**Introduction**

**For this churn prediction problem statement where we have to predict probable customers who may churn in coming future.**

**We will reach to the final solution after following each phase of machine learning life cycle.**

**Since our focus is on those customers who may churn in future so we will be focusing more on those customers and try to tune our model to increase recall value where model should predict a smaller number of false negatives. We will try to minimise false negatives and increase True positives. To tack this metric we will use” recall” as our primary metric.**

**We will be focusing on recall metric as accuracy score will be not that much relevant here. Suppose if we have predicted churn for few customers who have not, In this case there is not much impact on business but if we have predicted not churn for those who will leave then that will impact too much on the business that’s why will be focusing more on false negatives.**

**Note: We have provided appropriate comment in notebook to understand each steps**

**Perform exploratory data analysis.**

**In our first phase of ML life cycle we will try to see insights about data**

* **Total number of records**
* **Total number of columns**
* **If data is imbalanced or not(3:1)**
* **If there is any correlation between any features**
* **How many categorical features and numerical features we have**
* **Descr5 point summary of all features to check if there is any relation or not**
* **Plot charts if necessary, to find relation between features**
* **Check if there is any null/empty value in any features**
* **Check for incorrect format/ corrupt data**

**Preprocess the data**

**In this phase of ML cycle, we will pre-process our data and remove/impute incorrect/corrupt features.**

* **Impute the features where there is NULL values**
* **Remove tuples if necessary**
* **Convert categorical features into numerical feature using labelencoder**
* **Check for nominal and ordinal**
* **If feature is nominal, using onehot encoding or dummy\_columns transform those features into onehot encoded feature.**
* **Check if all features are numerical or not**
* **Separate dependent feature and independent feature**
* **Split train data into tarain and validation set**

**Since dataset in imbalanced we will use different sampling method to balance the data**

* **Random Sampling**
* **SMOTE**
* **Adasyn**

**Select the right model.**

**In this phase of ML cycle, we will define our baseline model and try different bagging, boosting, LR algorithm.**

* **Define baseline model**
* **Train model using different over sampled data**
* **Select best model out of all created models**

**Train the model**

**Once bast model is selected, tune hyperparameters of that model using gridsearch. Once trained, we can validate our model to check bias and variance of our model**

**Test the model (Predictions and reporting)**

**We can use this model for getting prediction from it.**

**Evaluate the model performance**

**We will evaluate our model using different metrics. Our main focus will be on recall value and in our training phase we have tried to increase this value. If it has crossed the baseline value and we have accepted this model then we can save and our this model for prediction in our applications.**

**Our base line model had recall value of 0.56 and after resampling and hypertuning we are getting recall value of 0.84.**

**Suggest ways of improving the model**

**Below are the ways using which we can improve the performance of our model.**

* **Since no threshold has been provided to accept/reject model, we have trained our model for only one algorithm(Logistic regression). We can check for tree based and bossing algorithms for it as well. For eg: RandomForestClassifier, DecisionTreeClassifie, XgboostClassifier, CatboostClassifier**
* **Since we have not much data, if possible, we can try to collect more data to get more accurate predictions**
* **We can drill down more on available features and can work on feature engineering to create some helpful feature.**
* **We can use feature cross to combine more than one feature to get better nonlinear boundaries on prediction (keeping in mind that we are not overfitting the model)**
* **We can try more available algorithms and tune those get better prediction.**

**-----------------------------------------------------------THANKS----------------------------------------------------**