Customer Churn Prediction Using ML

SAI RAM DASARAPU (SDASARAP@GMU.EDU)
JASHWANTH RAJ GOWLIKAR (JGOWLIKA@GMU.EDU)

1 INTRODUCTION

Churn is usually defined as the loss of customers as a result of their switching from one service provider to another, and churn prediction is the process of identifying those customers at the correct time and help companies retain their customer base. Churn prediction has become more important and popular in the present world, where companies are investing a lot of funds trying to identify the churn customers.
Retaining existing customers is essential for any company as it is one of the key aspects of a companies growth and brand name. Also the process of retaining existing customers is much more easier and cost-effective than to add new customers. Customer churn has become a major problem for companies in the recent times, as the customer's are able to switch between the providers with ease and this in turn takes a hit on the profits of the company as they lose their potential revenues from losing the customer base. so, the organizations need a model to perfectly identify the customers who are likely to churn.
The relevance of churn prediction is majorly found in industries such as telecommunication, Insurance where the customers break the contracts, In banking where customers switch to there competitors can result in loss of deposits, streaming services lose there subscription base etc. Considering these consequences an organization must distinguish clients who are in danger of churn before they churn out, so that they can send proactive retention campaigns.
This project aims to accurately identify and predict the possible customers who are thinking of leaving the company services, and help the companies take measures to prevent them from churning by offering them incentives, better services, free subscriptions etc.

2 PROBLEM DEFINITION

In this project we were trying to solve the problem of customer churn prediction which is faced by the organizations across the domains, and which can affect both the organizations growth and finances.
The key difficultly in churn prediction is identifying the customers at the right time before they really churn.The availability of the data might also be an issue which might be sensitive to the organization. The data may have imbalance as the number of people who churn are relatively low as compared to not churn.The methodology we implemented aims to solve this problems by using different ML preprocessing techniques, and algorithms to predict which customers are more likely to churn.
In the scope of the project we implemented methodologies like stacking which is a type of ensemble which involves combining predictions of multiple base models and passing it as input for the meta model to improve performance and overall accuracy of the predictions, WOE weight of evidence to perform feature selection and transformation, and other Feature Selection process to help us predict better.The data involved collecting large amounts of data and preprocessing the data like handling the imbalance, dealing with the noise in the data like null values, dealing with categorical attributes by converting them into numerical ones using label encoding etc. and finally evaluating the performance of the models by using various metrics, validation, fp, tp rates to build accurate churn prediction model.
By addressing the problem of customer churn prediction, the methodology can help organizations to improve their customer retention's and reduce the negative impact of churn on their business.

3 METHODOLOGY

In the current project we have used the methodology called stacking also known as stacked generalization which is a type of ensemble learning method where it consists of base model and a meta model for prediction of data.

3.1 Stacking Ensemble

We have implemented a series of steps for the prediction of churn from preprocessing the data to getting the predicted outputs.

Data collection and preprocessing:
The first step involves collecting and reading the dataset which can be done using pandas, and performing preprocessing on the data which involves removing redundant features, handling missing values, removing outliers, converting the categorical type attributes to numerical, normalizing the data.

WOE Implementation:
In this step we are implementing the weight of evidence methodology and predictive power of each attribute is calculated which helps us understand if it has higher distribution of good or bad.

Feature Selection:
In the next step we selected the important features that helps in the prediction by selecting the top 60 percentile attributes by using the select percentile method.

Model Implementation:
In this step we have converted the data into train and test data with a split of (70,30). And we have implemented four base models decision tree, random forest,XgBoost and lightGBM.

Model Evaluation:
In this step the above models are evaluated using metrics such as accuracy, fpr,tpr. A 10 fold cross validation is performed on all the models.

Stacking Ensemble:
In the final step the above four models are passed on to the stacking ensemble base models for learning and the results of these models are passed onto the meta model which is by default logistic regression and the final prediction is made and compared using the auc curve and accuracy.

3.1.1 Stacking Ensemble pseudo code.

- Load the dataset using pandas.
- Preprocessing of data such as handling imbalance, outliers, missing values, label encoding, normalization.
- Implementing WOE methodology.
- Performing feature selection using select percentile method.
- Divide the dataset into train and test split (70,30).
- Implementing base models like decision tree, random forest, lightGBM, XGBoost with 10 fold cross validation.
- Perform model evaluation on all those models using metrics like accuracy, auc,fpr,tpr.
- Implement Stacking ensemble model by passing the above four models as inputs to the base model and logistic regression as meta model and predicting the outputs.
- Finding the best combinations of base models in stack ensemble for each dataset.
- plot the auc curve and compare the value of stacking w.r.t to the base models.

4 EXPERIMENTS

The dataset used for the implementation of Stacking ensemble is same as we used for the modeule 2 from the paper [1] consists of 100k instances with 100 attributes related to the customer telecom data with attributes types including int, float, string.

Link to the datset: https://www.kaggle.com/datasets/abhinav89/telecom-customer

We have also used another dataset for analysis and validation purposes it consists of 51k instances with 58 attributes related to customer churn in telecom sector.This dataset was randomly chosen from kaggle as it was a bit similar to the main dataset.

Link to the datset: https://www.kaggle.com/datasets/jpacse/datasets-for-churn-telecom?select=cell2celltrain.csv

we have implemented our code from scratch and have used the models implementations and evaluation metrics and some preprocessing techniques from sckitlearn libraries.

The implementation of the customer churn prediction using stack ensemble starts with reading in the two data sets containing the customer telecom data, with the first dataset consisting of 100k customers and 100 attributes and the prediction label for each customer being label as 0 for non churn customers or 1 for churn customers and the second dataset consisting of 51k customers and 58 attributes and the prediction label for each customer being label as No for non churn and Yes for churn customers using the pandas library.we had to separate the labels from both datasets and store them in separate variables.

In the preprocessing step of the data, to dealt with the problem of null values we first separated the numerical and categorical attributes and then substituted the null values with the mean values of their respective attribute and categorical values using the mode value. Categorical variables are then converted into numerical attributes using label encoding and are concatenated with the original numerical attributes. For the problem of imbalance as the number of churn customers are relatively low than non churn customers we have used the SMOTE over-sampler to balance the data.Normalization of the whole data is done using the min-max scaling method so that the whole data is scaled down properly. Removal of attributes which are irrelevant to the prediction such as customer-id, area,ethnic,change-rev,change-mou is done.

$VIF = (\frac{1}{Tolerance})$

For feature selection that we have done for the last module of our project i.e the plot the matrix of the data features and finding the correlation between them and dropping the features based on the evaluations, though produced good results but the accuracy of the models were not as perfect as we thought. so in this module of the project we have tried using four different methods to see which of them gave better results to us compared to the techniques we have used in the last module of the project. We have first tried implementing VIF (variance inflation factor)[2] feature selection technique which calculates strong correlation between each of the attributes in the dataset with other,if the value is 1 then there is no correlation and if higher than 5 then those attributes should be discarded or transformed them to decrease their correlation for better results. Also we have implemented the hierarchical clustering [3] feature selection technique which group the similar kind of attributes into clusters and are ranked based on the importance, by picking a single feature from each of cluster we can reduce the total number of features.We also implemented PCA dimensionality reduction technique which is used to reduce the total number of attributes by selecting only those attributes that covers the max variance in the data.Lastly we tried

implementing WOE (weight of evidence) which helps us find out the weight of the each attribute w.r.t to the final label in the data and is also useful in transforming the data i.e continuous or categorical attributes into a single value.

$WOE = ln(\frac{Percentage\ of\ good\ in\ class}{Percentage\ of\ bad\ in\ class})$
Here good and bad means the binary labels in the dataset.

Accuracy of the models using the VIF on first dataset.

Model	Accuracy
Decision Tree	0.528
Random Forest	0.584
XGBoost	0.592
LightBGM	0.598

After implementing the above four methods and using the four models i.e decision tree, random forest, lightGBM, XGBoost that we implemented in the last module we have observed that, VIF has produced results which were a little less accurate than our previous methods.For PCA dimensionality reduction we have selected to number of features we needed so as to cover 95 percent of variation in data and the results were produced were almost similar to that of the previous methods.Coming to the Hierarchical clustering feature selection the results were also similar to that of the previous ones. so, in order to valuate the efficiency of the methods, we thought of taking an another similar kind of dataset and try implementing these methods and models on that dataset. so we have taken a second data set and implemented the hierarchical clustering feature selection and the implementation of the four models on the new dataset is done and we found out the results are good and methods are working properly on datasets. Finally the last method we have tried to implement is the WOE (weight of evidence) and it has provided great results for all the models in main dataset compared to the methods used in previous module and that of the methods we implemented above.As we got better results for this method we wanted to try this method on the second dataset to test whether it gives better results than the methods we implemented above on this dataset too and found out that it gave better results for this dataset too.So, we have chosen the WOE methodology as the optimal one for the prediction of churn.

Accuracy of the models using the Hierarchical clustering feature selection on first and second dataset.

Model	Accuracy
Decision Tree	0.53
Random Forest	0.581
XGBoost	0.59
LightBGM	0.596

First Dataset

Model	Accuracy
Decision Tree	0.68
Random Forest	0.73
XGBoost	0.83
LightBGM	0.77

Second Dataset

In the next step the we have selected the top sixty percentile of features from the dataset using the select percentile method and implementation of the four models is carried out and for that we first had to split the dataset into train and test data using train-test-split with 70 percent of the data was used for training and 30 percent was used for testing. We have used decision tree, random forest, XGBoost and lightGBM models as the base models since these four models were the top performing models from the last module we wanted to test on these models. All the models are fitted using the train data and train label and are implemented using the 10 fold cross validation on training data set before predicting the test data. For the model evaluation we have used

different metrics such as accuracy of the model which is the correct prediction of the label of the test data to the complete data, false positive rate which is values that ares false but predicted true and true positive rate which is values that are true positive and predicted positive.

Accuracy of the models using the PCA feature selection on first and second dataset.

Model	Accuracy
Decision Tree	0.528
Random Forest	0.592
XGBoost	0.63
LightBGM	0.60

First Dataset

Model	Accuracy
Decision Tree	0.63
Random Forest	0.735
XGBoost	0.79
LightBGM	0.724

Second Dataset

After the model evaluation is completed the four models are passed as base models for the stacking ensemble method. We wanted to try implementing the stacking ensemble methodology in this module as we have implemented the ensemble in the last module we wanted to try and extend on that grounds to implement different types of ensemble like bagging,boosting,stacking and finally opted to implement the stacking ensemble. In this stacking ensemble the the four models that we are using decision trees, random forest, xgboost, lightbgm are set as base models and are trained using the training data and the results of predict of test data is used as the training for the meta model which in this case is the logistic regression model classifier which uses to learn and predict the output of churn.we have also tried implementing the different combinations of base model models in stacking ensemble using the above four models and we have got better results for the combinations of three models(XGBoost, DecisionTree, LightBGM) for the first dataset and for the combinations of two models(XGBoost, Random Forest) for the second dataset.

Accuracy of the models using the WOE on first and second dataset.

Model	Accuracy
Decision Tree	0.67
Random Forest	0.799
XGBoost	0.812
LightBGM	0.812

First Dataset

Model	Accuracy
Decision Tree	0.745
Random Forest	0.84
XGBoost	0.841
LightBGM	0.844

Second Dataset

We have observed that the results produced by the stacking ensemble have improved over the accuracy of the base models and we have achieved accuracy greater than that of the last module.
Finally the AUC curve is plotted for the stacking ensemble method and compared along with the four models.

We have successfully implemented the stacking ensemble with WOE and select percentile methods.
Though the percentage increase of the stacking model is not much greater than what we have got from the base models implementation we were able to reproduce underlying concept of stack ensemble.

To conclude we have implemented the customer churn prediction using the stack ensemble method successfully with help of WOE and select percentile methods from the preprocessing to the ensemble prediction, we were able to correctly implement the methods we wanted to implement and achieve a greater accuracy for the models

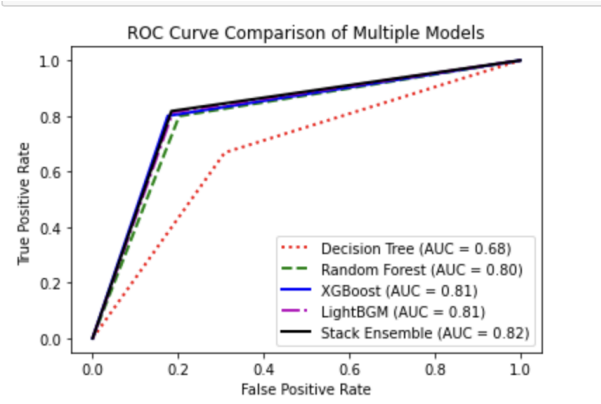

Fig. 1. AUC STACKING 1

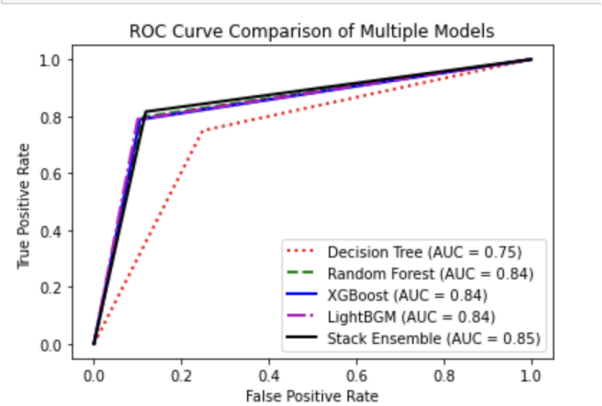

Fig. 2. AUC STACKING 2

then that we got from the previous module. though the stacking ensemble results were not to much higher than base models accuracy but we were able to implement and understand the underlying methodology.
Accuracy of the models after stacking on first and second dataset.

Model	Accuracy
Decision Tree	0.67
Random Forest	0.799
XGBoost	0.812
LightBGM	0.812
Best combination of stacking	0.82

First Dataset

Model	Accuracy
Decision Tree	0.745
Random Forest	0.84
XGBoost	0.841
LightBGM	0.844
Best combination of stacking	0.85

Second Dataset

Model	Accuracy
Decision Tree	0.59
Random Forest	0.63
XGBoost	0.634
LightBGM	0.636
Ensemble	0.638

Ensemble

AUC curves for stack ensemble of both datasets and Ensemble from the second module.

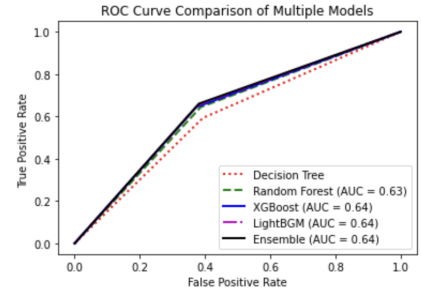

Fig. 3. AUC ENSEMBLE Curve

The followed approach for the implementation in this module.

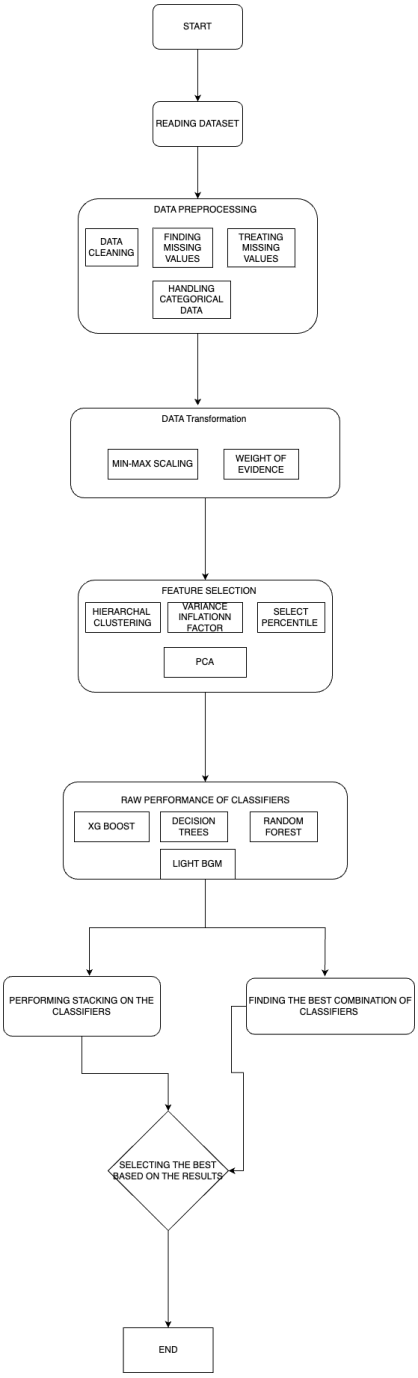

Fig. 4. Approach

5 WHO DID WHAT:

5.1 Member 1 (Sai Ram Dasarapu:)

- finding the second dataset.
- dropping of irrelevant features.
- conversion of categorical variables into continuous ones using label encoding.
- Implementing PCA techniques.
- Model exploration of Decision Tree and XGBoost.
- implementing Cross validation on the models.
- Performing the stacking Ensemble.
- Plotting the results using the AUC curve
  From Paper 1
- Found out what methods to implement and measures to take if the data is incomplete,noisy and have outliers.
- Was able to implement and understand how the ensemble learning methodology works.

5.2 Member 2 (Jashwanth Raj Gowllikar:)

- Finding the Null values in the dataset and replacing the data.
- Normalization of the data and implementing SMOTE on the imbalanced data.
- Feature selection using the select percentile method.
- Implementing of VIF, Hierarchical clustering feature selection.
- Implementing WOE and Hierarchical clustering on both the datasets.
- Model exploration of Random Forest,LightBGM.
- implementing Cross validation on the models.
- Implemented the stacking ensemble
- Extending stacking by finding the best combination of classifiers.
  From Paper 1
- Was able to find out how the proper feature selection helps achieve good predictions and accuracy.
- Was able to implement and understand how the ensemble learning methodology works.

6 FUTURE WORK:

The project can be further extended by

- Experimenting with different type of ensembles like bagging, boosting etc.
- Identifying the main features that really contributes to the prediction of churn by experimenting on various methods.
- We can also use models related to deep learning which can solve the problem more efficiently and accurately.
- Finding ways to properly handle the data like performing feature engineering so that the prediction is smooth and accurate.
- Exploring different type of data transformations.
- Handling multi-colinearity with proper preprocessing of data.
- Exploring different measures for evaluating churn prediction.

7 REFERENCES:

(1) Wang, Xing Nguyen, Khang Nguyen, Binh. (2020). "Churn Prediction using Ensemble Learning", ICMLSC 2020: The 4th International Conference on Machine Learning and Soft Computing. 2020

(2) https://towardsdatascience.com/7-techniques-to-handle-multicollinearity-that-every-data-scientist-should-know-ffa03ba5d29

(3) https://scikit-learn.org/stable/auto_examples/inspection/plot_permutation_importance_multicollinear.html