PROJECT REPORT

@Name : Subham Sarkar

@Github link : https://github.com/SubhamIO

Title : Delinquency Telecom Model

Definition:

Delinquency is a condition that arises when an activity or situation does not occur at its scheduled (or expected) date i.e., it occurs later than expected.

Use Case:

Many donors, experts, and microfinance institutions (MFI) have become convinced that using mobile financial services (MFS) is more convenient and efficient, and less costly, than the traditional high-touch model for delivering microfinance services. MFS becomes especially useful when targeting the unbanked poor living in remote areas. The implementation of MFS, though, has been uneven with both significant challenges and successes.

Today, microfinance is widely accepted as a poverty-reduction tool, representing $70 billion in outstanding loans and a global outreach of 200 million clients.

One of our Client in Telecom collaborates with an MFI to provide micro-credit on mobile balances to be paid back in 5 days. The Consumer is believed to be delinquent if he deviates from the path of paying back the loaned amount within 5 days

Machine Learning problem :

Create a delinquency model which can predict in terms of a probability for each loan transaction, whether the customer will be paying back the loaned amount within 5 days of insurance of loan (Label ‘1’ & ’0’)
Basically a Binary Classification setup .

Real-world/Business objectives and constraints.

No low-latency requirement.

Interpretability is important.

Probability of a data-point belonging to each class is needed.

Performance Metric

- **Log-loss (Since probabilities is our concern)**

- Confusion matrix (Also want to check some precision and recalls)

Approach :

1. **Exploratory Data Analysis**
2. **Checking Missing values,NaN, Duplicates etc.**
3. **Checking Data Imbalances**
4. **Checking Correlations among features**
5. **Preprocessing the data**
6. **Train-Test Split**
7. **Random Model Design for comparing it's LogLoss with the ML models developed later on the dataset.**
8. **MODELS USED and their results :**

Model Name	Train LogLoss	CV LogLoss	Test LogLoss	% Misclassified Points
Logistic Regression With Class balancing	0.298	0.297	0.302	0.122
Linear SVM	0.309	0.308	0.312	0.122
Random Forest Classifier	0.265	0.266	0.272	0.095
Logistic Regression With Class balancing(UPSAMPLING)	0.522	0.521	0.522	0.247

9. **Also we can see the probabilities for both classes, since probabilistic interpretation was needed.**
This is done for all models. Please do check in the ipynb file. For example :

```
Predicted Class : 1
Predicted Class Probabilities: [[0.3022 0.6978]]
Actual Class : [[1]]
```

10.OBSERVATION :

- **All the models performed better than the random model, which makes sense.**

- **From the pretty table we can see that , RandomForest performed best here.**

- **Even the overfitting is not present if we check the train and test logloss, they are very close**

- **Over sampling method was also applied on the training data to make the data more balanced, but it gave worse results .**