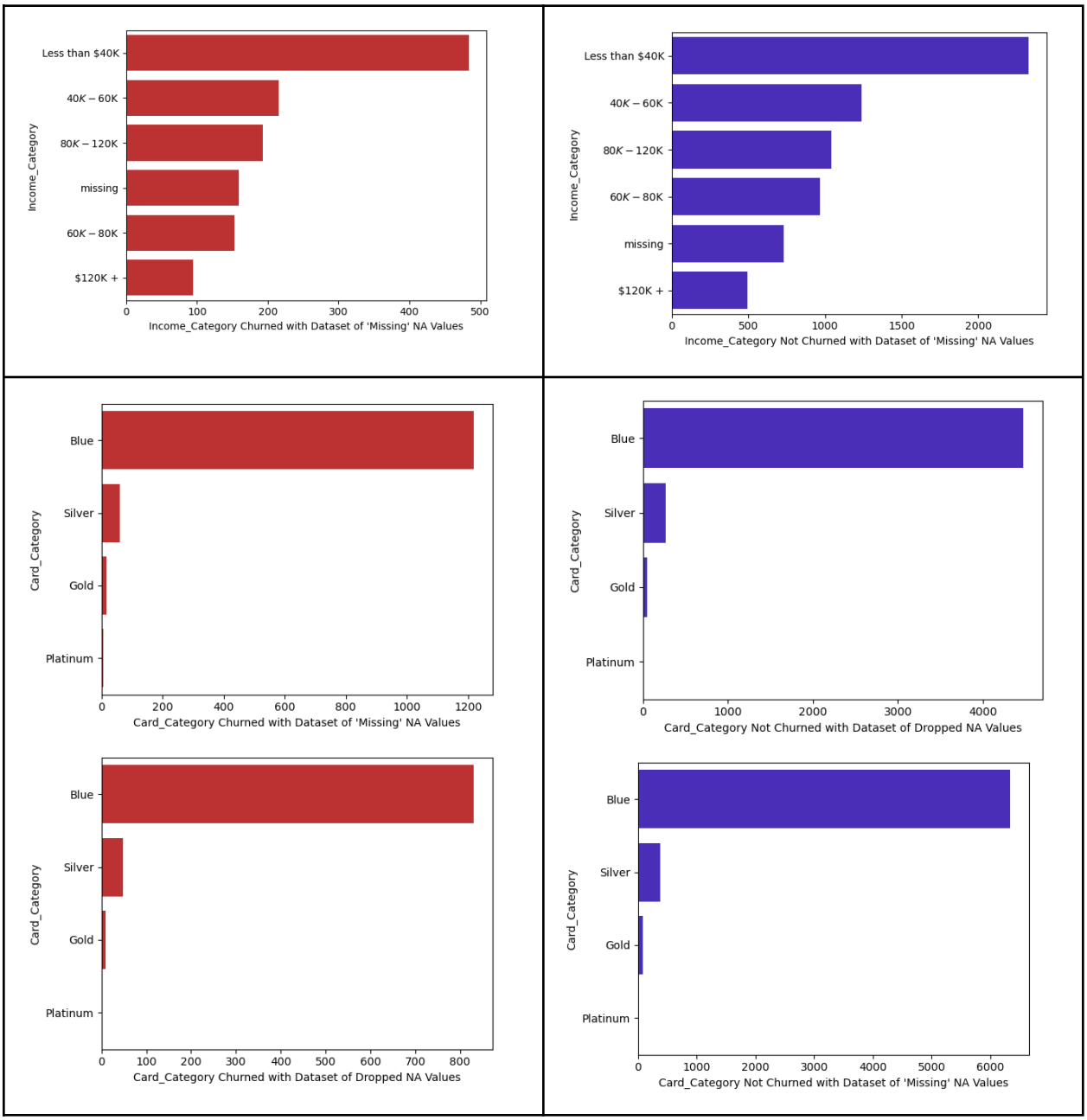

Married
Single
Divorced
Marital_Status
0
500
1000
1500
2000
2500
Marital_Status Not Churned with Dataset of Dropped NA Values


Married
Single
Divorced
missing
Marital_Status
0
100
200
300
400
500
600
Marital_Status Churned with Dataset of 'Missing' NA Values


Married
Single
Divorced
missing
Marital_Status
0
500
1000
1500
2000
2500
3000
Marital_Status Not Churned with Dataset of 'Missing' NA Values


Less than $40K
40K − 60K
80K − 120K
60K − 80K
$120K +
Income_Category
0
50
100
150
200
250
300
350
400
Income_Category Churned with Dataset of Dropped NA Values


Less than $40K
40K − 60K
80K − 120K
60K − 80K
$120K +
Income_Category
0
250
500
750
1000
1250
1500
1750
Income_Category Not Churned with Dataset of Dropped NA Values



Viewing the above analysis, it seems as a whole between both the ‘missing’ and ‘dropped’ datasets there are more females than males, the most common education level is graduate, the most common marital status are married, the most common income category are those in the less than 40k category, and most of the users are Blue card holders. A note to mention is that there is a seeming discrepancy in terms of data for gender between ‘missing’ and ‘dropped’ datasets where in the ‘dropped’ dataset males seemed to be larger than females. However, this seems to indicate that Females tended to leave more values unknown than males possibly for protecting their own personal information.

Taking this above assessment and all the features from the dataset, further analysis will be needed to determine which features have some correlation with the target feature of prediction 'Attrition_Flag.'

'Dropped' Correlation List 'Attrition_Flag'		'Missing' Correlation List 'Attriton_Flag'	
Attrition_Flag	1.000000	Attrition_Flag	1.000000
Total_Trans_Ct	0.364835	Total_Trans_Ct	0.380618
Total_Ct_Chng_Q4_Q1	0.281174	Total_Ct_Chng_Q4_Q1	0.288937
Total_Revolving_Bal	0.268979	Total_Revolving_Bal	0.263335
Avg_Utilization_Ratio	0.185690	Avg_Utilization_Ratio	0.179838
Total_Trans_Amt	0.170178	Total_Trans_Amt	0.178078
Total_Relationship_Count	0.149125	Total_Relationship_Count	0.150889
Total_Amt_Chng_Q4_Q1	0.133285	Total_Amt_Chng_Q4_Q1	0.128559
Gender_M	0.045186	Gender_M	0.047382
Income_Category_$60K - $80K	0.028017	Income_Category_$60K - $80K	0.026338
Income_Category_$40K - $60K	0.023828	Credit_Limit	0.024437
Education_Level_High School	0.018403	Education_Level_College	0.018937
Credit_Limit	0.017750	Marital_Status_Married	0.017872
Marital_Status_Married	0.015278	Income_Category_$40K - $60K	0.015050
Education_Level_College	0.012472	Card_Category_Silver	0.014854
Education_Level_Uneducated	0.007257	Education_Level_High School	0.009930
Card_Category_Gold	0.003949	Education_Level_Uneducated	0.006614
Card_Category_Silver	0.003173	Income_Category_$80K - $120K	0.005148
Income_Category_$80K - $120K	0.001899	Education_Level_Graduate	0.004847
Card_Category_Blue	-0.001150	Avg_Open_To_Buy	0.000803
Education_Level_Graduate	-0.001751	Income_Category_$120K +	-0.000831
Avg_Open_To_Buy	-0.006245	Marital_Status_missing	-0.002723
Marital_Status_Divorced	-0.006688	Card_Category_Gold	-0.003397
Dependent_count	-0.008122	Marital_Status_Divorced	-0.006305
Marital_Status_Single	-0.011789	Card_Category_Blue	-0.008463
Income_Category_$120K +	-0.014601	Education_Level_missing	-0.009100
Education_Level_Post-Graduate	-0.018346	Education_Level_Post-Graduate	-0.013375
Months_on_book	-0.018599	Marital_Status_Single	-0.013437
Card_Category_Platinum	-0.022579	Income_Category_missing	-0.017580
Customer_Age	-0.024498	Card_Category_Platinum	-0.020284
Income_Category_Less than $40K	-0.033687	Dependent_count	-0.020550
Education_Level_Doctorate	-0.043378	Months_on_book	-0.022270
Gender_F	-0.045186	Income_Category_Less than $40K	-0.023134
Months_Inactive_12_mon	-0.160701	Customer_Age	-0.035203
Contacts_Count_12_mon	-0.203447	Education_Level_Doctorate	-0.039483
		Gender_F	-0.047382
		Months_Inactive_12_mon	-0.156552
		Contacts_Count_12_mon	-0.211327

After applying pd.get_dummies() and sorting the list based on its correlation level with the target feature 'Attrition_Flag', a few relationships were seen with 'Attrition_Flag.' It seems the moderate positive correlations come from feature values in the Total_Trans_Ct, Total_Ct_Chng_Q4_Q1, Total_Revolving_Bal, Avg_Utilization_Ratio, Total_Trans_Amt, Total_Relationship_Count, and Total_Amt_Chng_Q4_Q1 columns. There is an intuitive sense to this, as the customer's spending pattern would seem to affect if the customer is in fact more likely to churn or not. However, what also seems of interest is that how many dependents the customer had or their total relationship count also positively affected some of the variance with the Attrition_Flag column. And that females had a negative correlation versus male users having a more positive level. Also it seems that the customers age is negatively correlated meaning

younger members seemed to be less likely to churn than the older demographic. Also between the ‘Dropped’ and ‘Missing’ datasets, the datasets seem to overall agree with each other as per the top positive and top negative correlation levels with variances in the middle due to differing datasets.

Preprocessing & Training:

In order to avoid the dummy variable trap for model development or potentially having issues between because of multicollinearity within the dummy variables, the different categorical features were instead arranged to instead have k-1 features where k is the number of unique categories for the feature. The dropped values chosen were values that occurred the most frequently in the datasets as well as Gender_F since this was the most common over most of the datasets.

Scikit learns train_test_split and StandardScaler functions were applied to both ‘dropped’ and ‘missing’ datasets in order to create arbitrary 70/30 splits of test and train data and appropriately scale each of the features for next steps of model development.

Modeling:

Different predictive models for classification were developed and trained with the 70/30 train test sets as well as 5 fold cross-validated with a list of associated model hyperparameters and then assessed based on the overall Accuracy, Precision, Recall, F1 Score, ROC_AUC and training time to determine best model performer. Models assessed were Logistic Regression, RandomForestClassifier with Bagging, Gradient Boosting Classifier which is random forest but with boosting, Support Vector Machines, AdaBoost Classifier to improve RandomForest with Optimal Parameters and GradientBoosting with Optimal Parameters, and Neural Network Multi-Layer Perceptron Classifier models.

Below are the following results table and ROC curves:

Results Table ‘Missing’ Dataset:

Model Name	Accuracy_score	Precision_score	Recall_score	f1_score	ROC_AUC_score	Training Time (Sec)
Logistic Regression	0.906211	0.730769	0.580556	0.647059	0.771688	15.40
Random Forest Classifier	0.939120	0.886861	0.675000	0.766562	0.830016	62.87

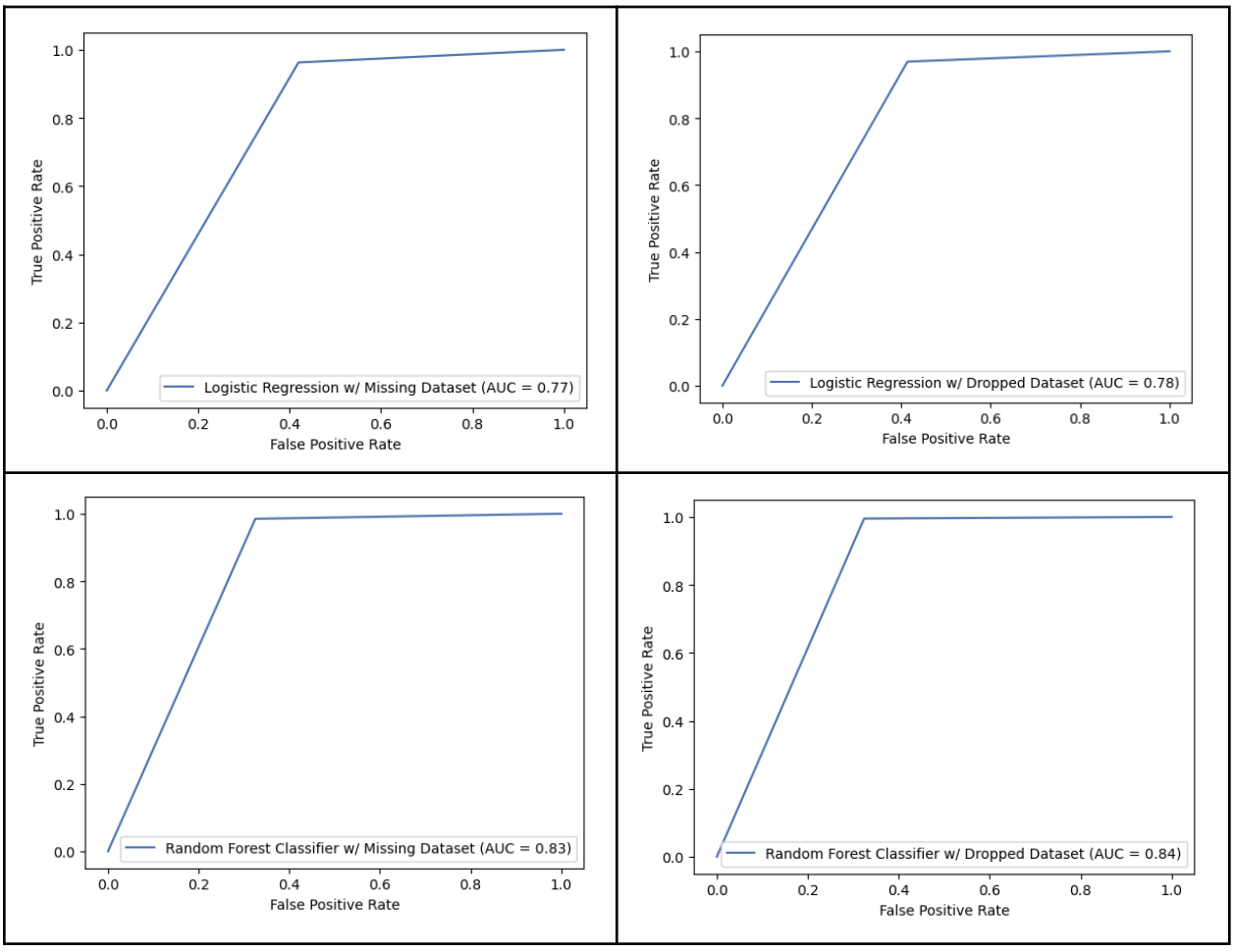
Gradient Boosting Classifier	0.967914	0.927273	0.850000	0.886957	0.919206	846.87
Support Vector Classifier	0.915673	0.725948	0.691667	0.708393	0.823139	80.43
AdaBoost w/ Random Forest Classifier	0.963801	0.944444	0.802778	0.867868	0.897285	1535.73
AdaBoost w/ Gradient Boosting Classifier	0.969148	0.930514	0.855556	0.891462	0.922225	213.50
MLP Neural Network Classifier	0.933772	0.806154	0.727778	0.961456	0.848679	514.61

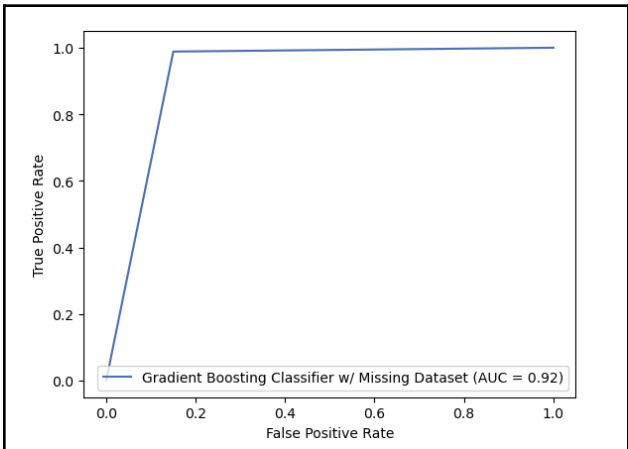
Results Table 'Dropped' Dataset:

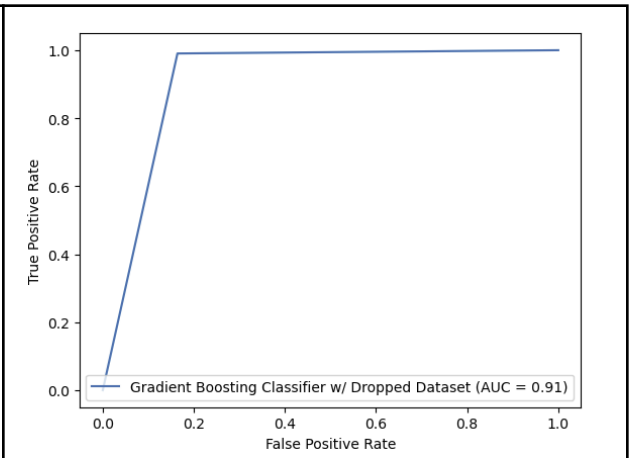
Model Name	**Accuracy_score**	**Precision_score**	**Recall_score**	**f1_score**	**ROC_AUC_score**	**Training Time (Sec)**
Logistic Regression	0.914470	0.760638	0.586066	0.662037	0.777653	14.31
Random Forest Classifier	0.949619	0.959302	0.676230	0.793269	0.835722	52.23
Gradient Boosting Classifier	0.968366	0.935780	0.836066	0.883117	0.913248	603.65
Support Vector Classifier	0.930287	0.796209	0.688525	0.738462	0.829566	40.19

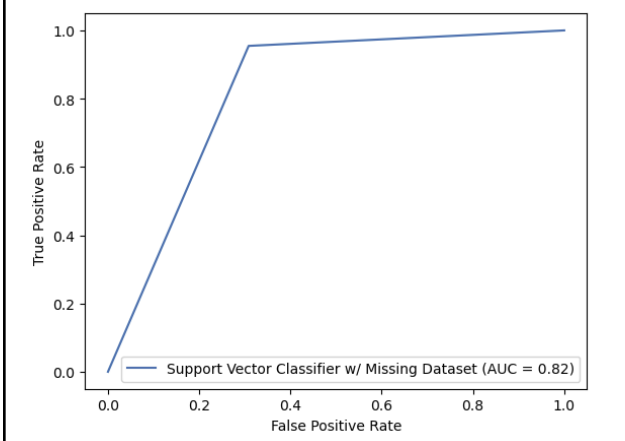
AdaBoost w/ Random Forest Classifier	0.961921	0.963731	0.762295	0.851259	0.878755	1171.65
AdaBoost w/ Gradient Boosting Classifier	0.971880	0.945455	0.852459	0.896552	0.922128	926.41
MLP Neural Network Classifier	0.937317	0.827751	0.709016	0.963864	0.842205	351.12

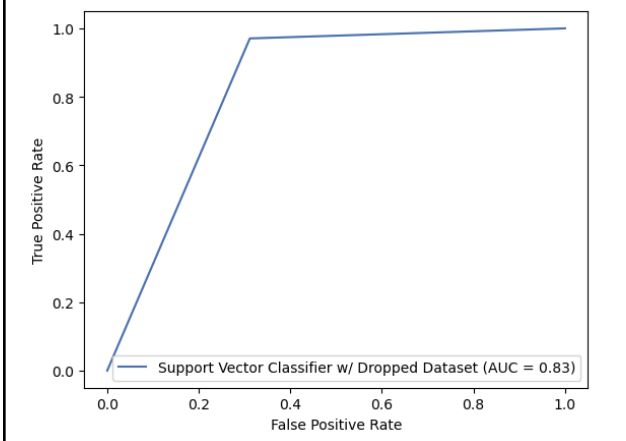
ROC Curves Modeling Metrics:

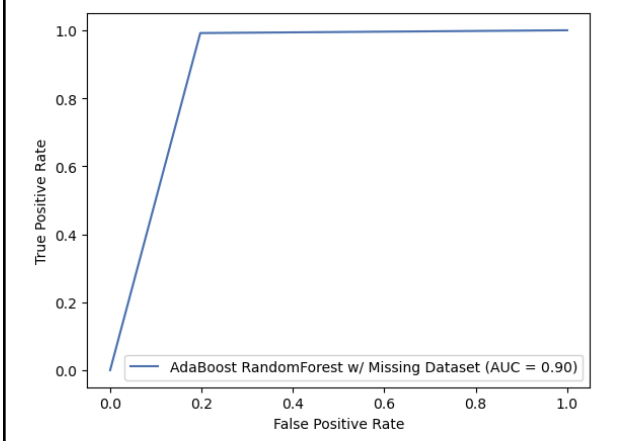


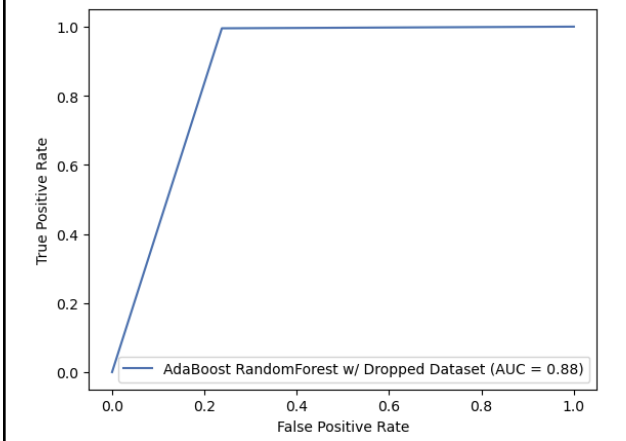

True Positive Rate
False Positive Rate
Gradient Boosting Classifier w/ Missing Dataset (AUC = 0.92)

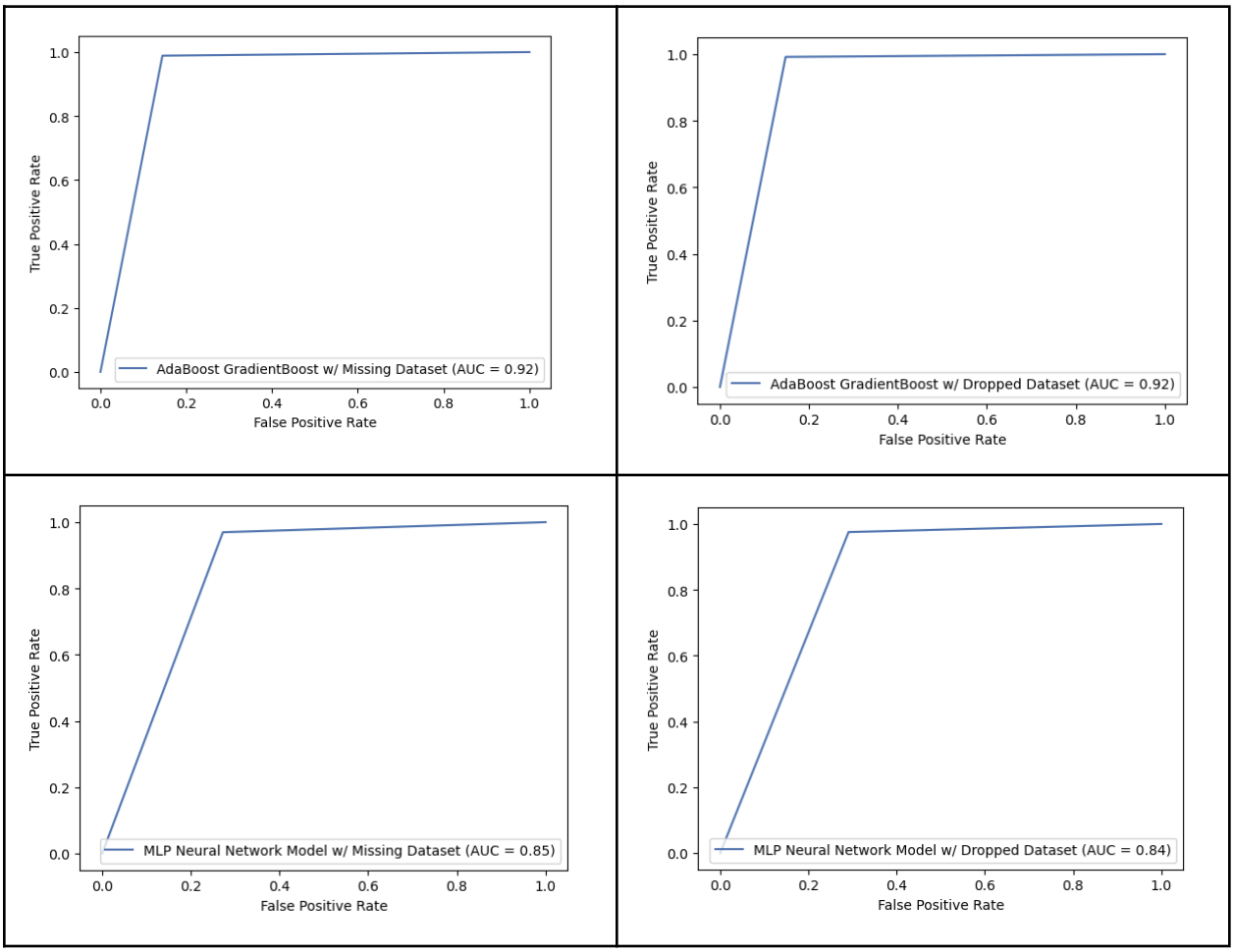

True Positive Rate
False Positive Rate
Gradient Boosting Classifier w/ Dropped Dataset (AUC = 0.91)


True Positive Rate
False Positive Rate
Support Vector Classifier w/ Missing Dataset (AUC = 0.82)


True Positive Rate
False Positive Rate
Support Vector Classifier w/ Dropped Dataset (AUC = 0.83)


True Positive Rate
False Positive Rate
AdaBoost RandomForest w/ Missing Dataset (AUC = 0.90)


True Positive Rate
False Positive Rate
AdaBoost RandomForest w/ Dropped Dataset (AUC = 0.88)



Results & Conclusion:

Looking at the above results table and ROC_AUC curves, Logistic Regression, though having the shortest training time, turned out to be one of the worst performers for both the missing and dropped datasets per predicting label of churn or '0'. The best performer overall for both the dropped and missing datasets appears to be the Gradient Boosting Classifier model when applied with an AdaBoost Classifier model, slightly outperforming its individual Gradient Boosting Classifier counterpart. However, the AdaBoost Classifier model with Optimal Gradient Boosting is not the most intuitive to reproduce as it requires first optimizing a Gradient Boosting model then also training with the AdaBoost classifier.Therefore, in terms of ease of reproducibility and scale, just training and deploying to production the sole Gradient Boosting Model seems sufficient as the prediction model as it was able to achieve ROC_AUC scores of 0.92 and 0.91 for missing and dropped datasets respectively which is significantly greater than random guessing churn within the dataset.

The limitation, however, is the training time in order to acquire the optimal parameters for the Gradient Boosting model as it was on the higher end in terms of models and their respective training time.

In terms of whether initially dropping the observations that had null values or imputing them with the category of missing had any affect on the performance of the different models, the ‘dropped’ dataset lead to Logistic Regression, Random Forest, and Support Vector Classifier models having a slightly better performance than its imputed counterpart, however performance was slightly worse for the ‘dropped’ dataset when applied with the Gradient Boosting, and the AdaBoost Classifiers in terms of recall in predicting 0 label or churn, f1_score, and ROC_AUC_score. And since the best performers were the Gradient Boosting and AdaBoost Classifier models, it seems that initially imputing the ‘Unknown’ value in the dataset with missing improved the already stronger performing models. Therefore, in terms of future modeling, imputing the dataset with ‘missing’ instead of just dropping the ‘Unknown’ values will be a suggested step moving forward.

In terms of correlations as well as per the datasets user base, it seems that additional efforts along with deploying the Gradient Boosting model can be made as well to potentially increase customer retention. It seems that females and the older audience are more likely to attrite therefore having market deals that are friendlier towards females or the older audience will help in customer retention. Or potential efforts can be made instead to maximize the seeming not churned population such as making marketing efforts towards the younger married male population with children or dependents.

Future Work:

Future work will be to further fine tune the Gradient Boosting Model in order to reduce its respective training time as well as further optimize the Multi-Layer Perceptron Neural Network Classifier model in order to see if the model with further tuned hyperparameters can outperform the Gradient Boosting model as well as the Optimal Gradient Boosting Model with the AdaBoost Classifier model.