App Subscription - Binomial Classification

1. **Problem Statement:**

A financial company developed the app and released the free version to the customers . Recorded the Behaviour of the customer on various independent features like date , time, number of screens, time , features liked etc and Target variable of Enrolled / Not Enrolled.

Challenge: Need to Develop and train the classification model and on testing with the new data , it need to predict whether the customer will Enrolled / Not Enrolled with good accuracy .

1. **Import Data & Review**

* **Data – 5000 instances \* 12 features/ columns**
* **Missing values in enrolled\_date**
* **first\_open & enrolled\_date is in object format - So need to convert them into date format**
* **hour also in object format**

1. **EDA**

**Observations**

* Mean age of the users – 31
* 10.7% users played the minigame
* 17.2% users used the preminum featues
* **62.1% users were enrolled**
* 16.5% users liked
* Hour need to convert from object to int & 02:00:00 format to 2

**Correlation Data Analysis**

* dayofweek', 'numscreens', 'minigame' is positive correlated with enrolled
* 'hour', 'age', 'used\_premium\_feature', 'liked' is negatively correlated with enrolled.
* 'dayofweek' - users used long hours in later part of the week and of them are youngster
* 'hour' - less numb od screen used at late hours
* 'age' - more younsters played the minigame & yougnsters used more screens
* 'numscreens' - more screens used more played minigame & used premium features
* 'minigame'- more played minigame then more used premium features

1. **Feature Engineering**

* convert first\_open & enrolled\_date into date format
* Created a new columns diff\_hours
* diff\_hours = (enrolled\_date - first\_open) .astype('timedelta64[h]')
* value\_counts of diff\_hours shows that
  + Enrolments in first 1 hr is 67.21 %
  + Enrolments between 1 to 48hrs is 12.76 %
  + Enrolments between 48hrs to 5434 hrs is 20.03 %
  + As 80% of the enrolments happen in less than 2 days( 48hrs). So we are considering only these enrolments
  + Remove the Enrolment if enrolled more than 48hrs
  + Process the screen\_list – Create columns for top screen items and count them in Screen\_list
  + Group all the similar columns.

1. **Machine Learning model**

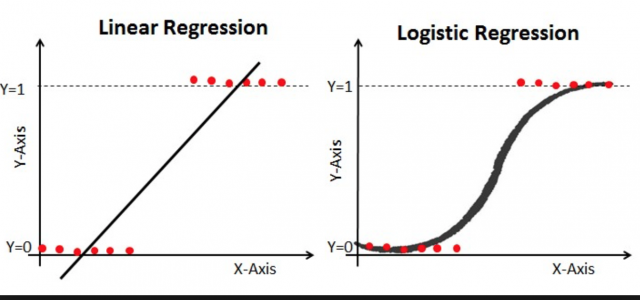
* **Split the train\_test with stratify with target**
* check distribution of target in both train & test
* Standard scaling of all the independent columns. As StandardScaler looses index & column names we saving the results in other df and later pass it back to X\_train & X\_test
* **Model Building** : LogisticRegression
* **Model Evaluation** : confusion\_matrix, classification\_report, accuracy\_score, f1\_score, precision\_score, recall\_score

**Logistic Regression:**

 Logistic Regression is used when the dependent variable(target) is categorical.

For example,

* To predict whether an email is spam (1) or (0)
* Whether the tumor is malignant (1) or not (0)



* **Accuracy of the raw model with cross validation is 76% with St. Deviation +/- 0.007**

1. **Further Improving the Model - Parameter tuning by Grid Search**

* **Important parameters of Logistic Regression is are C, and penalty [ L1, L1]**

**C: Penalty parameter C of the error term. It also controls the trade off between smooth decision boundary and classifying the training points correctly.**

**Penalty : ['l1', 'l2']**

**L1 regularization - *Lasso Regression***

**L2 regularization -  *Ridge Regression*.**

**The key difference between these techniques is that Lasso shrinks the less important feature’s coefficient to zero thus, removing some feature altogether. So, this works well for feature selection in case we have a huge number of features.**

* **Results : With the Parameter Tuning the accuracy of the model was futher increased to 77%.**