Telecom User Churn Prediction

Jobis James, SID: 10902871, jamesj29@uni.coventry.ac.uk

***Abstract* - Customer turnover is a main problem for telecommunication corporations. Because of the impact on company profits, particularly in the telecommunications industry, organisations are attempting to create methods to forecast possible customer churn. As a result, identifying the variables that contribute to customer turnover is critical to take the required steps to decrease churn. My major contribution is the development of a churn prediction model that aids telecom providers in predicting consumers who are likely to churn. In this paper, I applied different pre-processing techniques, sampling methods and machine learning techniques like Random Forest, Logistic Regression, Light GBM Classifier and Extreme Gradient Boosting (XGB) Classifier to predict the customer churn. To evaluate the performance of machine learning techniques, I have considered the measures like recall, AUC (Area under curve) Score and F1 score.**

Keywords: Logistic Regression, Random Forest, Light GBM Classifier and eXtreme Gradient Boosting (XGB) Classifier.

I. INTRODUCTION

The telecommunications industry has grown to be one of the most important sectors in the world. The degree of competition has risen because of technological advancements and an increase in the number of operators. The objective of every firm is to have as many customers as possible. It is critical not just to strive to acquire new customers, but also to keep existing ones. Retaining a customer is less expensive than acquiring a new one. Furthermore, a new customer may be uninterested in business services, making it harder to deal with him, but previous clients already have the essential data on service interaction.

Companies are working hard to stay afloat in this modern world, employing a variety of methods. One of the main strategies is to hold the existing customers. This strategy is the best profitable strategy in terms of return on investment. To implement this approach, organizations must reduce the possibility of customer churn, which means the customer migration from one telecom company to another [1].

Telecom companies can respond in time and try to keep the client who wants to depart by forecasting the churn value of users, who are about to leave the subscription. Companies can provide unique offers depends on the services used by the customers who are about to leave. In that way, companies can attempt to persuade the customers to reconsider leaving the operator. This will make the work of retaining users easier to accomplish than the effort of recruiting new users.

The dataset got from Kaggle has some problems, one of which was an imbalance problem, in which the class of churn consumers was relatively small in comparison with the class of active customers. To address the imbalance issue, we tried to train the models using oversampled, under sampled and imbalanced dataset itself. For this course work, I have used machine learning methods like Logistic regression, Random forest, LGBM and XGB classifiers to predict the customer's churn. The AUC was utilised in the assessment since it is general and is utilised in the event of imbalanced datasets.

In this course work, Random Forest Classifier and Logistic regression methods are the best among all other models. To select the best model, I have used performance metrics recall, AUC score, and f1 score.

Makhtar et al. [3] presented a methodology for churn prediction in telecom using rough set theory. According to this study, Rough Set classification algorithm beat standard ML methods.

II. DATASET

For this course work, we took the dataset from Kaggle [2].

Attributes	Description
CustomerID	Customer Id
gender	Client Gender (Male / Female)
SeniorCitizen	Is the Client Retired (1, 0)
Partner	Is the Client Married (Yes, No)
tenure	How Many Months A Person Has Been A Client Of The Company
PhoneService	Is the Telephone Service Connected (Yes, No)
MultipleLines	Are Multiple Phone Lines Connected (Yes, No, No Phone Service)
InternetService	Client's Internet Service Provider (Dsl, Fiber Optic, No)
OnlineSecurity	Is the Online Security Service Connected (Yes, No, No Internet Service)
OnlineBackup	Is the Online Backup Service Activated (Yes, No, No Internet Service)
DeviceProtection	Does the Client Have Equipment Insurance (Yes, No, No Internet Service)
TechSupport	Is the Technical Support Service Connected (Yes, No, No Internet Service)
StreamingTV	Is the Streaming Tv Service Connected (Yes, No, No Internet Service)
StreamingMovies	Is the Streaming Cinema Service Activated (Yes, No, No Internet Service)
Contract	Type of Customer Contract (Month-To-Month, One Year, Two Year)
PaperlessBilling	Whether the Client Uses Paperless Billing (Yes, No)
PaymentMethod	Payment Method (Electronic Check, Mailed Check, Bank Transfer (Automatic), Credit Card (Automatic))
MonthlyCharges	Current Monthly Payment
TotalCharges	The Total Amount That the Client Paid for The Services For The Entire Time
Churn	Whether There Was A Churn (Yes or No)

Figure 1 Dataset Attributes

This dataset contains data of users of a telecommunication company. Our dataset contains 5986 user data with 22 features such as demographic characteristics, services used, length of usage of the operator's services, payment method, and payment amount. The attributes of our dataset are listed in Figure 1. Our dataset is imbalanced because it has only 26.5 % churned user's data.

III. EXPLORATORY DATA ANALYSIS

1. Relationship between Attributes

Figure 2 illustrates the correlation graph of each variables with attribute Churn. Contract and tenure attributes have highest positive correlation with customer churn.

```
corrMatrix = users.corr().abs()['Churn'].sort_values(ascending=False)
corrMatrix

Churn               1.000000
Contract            0.395848
tenure              0.350420
OnlineSecurity      0.289879
TechSupport         0.277036
OnlineBackup        0.193675
PaperlessBilling    0.188653
MonthlyCharges      0.186122
DeviceProtection    0.179921
Dependents          0.159025
SeniorCitizen       0.149726
Partner             0.146378
PaymentMethod       0.104698
InternetService     0.046839
StreamingMovies     0.039352
StreamingTV         0.037394
MultipleLines       0.034820
gender              0.009377
PhoneService        0.009097
Name: Churn, dtype: float64
```

Figure 2 Correlation values of Attributes with Churn

2. Insights from EDA

Figure 3 shows that customers who are just beginning to utilise a telecoms company's services are more likely to depart. If the monthly payment is low, then the customer will stay most probably. On the other hand, the higher the monthly charges, the more likely the customer will depart.

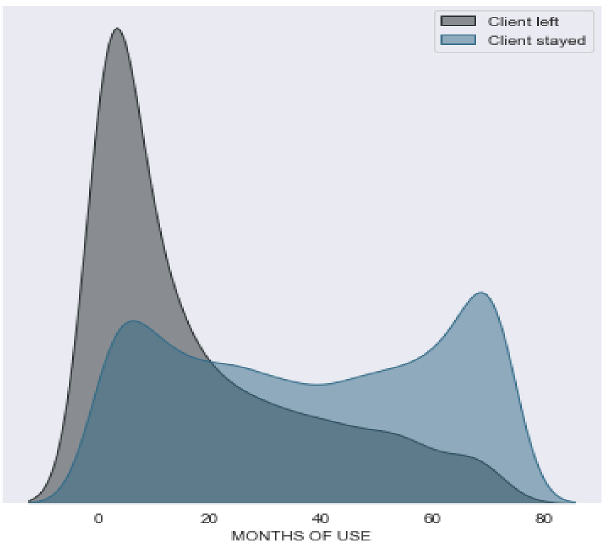

Figure 3 Distribution of Churn with Months of use

Pensioners and unmarried people are more likely to depart. The firm has issues with fibre optic Internet. There is a lack of internet security, backup, device protection, and technical assistance. Clients that quit the firm nearly always have a month-to-month payment; customers who have a one- or two-year commitment practically never leave the company. Customers who pay for services using an electronic check are more likely to leave the company.

IV. METHODOLOGY

In this paper, we will be predicting the values for customer churn using four different machine learning algorithms such as

1. Logistic Regression
2. Random Forest
3. Light GBM Classifier
4. eXtreme Gradient Boost Classifier (XGB)

(I) Logistic Regression

Logistic regression is a linear approach; however, the logistic function is used to modify the predictions. The core of the logistic regression is the logistic function. To forecast an output value, input values are linearly blended using weights or coefficient values (Beta). The output of the model is a binary value (0 or 1) rather than a numeric number, which distinguishes it from linear regression.

(II) Random Forest

Random forest creates multiple decision trees on different samples of the dataset and combines them together to give a stable prediction. The sub-sample size can be controlled with the max_sample parameter. This algorithm introduces randomness to the model while growing. This method will search for the important feature while training. This is one of the reasons for the better prediction.

(III) Light GBM Classifier (LGBM)

Light GBM is a gradient boosting framework based on decision tree techniques [4]. LGBM can handle huge amounts of data while using less memory. And it gives more importance to the correctness of results. LGBM also enables GPU learning, so it is very useful for creating data science applications. The growth of the tree is vertical in LGBM, which means LGBM will grow in leaf-wise manner. The selection of leaf is based on the greatest delta loss.

(IV) eXtreme Gradient Boost Classifier (XGB)

XGBoost is an ensemble Machine Learning method based on decision trees that use a gradient boosting framework. Ensemble learning provides a methodical approach to combining the predictive potential of several learners. The end result is a single model that aggregates the output of multiple models. When adding new models, it employs a gradient descent technique to minimise loss. This method is applicable to both regression and classification predictive modelling problems.

V. EXPERIMENTAL SETUP

A. Data Pre-processing

Data pre-processing is a data mining technique for cleansing and making accessible raw data acquired from diverse sources. Missing data, null values, noisy data, and untrustworthy data are all eliminated when data is prepared. In this situation, the following pre-processing procedures are used:

1. Handling missing and null values

Real-world data has a large number of missing values. If the null or missing value attribute has a poor relationship with class prediction, the problem can be handled by removing the null or missing value row or column. Missing or empty data can be replaced with zero or another integer, such as mean, min-max, or so on.

There are no missing values in our dataset at a glance. But after analysing the dataset more deeply, I have found 10 empty strings in TotalCharges attribute of the dataset. The corresponding ten rows have tenure value as zero. TotalCharges is the product of tenure and MonthlyCharges. That means, attribute Totalcharges has a linear relationship between tenure and MonthlyCharges. Therefore, I have deleted the column TotalCharges which contains empty strings.

2. Convert categorical values to numerical values

There are 15 categorical attributes. So, I have tried label encoder and one hot encoding methods to convert the data type. And then tried to evaluate the model. The training time, AUC score and recall is better for the dataset which used label encoder method. Therefore, I have proceeded with the label encoding method to convert the categorical values to numerical values.

3. Standardisation

Standardisation is the process of making the mean and standard deviation of attributes into 0 and 1 respectively. Variables assessed at various scales may not contribute equally to the model fitting and learnt function and may result in bias.

There are 2 numerical columns – tenure and monthly charges. I standardised the columns to create a better model. I have tried to predict the churn using standardised and non-standardised dataset. But Standardised dataset gives better prediction of all models.

4. Feature Selection

The process of identifying appropriate characteristics for data modelling. Feature selection improves the performance of our model. I removed unwanted columns like Unnamed:0, customerID, TotalCharges, gender and PhoneService. Unnamed:0 and CustomerId attributes are removed because it does not contain any data that is useful for prediction. As we discussed in EDA, Totalcharges attribute is linearly dependent on tenure and MonthlyCharge. As in Figure 1, attributes gender and PhoneService have the least correlation value of 0.009 with the dependent variable Churn. Therefore, I removed those 2 columns from the dataset.

B. Imbalanced data handling

Standard machine learning methods have a tendency to learn and predict the major classes. There is a chance for neglecting the minority classes. In terms of model performance evaluation, if our dataset has an unbalanced data distribution, then there is a chance for low recall.

The two main algorithms used to balance the dataset are SMOTE and Near Miss.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5986 entries, 0 to 5985
Data columns (total 17 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   SeniorCitizen     5986 non-null   int64
 1   Partner           5986 non-null   int32
 2   Dependents        5986 non-null   int32
 3   tenure            5986 non-null   int64
 4   MultipleLines     5986 non-null   int32
 5   InternetService   5986 non-null   int32
 6   OnlineSecurity    5986 non-null   int32
 7   OnlineBackup      5986 non-null   int32
 8   DeviceProtection  5986 non-null   int32
 9   TechSupport       5986 non-null   int32
 10  StreamingTV       5986 non-null   int32
 11  StreamingMovies   5986 non-null   int32
 12  Contract          5986 non-null   int32
 13  PaperlessBilling  5986 non-null   int32
 14  PaymentMethod     5986 non-null   int32
 15  MonthlyCharges    5986 non-null   float64
 16  Churn             5986 non-null   int64
dtypes: float64(1), int32(13), int64(3)
memory usage: 491.2 KB
```

Figure 4 Dataset Structure after Data Pre-Processing

SMOTE (Synthetic Minority Oversampling Technique) is a oversampling technique for resolving the imbalanced class distribution problem. It creates random instances of minority class. By using linear interpolation, SMOTE will create random instances of minority class.

NearMiss is a under sampling technique to balance the class distribution in the dataset. This technique will remove majority class instances in a random way. This facilitates the categorization process.

There are 16 features including predictor Churn after data cleaning and pre-processing. I divided the dataset into training and testing datasets in a ration of 80:20. Our dataset is an imbalanced dataset with 26.5%

of churned data. So, I have created an oversampled and under sampled dataset of 80% training data using SMOTE and NearMiss algorithm. Then, I have used the imbalanced dataset, oversampled dataset and under sampled dataset to train the different models.

C. Implementation of Methods

I have implemented the ML models with the help of python and sklearn library. First, I have trained ML algorithms like Logistic Regression, Random Forest, LGBM and XGB with default hyperparameter values on imbalanced, oversampled and under-sampled training dataset. Then I have used the trained models to predict the customer churn of the 20% testing dataset. After that, I have analysed the performance of models on the 20% testing dataset using evaluation metrics AUC score, recall and f1 score. I have also used the cross-validation technique to assess our machine learning model's competence on unseen data. For this course work, stratified cross-validation technique is used. In the technique, each fold contains same number of instances of each class. This technique is mainly used for imbalanced datasets.

The following is the general procedure of cross-validation:

- Randomly shuffle the dataset.
- Divide the data into k groups. `in our case k=5.
- For each distinct group:
  1. Consider one group as a validation data set.
  2. Consider the all other groups as a training data set.
  3. Fit the model with the training dataset and use this model to predict the output of testing dataset.
  4. Keep the assessment score.
- The average of the scores is the value of evaluation metrics.

After training different models, I choose four models which have good AUC score and Recall. All the four models are trained using training dataset and validate the model using testing dataset. The results of prediction are discussed in next section.

To reduce the error rates of prediction, I have performed hyperparameter tuning. Hyperparameter is a parameter that are the internal weights required for an algorithm to work. I have found the best hyperparameter values for each model to attain maximum recall. There are two methods for hyperparameter tuning – Grid Search and Random Search. I have used Random search method to tune hyperparameters of logistic regression and random forest. Grid search method is used for finding the best parameters of LGBR and XGB. Hyperparameter tuning is done for oversampled dataset. Because all models with default hyperparameters shows better performance on oversampled dataset.

Hyperparameter Tuning of Logistic Regression.

There are no critical hyperparameters to adjust in logistic regression. Then also, I have tuned the hyperparameters like solvers and C. C parameter controls the penalty value. The best recall score of 0.813 after 5-fold cross validation is achieved for hyperparameter solver : 'newton-cg' and C: 0.1.

Hyperparameter Tuning of Random Forest

There are mainly three important hyperparameters for Random Forest classifier- max_features, n_estimators, and min_sample_leaf. n_estimators hyperparameter specifies the number of trees the algorithm need to construct before doing voting. Max_features denoted the maximum number of features that random forest will evaluate when splitting a node. min_sample_leaf specifies the minimum count of leaves that is necessary to divide an internal node. The best recall score of 0.881 after 5-fold cross validation is achieved for n_estimators : 1000 , max_features= sqrt

Hyperparameter Tuning of LGBM

There are mainly three important hyperparameters for LGBM – num_leaves, max_depth. and min_data_in_leaf. num_leaves is the primary parameter for controlling the tree model's complexity. Overfitting will occur if the value is greater than this. min_data_in_leaf can control the growth of the tree. Setting it to a big number prevents the tree from growing too deep, but it may result in under-fitting. In reality, for a big dataset, setting it to hundreds or thousands is sufficient. max_depth can also be used to explicitly limit the tree depth. The best recall score of 0.864 after 5-fold cross validation is achieved for max_depth : 3 and num_leaves :5

```
Best: 0.846298 using {'max_depth': 3, 'num_leaves': 5}
0.843045 (0.011410) with: {'max_depth': 2, 'num_leaves': 2}
0.831733 (0.012604) with: {'max_depth': 2, 'num_leaves': 3}
0.844177 (0.012088) with: {'max_depth': 2, 'num_leaves': 5}
0.843045 (0.011410) with: {'max_depth': 3, 'num_leaves': 2}
0.831733 (0.012604) with: {'max_depth': 3, 'num_leaves': 3}
0.846298 (0.010448) with: {'max_depth': 3, 'num_leaves': 5}
0.843045 (0.011410) with: {'max_depth': 5, 'num_leaves': 2}
0.831733 (0.012604) with: {'max_depth': 5, 'num_leaves': 3}
0.844318 (0.011192) with: {'max_depth': 5, 'num_leaves': 5}
```

Figure 5 Best Recall and hyperparameter of LGBM

Hyperparameter Tuning of XGB

For our XGB model, we tuned hyperparameters like max_depth and min_child_weight. Max_depth is the maximum tree depth. Its default value is 6. It helps to control the overfitting. Min_child_weight defines the minimal weighted total of all required observations in a child. The best recall score of 0.858 after 5-fold cross validation is achieved for max_depth : 5 and min_child_weight : 3.

After hyperparameter tuning, I have used the models with tuned hyperparameters to train the data. The results will be discussed in next section

VI. RESULTS

We have successfully implemented the four ML classification models and predict the customer churn on the 20% testing dataset. We have used three different evaluation matrices to evaluate the models. The evaluation metrics used are ROC AUC Score, Recall and f1 score.

1. Area Under the Receiver Operating Characteristic Curve (ROC AUC) Score

The AUC reflects the degree or level of separability. It indicates how well the model can separate the classes. The best AUC score of a model is 1.

2. Recall

The recall is defined as the ratio of correctly predicted positive instances to all observations in the actual class. In the case of predict customer churn, the recall has more importance than precision. If our model fails to predict the customer who is about to leave, then there is no use for this model for telecom companies.

3. F1 score

The F1 Score is the weighted average of Precision and Recall. For imbalanced dataset, F1 score is more important than accuracy

A. Result Evaluation before hyperparameter tuning

The figure 6 contains the model name, type of dataset used for training, time takes to train the model, AUC, Recall, precision and f1 score of 20% test dataset, 5 fold cross-validation score of AUC, recall, precision, and f1 score of the training dataset.

I took four models with default hyperparameters which have high AUC, recall, and f1 score. The best four models are Logistic regression, random forest, LGBM, and XGB. Logistic regression has the lowest training time on imbalanced, oversampled, and under-sampled datasets. Random Forest classifier has taken the highest time to train the data.

Among the four models, Logistic regression is the best model without any hyperparameter tuning on the imbalanced, oversampled, and imbalanced training dataset. The logistic regression model has the highest AUC, recall, precision, and f1 score on the training and testing dataset. All the evaluation metrics of logistic regression on the 20% testing dataset are almost similar to the 5-fold cross-validation scores on the training dataset. Logistic regression trained with oversampled dataset has highest AUC, recall, and f1 score of 0.834, 0.81, and 0.63 respectively, on the testing dataset. The LGBM algorithm comes in next place, the XGB and the random forest came third and fourth regarding AUC values and recall. In Figure 5, it is very clear that all models trained with oversampled dataset has the best combination of AUC and recall scores. Therefore, I choose oversampled data for hyperparameter tuning.

In this paper, I have given more importance to AUC and recall. So, I have tried to find the best hyperparameters for the best recall score.

	Model	Data	Training Time	AUC Test	Recall Test	Precision Test	f1 Score test	AUC CV Train	Recall CV Train	Precision CV Train	f1 CV Train
0	Logistic Regression	Imbalanced data	0.043881	0.849	0.59	0.63	0.61	0.836	0.51	0.65	0.57
1	Logistic Regression	Over_Sampled_data	0.037868	0.834	0.81	0.52	0.63	0.859	0.81	0.76	0.79
2	Logistic Regression	Under_Sampled_data	0.021910	0.803	0.84	0.47	0.60	0.745	0.78	0.64	0.70
3	Random Forest	Imbalanced data	0.456777	0.819	0.54	0.62	0.58	0.815	0.47	0.62	0.53
4	Random Forest	Over_Sampled_data	0.586440	0.811	0.66	0.54	0.60	0.915	0.87	0.82	0.84
5	Random Forest	Under_Sampled_data	0.273749	0.647	0.80	0.35	0.49	0.736	0.70	0.66	0.68
6	LGBM	Imbalanced data	0.063791	0.839	0.58	0.62	0.60	0.828	0.51	0.62	0.56
7	LGBM	Over_Sampled_data	0.091831	0.832	0.74	0.54	0.63	0.907	0.86	0.80	0.83
8	LGBM	Under_Sampled_data	0.068042	0.611	0.79	0.35	0.48	0.754	0.70	0.67	0.69
9	XGB	Imbalanced data	0.323128	0.823	0.57	0.60	0.58	0.810	0.50	0.59	0.54
10	XGB	Over_Sampled_data	0.483098	0.815	0.68	0.55	0.61	0.913	0.85	0.81	0.83
11	XGB	Under_Sampled_data	0.208178	0.581	0.79	0.35	0.48	0.742	0.69	0.66	0.68

Figure 6 Performance evaluation of Models with default hyperparameter

	Model	Data	Training Time	AUC Test	Recall Test	Precision Test	f1 Score test	AUC CV Train	Recall CV Train	Precision CV Train	f1 CV Train
0	Logistic Regression	Oversampled Data	0.050864	0.834	0.81	0.52	0.63	0.859	0.81	0.76	0.79
1	Random Forest	Oversampled Data	4.973756	0.829	0.73	0.54	0.62	0.911	0.88	0.81	0.84
2	LGBM	Oversampled Data	0.059322	0.842	0.81	0.53	0.64	0.879	0.85	0.77	0.81
3	XGB	Oversampled Data	0.398728	0.828	0.70	0.55	0.61	0.909	0.86	0.81	0.83

Figure 7 Performance Evaluation of Models after hyperparameter tuning

B. Result Evaluation after hyperparameter tuning

From figure 7, it is obvious that Light GBM Classifier is the best model after hyperparameter tuning with AUC, recall, and f1 score of 0.842, 0.81, and 0.64 respectively on the testing dataset. The second-best model is logistic regression algorithm and the random forest and the XGB follows third and fourth in terms of AUC and the recall. Recall scores after 5-fold cross-validation on the training dataset have increased for all models. 5-fold Cross-validation score of recall on train dataset for models' random forest, LGBM, XGB, and logistic regression were 0.88, 0.85, 0.86, and 0.81 respectively. There is a significant increase in the recall for LGBM, XGB, and the random forest model on the 20% testing dataset. Recall of Logistic regression does not have a big change.

```
Model  LGBM
Data  Oversampled Data

ROC AUC Test set score: 0.842
Recall Test set score: 0.81
Precision Test set score: 0.53
f1 Score Test set : 0.64
-----------------------------------------
Cross-validation scores with 5 folds:
ROC AUC: 0.879
Recall: 0.85
Precision: 0.77
f1: 0.81
```

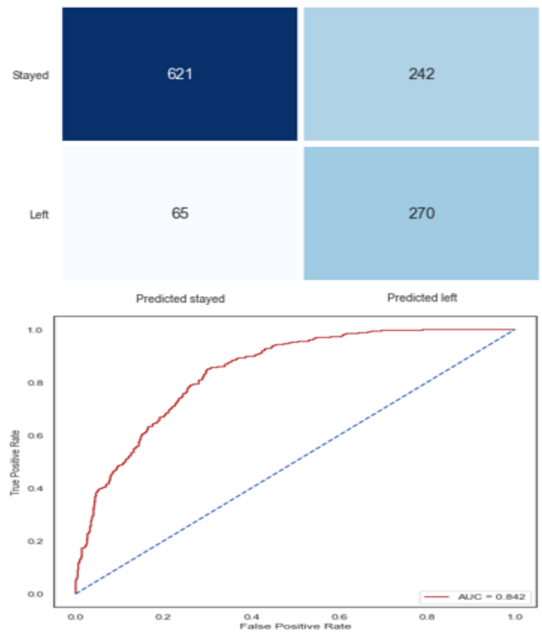

Figure 8 Evaluation metrics, confusion matrix and ROC curve of LGBM

LGBM and logistic regression models have only taken around 0.05 seconds for training. The training time of the random forest classifier is around 4.97 seconds which is very high when compared to all other three models.

```
              feature  importance
4              tenure         120
17     MonthlyCharges          88
14           Contract          34
16      PaymentMethod          32
8      OnlineSecurity          18
1       SeniorCitizen          16
0              gender          15
11        TechSupport          14
5        PhoneService          11
3          Dependents          11
9        OnlineBackup          11
2             Partner           9
6       MultipleLines           8
7     InternetService           4
15   PaperlessBilling           4
10   DeviceProtection           3
12        StreamingTV           1
13    StreamingMovies           1
```

Figure 9 Feature importance of Best Model LGBM

The important features in terms of LGBM is shown in Figure.9

VII. DISCUSSION AND CONCLUSION

The objective of this course work in the telecom sector is to help corporations in increasing their profits. It has come to light that anticipating churn is one of the most significant sources of revenue for telecom firms. As a result, the aim of this project was to develop a system that predicts customer churn. The dataset is divided into training and testing. For validation, we selected 5-fold cross-validation. Hyperparameter tuning is performed with the help of Grid search CV and Randomized search. To prepare the features for machine learning algorithms, data pre-processing and feature engineering are done.

To implement a solution for this problem, we have mainly considered four models - Logistic regression, random forest, LGBM, and XGB. Our dataset is imbalanced. So, I have created an oversampled and under-sampled dataset. Then I have built and trained the models with the imbalanced dataset, oversampled dataset, and under-sampled dataset. Oversampled data helps the models to predict more accurately. Among these models, Logistic regression and Light GBM are the best models with high AUC, recall, and f1 scores of around 0.84, 0.81, and 0.64 respectively on the testing dataset.

APPENDIX

1. Source code and Dataset: https://livecoventryac-my.sharepoint.com/:f:/g/personal/jamesj29_uni_coventry_ac_uk/Erk-LYicn8FAhKW4WGyXdUQBVw2IUmVAfihyrg6faSeHbw?e=LuHpwo

REFERENCE

1. Predicting Customer Churn https://scholar.google.com/scholar_lookup?title=Predicting%20customer%20churn%20in%20telecom%20industry%20using%20multilayer%20preceptron%20neural%20networks%3A%20modeling%20and%20analysis&journal=Igarss&volume=11&issue=1&pages=1-5&publication_year=2014&author=Bott%2C

2. Churn Classification Model https://scholar.google.com/scholar_lookup?title=Churn%20classification%20model%20for%20local%20telecommunication%20company%20based%20on%20rough%20set%20theory&journal=J%20Fundam%20Appl%20Sci&volume=9&issue=6&pages=854-868&publication_year=2017&author=Makhtar%2CM&author=Nafis%2CS&author=Mohamed%2CM&author=Awang%2CM&author=Rahman%2CM&author=Deris%2CM

3. Kaggle Dataset: Telecom users dataset | Kaggle

4. LGBM

   https://www.researchgate.net/figure/Step-wise-work-flow-for-the-purposed-LGBM-machine-learning-model_fig1_328140093