VISIT WITH US TRAVEL PACKAGE PREDICT CASE STUDY

JAMEEL KHAN

Objective

- The Policy Maker of the company "Visit with us" want to enable and establish a viable business model to expand the customer base. A viable business model is a central concept that helps you to understand the existing ways of doing the business and how to change the ways for the benefit of the tourism sector.
- One of the ways to expand the customer base is to introduce a new offering of packages.

- To predict which customer is more likely to purchase the newly introduced travel package.

- Provide Key insights and Recommendations from Data Analysis

Dataset Characteristics

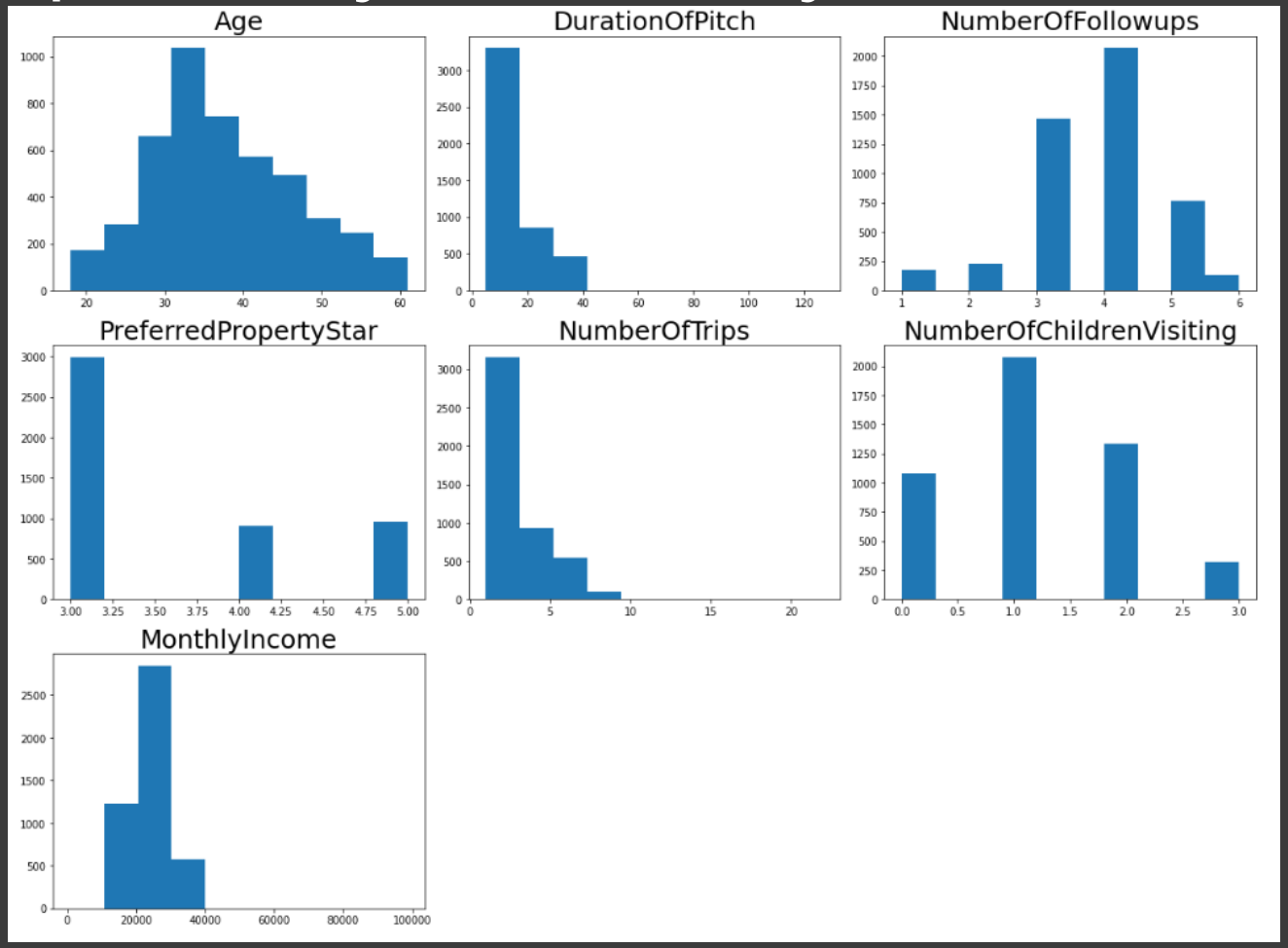
Dataset Column	Description
ID	Customer ID
Age	Customer's age in completed years
ProdTaken	Product taken flag
TypeofContact	How customer was contacted (Company Invited or Self Inquiry
CityTier	City tier
Occupation	Occupation of customer
Gender	Gender of customer
NumberOfPersonVisited	Total number of person came with customer
PreferredPropertyStar	Preferred hotel property rating by customer
MaritalStatus	Marital status of customer
NumberOfTrips	Average number of the trip in a year by customer
OwnCar	Customers owns a car flag
NumberOfChildrenVisited	Total number of children visit with customer
Designation	Designation of the customer in the current organization
MonthlyIncome	Gross monthly income of the customer
PitchSatisfactionScore	Sales pitch satisfactory score
ProductPitched:	Product pitched by a salesperson
NumberOfFollowups	Total number of follow up has been done by sales person after sales pitch
DurationOfPitch	:Duration of the pitch by a salesman to customer

The dataset contains Customer centric and other metrics of customers

Observations on Data Set:

1. There are 20 columns of data for each customer, with a total of 5000 rows of data
2. The dataset has some missing data that will get mode imputation

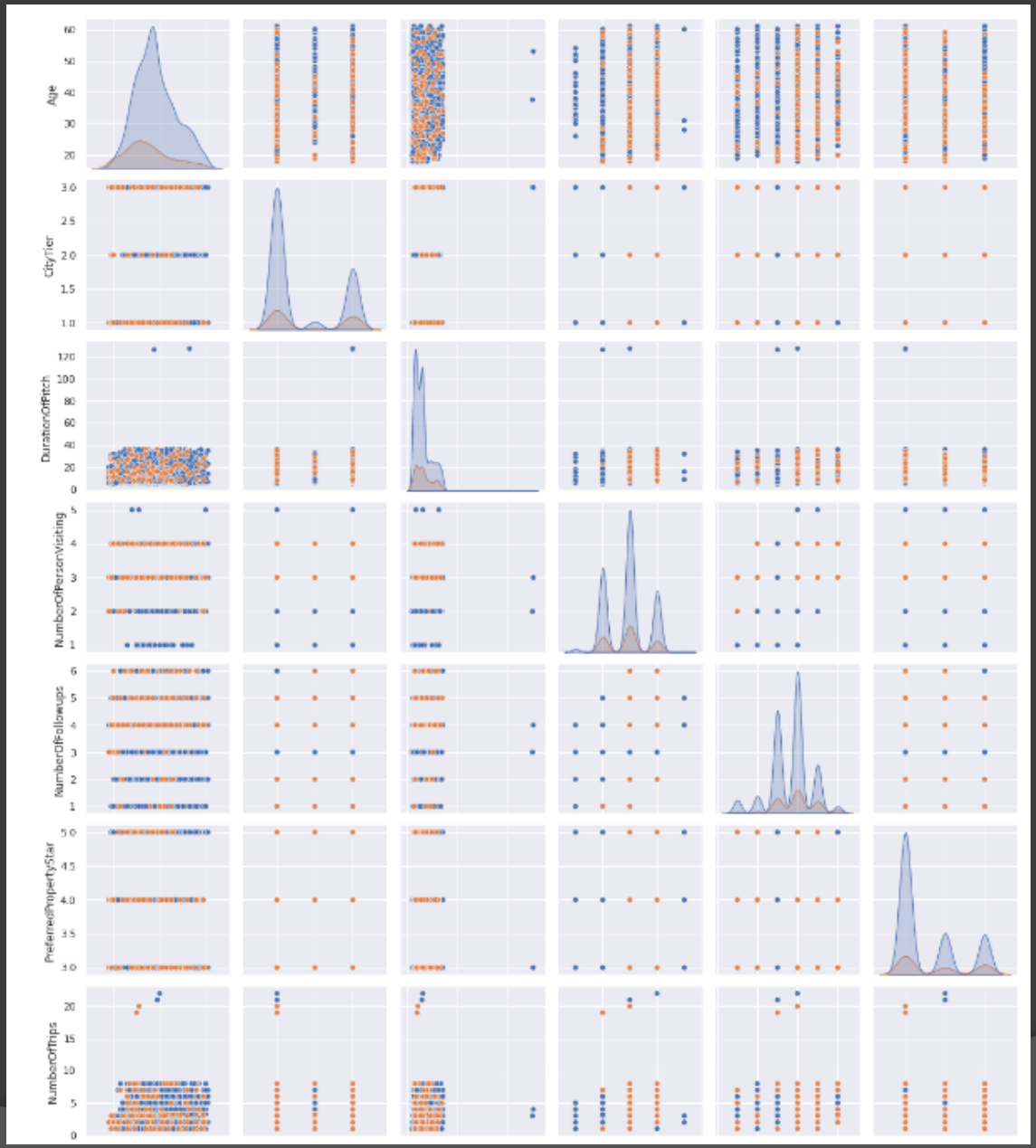
Exploratory Data Analyses – Distribution



Observations on Numerical Data Distribution in the dataset:

1. Age is normally distributed. Monthly Income are rightly skewed.
2. There are some outliers but we will leave them untreated

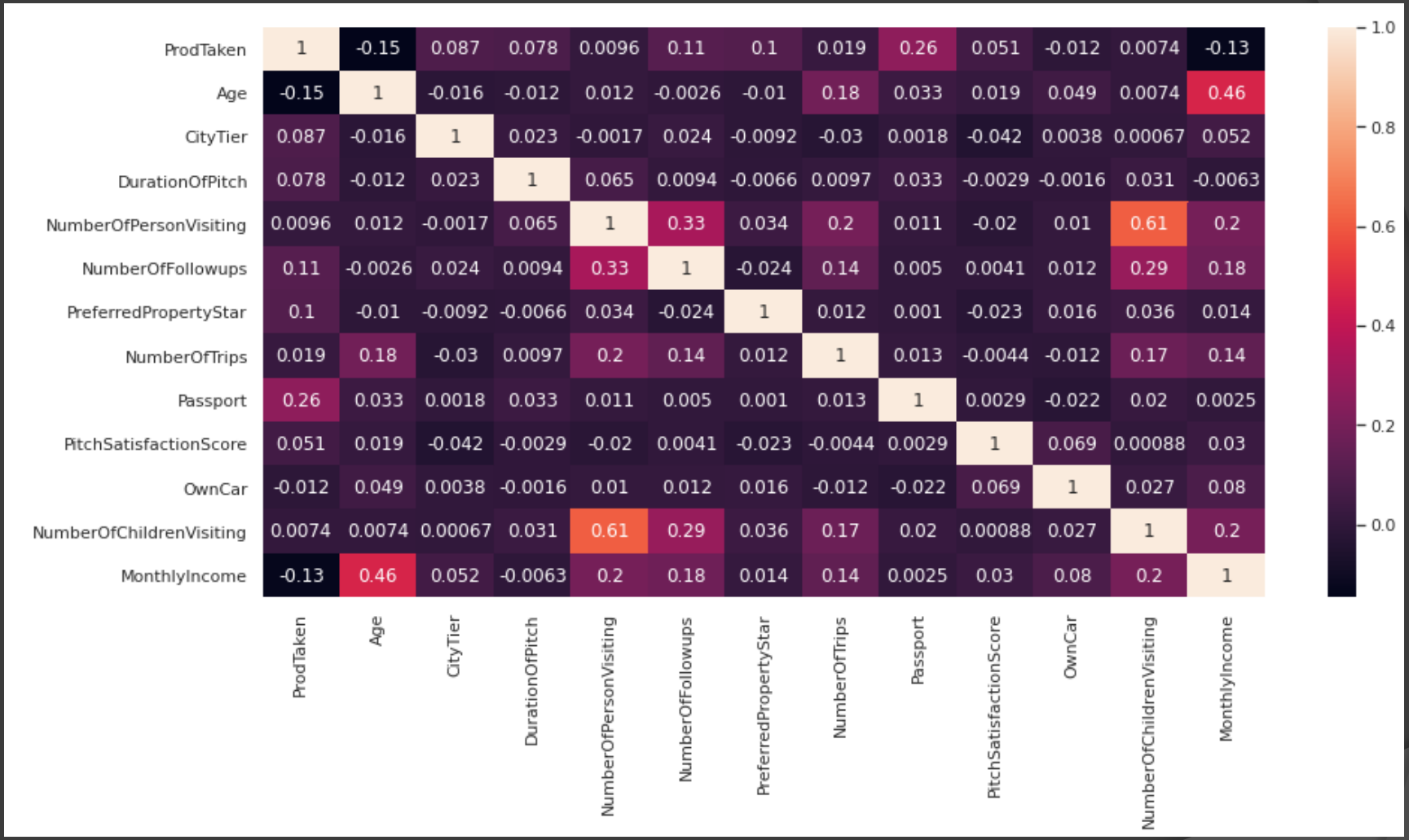
Exploratory Data Analyses – Correlation



Observation on pair plots

1. Using hue of Product Taken, the data imbalance is clear in each feature
2. No strong correlations between features. We will keep all features for modelling.
3. Age and MonthlyIncome are linearly related, as expected.
4. Most of the features are categorical - We'll one-hot encode these features before modelling.
5. Self Inquiry is much higher than Comp any Invited. Needs to be investigated f or better marketing strategy
6. Couples and families are large demog raphic
7. Deluxe and Basic packages are popul ar packages.

Exploratory Data Analyses – Correlation



Observation on Correlation Matrix of Numeric Features

1. No strong correlations between features. We will keep all features for modelling.
2. Age and MonthlyIncome are linearly related, as expected.

ML Model Building – Decision Tree

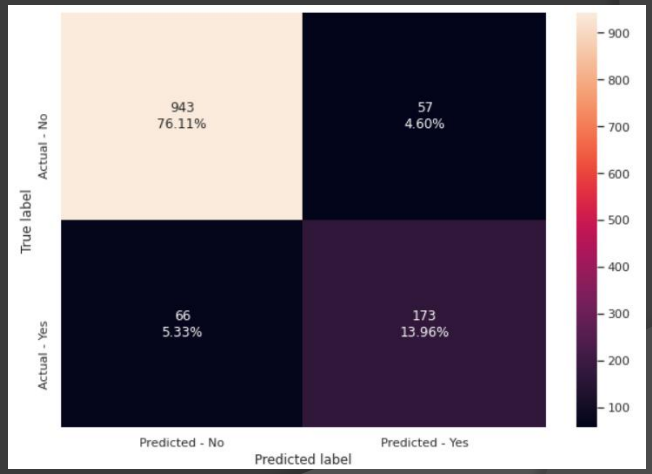
Post Data Preprocessing,

1. We selected Product Taken as Dependent(Predicted) Feature and majority of Numerical features as Independent Features(Predictors)

2. Next, we split the dataset into 70% to Train on and 30% to Test the Model on subsequently and used Stratify feature during the Test-Train split to preserve the proportion of target as in the original dataset.

3. Next, we fit the Decision Tree model on the Train and got following model metrics and Train and test confusion matrix

```
Accuracy on train data: 1.0
Accuracy on test data: 0.9007263922518159
Recall on train data: 1.0
Recall on test data: 0.7238493723849372
Precision on train data: 1.0
Precision on test data: 0.7521739130434782
f1 score on train data: 1.0
f1 score on test data: 0.7377398720682303
```



4. In this case, not being able to identify a potential customer for travel package is the biggest loss we can face. Hence, recall is the right metric to check the performance of the model.

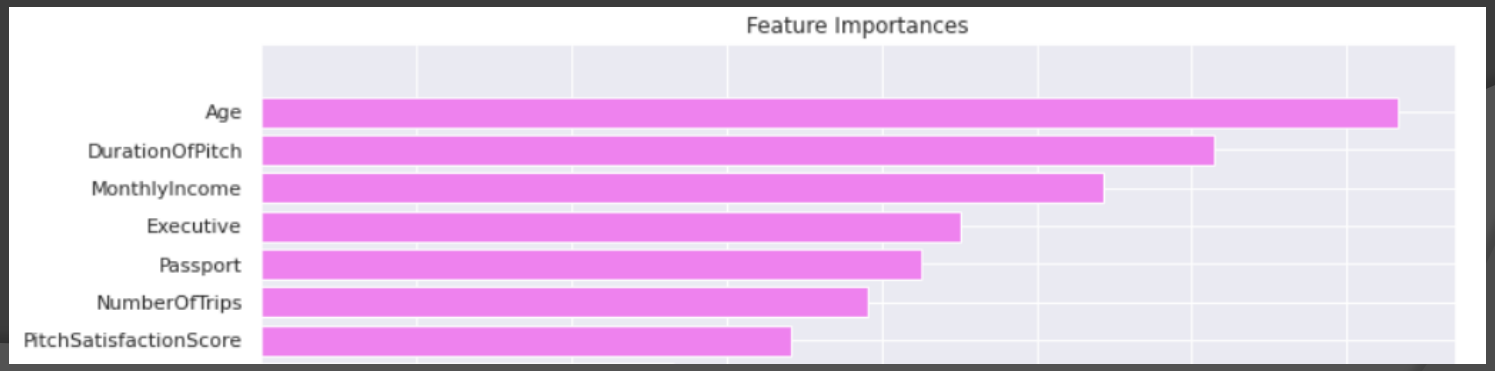
ML Model Building – Tuned Decision tree

1. We tuned the Decision Tree model with following hyper parameters and executed a Gridsearch to find the best estimator

```
parameters = {'max_depth': list(np.arange(2,20)) + [None],
              'min_samples_leaf': [1, 3, 5, 7, 10],
              'max_leaf_nodes' : [2, 3, 5, 10, 15] + [None],
              'min_impurity_decrease': [0.001, 0.01, 0.1, 0.0]
             }
```

2. Resulted in the following metric and Feature importance for the best estimator. Age is the most important feature, in addition to DurationofPitch and MonthlyIncome

```
Accuracy on train data: 0.9785392869505019
Accuracy on test data: 0.8870056497175142
Recall on train data: 0.8942652329749103
Recall on test data: 0.6485355648535565
Precision on train data: 0.9940239043824701
Precision on test data: 0.7345971563981043
f1 score on train data: 0.9415094339622642
f1 score on test data: 0.6888888888888889
```



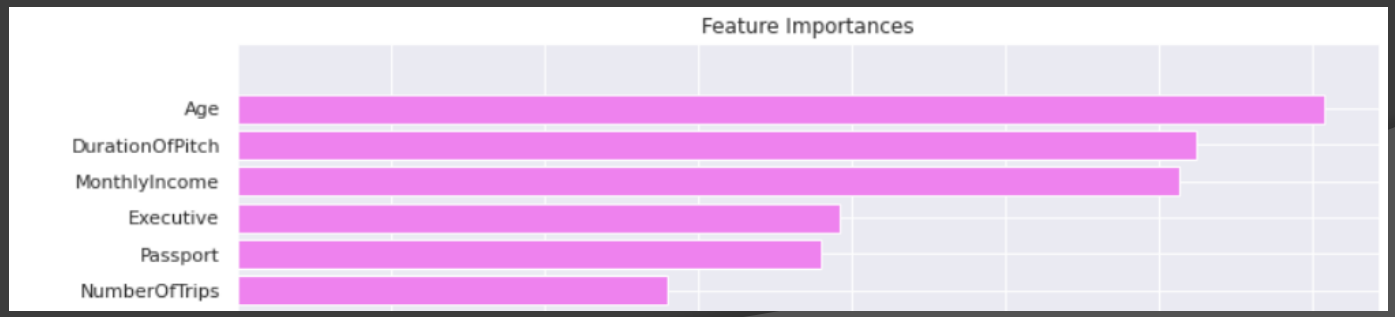
ML Model Building – Random Forest

1. First we ran a baseline RF model and then tuned with following hyper parameters and executed a Gridsearch to find the best estimator

```
parameters = {
                'max_depth':[4, 6, 8, 10, None],
                'max_features': ['sqrt','log2',None],
                'n_estimators': [80, 90, 100, 110, 120]
}
```

2. Resulted in the following metric and Feature importance for the best estimator. Age is the most important feature, in addition to DurationofPitch and MonthlyIncome(Same as DT)

```
Accuracy on train data: 1.0
Accuracy on test data: 0.9225181598062954
Recall on train data: 1.0
Recall on test data: 0.7071129707112971
Precision on train data: 1.0
Precision on test data: 0.8666666666666667
f1 score on train data: 1.0
f1 score on test data: 0.7788018433179724
```



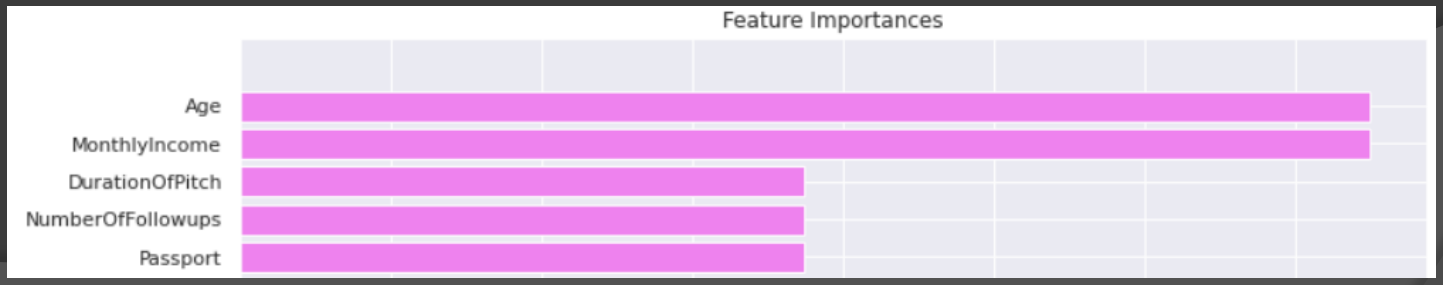
ML Model Building – AdaBoost Classifier

1. For boosting models, first we ran a baseline AdaBoost model and then tuned with following hyper parameters and executed a Gridsearch to find the best estimator

```
parameters = {'n_estimators': np.arange(10,100,10),
              'learning_rate': [1, 0.1, 0.5, 0.01],
              }
```

2. Resulted in the following metric and Feature importance for the best estimator. AdaBoost performed particularly poorly for recall and F1 metric for baseline and tuned models.

```
Accuracy on train data: 0.8584285219799238
Accuracy on test data: 0.8288942695722357
Recall on train data: 0.3727598566308244
Recall on test data: 0.2803347280334728
Precision on train data: 0.7790262172284644
Precision on test data: 0.6261682242990654
f1 score on train data: 0.5042424242424243
f1 score on test data: 0.3872832369942196
```



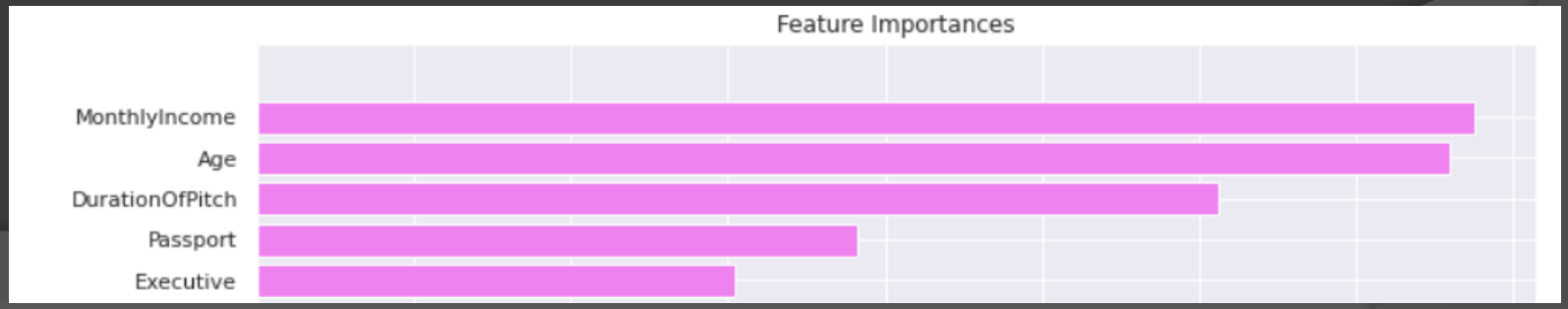
ML Model Building – GradientBoost Classifier

1. Next we ran a baseline GradientBoost model and then tuned with following hyper parameters and executed a Gridsearch to find the best estimator

```
parameters = {'n_estimators': np.arange(50,200,25),
              'subsample':[0.7,0.8,0.9,1],
              'max_features':[0.7,0.8,0.9,1],
              'max_depth':[3,5,7,10]
             }
```

2. Resulted in the following metric and Feature importance for the best estimator. For tuned GBmodel, Recall and F1 metric are performing as well as the Decision Tree and Random Forest models. Although the top5 features here are the same as teh Decision tree and Random Forest classifiers, Monthly Income is higher importance than Age.

```
Accuracy on train data: 1.0
Accuracy on test data: 0.9297820823244553
Recall on train data: 1.0
Recall on test data: 0.7238493723849372
Precision on train data: 1.0
Precision on test data: 0.8917525773195877
f1 score on train data: 1.0
f1 score on test data: 0.7990762124711316
```



ML Model Building – XGBoost Classifier

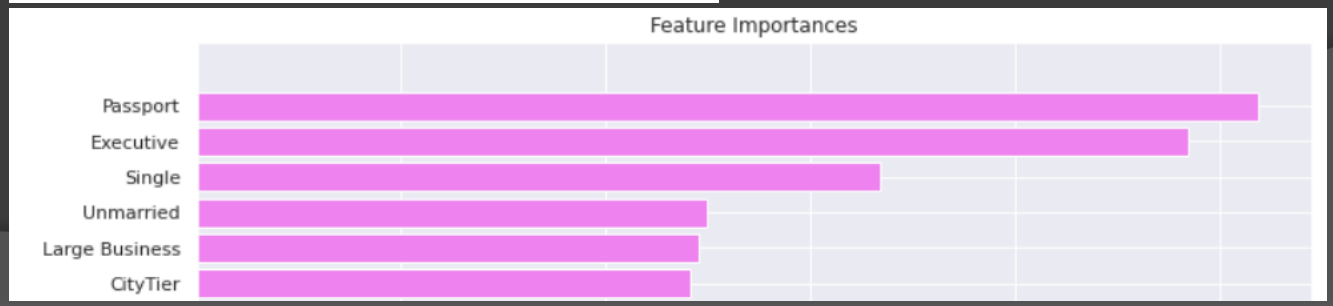
1. For next Boosting models, we used Google's Hardware accelerator (GPU) on Google Colab to run these models as local RAM wasn't able to keep up with the intensive computation.

   Next we ran a baseline XGBoost model and then tuned with following hyper parameters and executed a Gridsearch to find the best estimator

```
parameters = {'n_estimators': [75,100,125,150],
              'subsample':[0.7, 0.8, 0.9, 1],
              'gamma':[0, 1, 3, 5],
              'colsample_bytree':[0.7, 0.8, 0.9, 1],
              'colsample_bylevel':[0.7, 0.8, 0.9, 1]
              }
```

2. Resulted in the following metric and Feature importance for the best estimator. For tuned XGBoost the modle performing badly for Recall metric. Therefore, the top 5 features are also quite different from Decision tree and Random Forest classifiers.

```
Accuracy on train data: 0.9093111803392178
Accuracy on test data: 0.8684422921711057
Recall on train data: 0.5931899641577061
Recall on test data: 0.4435146443514644
Precision on train data: 0.9043715846994536
Precision on test data: 0.7794117647058824
f1 score on train data: 0.7164502164502164
f1 score on test data: 0.5653333333333332
```



ML Model Building – Stacking Model

1. Next we ran a Stacking model with tuned models of decision tree, random forest and gradient boosting, then use XGBoost to get the final prediction.

2. Resulted in the following metric. Stacking Classifer score highly on Recall and F1 scores

```
Accuracy on train data: 1.0
Accuracy on test data: 0.930589184826473
Recall on train data: 1.0
Recall on test data: 0.7615062761506276
Precision on train data: 1.0
Precision on test data: 0.8625592417061612
f1 score on train data: 1.0
f1 score on test data: 0.8088888888888888
```

Model Comparison

Following table illustrates and compares all the results of all the Baseline and Hypertuned Models

	Model	Accuracy_train	Accuracy_test	Recall_train	Recall_test	Precision_train	Precision_test	f1_score_train	f1_score_test
0	Decision Tree	1.000000	0.900726	1.000000	0.723849	1.000000	0.752174	1.000000	0.737740
1	Tuned Decision Tree	0.978539	0.887006	0.894265	0.648536	0.994024	0.734597	0.941509	0.688889
2	Random Forest	1.000000	0.913640	1.000000	0.615063	1.000000	0.907407	1.000000	0.733167
3	Tuned Random Forest	1.000000	0.922518	1.000000	0.707113	1.000000	0.866667	1.000000	0.778802
4	AdaBoost Classifier	0.849775	0.822437	0.394265	0.317992	0.696203	0.571429	0.503432	0.408602
5	Tuned AdaBoost Classifier	0.858429	0.828894	0.372760	0.280335	0.779026	0.626168	0.504242	0.387283
6	Gradient Boosting Classifier	0.901696	0.858757	0.560932	0.422594	0.889205	0.731884	0.687912	0.535809
7	Tuned Gradient Boosting Classifier	1.000000	0.929782	1.000000	0.723849	1.000000	0.891753	1.000000	0.799076
8	XGBoost Classifier	0.895812	0.857950	0.530466	0.410042	0.883582	0.736842	0.662934	0.526882
9	Tuned XGBoost Classifier	0.909311	0.868442	0.593190	0.443515	0.904372	0.779412	0.716450	0.565333
10	Stacking Classifier	1.000000	0.930589	1.000000	0.761506	1.000000	0.862559	1.000000	0.808889

1. Tuned random forest model, Tuned Gradient Boosting and Stacking classifers are the best models here. They have highest Recall and F1 scores

2. Other boosting classifiers like AdaBoost and XGboost classifers perfomred poorly on this dataset

Actionable Insights & Recommendations

Insights -

1. We analyzed the “Travel Package Dataset" using EDA and used several Ensemble Baseline and Boosting ML classifier and compared results
2. The best models built can be used to predict if a customer is going to sign up for a travel package.
3. We compared the recall metrics of different ML models as outlined in the comparison table. Decision tree with pruning had the best metric.
4. Overall we can see that the Tuned Decision tree And Random Forest models performs better on a given dataset
5. -For Decision tree And Random Forest models, Feature importance also matches. The top 5 features include Age, Duration of Pitch, Monthly Income, Executive position and holding Passport are important features that has a positive predictive relation with a customer taking the travel package
6. We established the importance of hyper-parameters/ pruning to reduce overfitting

Recommendations -

1. We can use this predictive model with top5 important features viz. Age, Duration of Pitch, Monthly Income, Executive position and holding Passport to predict if the customer will select a travel package.
2. The demographic age of 30-45 that are in executive positions and holding passports can be the focus customer base. This group will need to be given a effective pitch
3. Self Inquiry is much higher than Company Invited. Needs to be investigated for better marketing strategy.
4. ML model with more data to predict the perfect time to make the pitch will help us reach customers before they reach us.
5. Couples and families are large demographic. Targeted advertisement to this group will help.
6. Deluxe and Basic packages are popular packages. We can investigate these further to know what makes them popular to offer packages in future.