CAR INSURANCE PURCHASE PREDICTION USING CROSS-SELL PREDICTION

Santhan Ratakonda [#1], Sai Charan Reddy Pannala [#2], Jaswanth Manimala [#2]
Maneesha Kurremula [#4]
Computer Science Department, University of Central Florida

Abstract*—This model predicts that if the customer is interested in purchasing the vehicle insurance given by the company who are also an existing customer that they purchased health insurance from the same company. What is an insurance? Insurance is an accord that if the customer undergoes any kind of loss for example loss, injury, sickness, or death due to any kind of situation which is unfortunate in this case company provides a compensation for it. In this case they have to monthly guarantee amount which in not refundable. In this model we have developed an algorithm which predicts if the customer is interested or not based on previous insurance history. It is very beneficial for the business to have a model to forecast whether a client would be interested in vehicle insurance since it allows it to plan its communication strategy to reach out to those customers and maximize its business model and revenue. Starting from data cleaning all the filling null values in the dataset. We got data set from kaggle which is having 12 columns about all insurance and what they bought. Find out all the outliers and data analysis we got to know about the main feature which is previous insurance based on that we performed GridSearchCV and RandomizedSearchCV which took lot of time to train model. After specific data cleaning that we run logistic regression algorithms but after results we got really low accuracy value. Then performed Random Decision Tree, Gaussian Naive Bayes, Bagging Classifier, AdaBoost Classifier, LightGBM Classifier algorithms the performance of them are acceptable for all of them except logistic regression. Some more data cleaning and performed QGBoost ensembling method that gives the highest accuracy value that is 94.3 which is greater than any other algorithm. Finally the directions for improving this algorithm are pointed out. The algorithm proposed in this paper has promising future in predicting process.

I. INTRODUCTION

This project is about the predicting if their customer will retain for another new type of insurance which is offered by the same company. Car insurance works similarly to medical insurance in that customers must pay an annual premium to the insurance provider firm in order for them to be compensated (referred to as "sum assured") in the event that their vehicle is responsible for an unfortunate accident. Building a model to forecast a customer's interest in Vehicle Insurance is very beneficial for the business because it allows it to design its communication strategy to reach out to those clients in the most effective way possible and maximize its business model and revenue. Now that you have demographic data, you can forecast if a customer will be interested in auto insurance.

Many of the existing companies trying to find out and asking every existing customer to find out if they are interested in this new policy or not. Many of companies are trying hard to find out if they are in interested or not. To find a solution for it we collected the data to train a model and get the highest accuracy, in more detailed information.

The importance of this we can almost easy to find if the client if they are interested in bying the insurance or not. This helps corporates to find them easily and only takes call for them and give advt for them only. This saves a lot of money for them. This paper examines the connection between a customer's demographics and the final transaction they make. This assignment is finished using both methodologies and machine learning models.

A. *Background and existing literature*

During the past 3 years there are many insurance buying prediction system using many algorithms on learning based regression, collaborative filtering, content based filtering and different hybrid filtering methods. These are implemented by using big data and neural networks and machine learning methods. We are going to discuss about two of the paper that are previous written. How they are performed initially and later how they improved.

1. Car Insurance buying prediction Model Development using Neural Network Approach

A Car insurance companies collect the data of the customers while using their contract. Where the collected information is of both the negatives sides and positive side of the driver. The took dataset from the kaggle which consist of 25 columns from that they have cleaned the unwanted data. Followed by data pre processing by filling null values by mean, meadian, or mode value based on the priority of the attribute column. Discretisations of a

continue column using Decision Tree. Discretisation means is the computing of changing the continuous variables values to the discrete variable values with the help of bins. And the continuous variables are changed to predicted probability. Apply with the decision tree classifier with different depths in it and later from this got to age_tree attribute column is the good predictor for target. Followed by overfitting and grouping them by age based on their previous annual subscriptions. In the existing model the data is grouped by age based on their type or premium subscriptions optimized to the small number of groups. This model predicts based on the people having to the same group have same level of subscriptions and similar to other attributes. Where as this process optimizes having lower accuracy.

Car Insurance bying prediction Model Development using Hyperparameter technique us Machine Learning Algorithms.

Hyperparameter tuning is a process of selecting a group of optimal hyperparameters for learning an algorithm. Which is a key to machine learning algorithms. Hyperparameter is argument model which assign the value before learning process begins. In this model the data consists of 381109 rows and 12 features. Here it has a classification represent variable which gives the result that if the client is interested in vehicle insurance or not. Firstly starting from filling the null values. Check if any duplicates are present in the dataset to remove them. Followed by the data cleaning and preparation. And also for which values having the range from 20 to 100 they are normalizing to 1 to 10 that is making all the values which are rationally equal to and same under the same scale. While analyzing the data they observed that the age are categorized into three different ranges of time they are young, mid, senior level ages. And one more category also found to extract more information which is valuable information that is region code and policy sales channel and also analyzing and get more information about the non dependent variables features using some graphs. They applied mutual information technique and kendall's rank correlation coefficient (KRCC) for numerical features and also for the categorical features. Using different models like Ada Boost, Decision Tree Classifier, LightGBM and more algorithms to get good accuracy moreover by applying hyperparameter tuning in order to avoid overfitting to get more than previous accuracy.

2. *Dataset Overview:*

Our dataset consists of around 381109 rows of clients and 12 columns about each client details. And the details in each column has different ranges but at the end of training we bring the data under the same scale range. Some of the rows having unique numbers of primary ID which differs for every client those are removed before training the data. The size of the complete data set is of 29.5 Mega Bytes. Where as train data is of 21.43 Mega Bytes and test data is of 8.03 Mega Bytes. All if the data is in the format of .CSV format.

3. *Problem :*

Now a days everything in the world is almost automated from the time woke up to till sleep and some of them continuous working even when you sleep. Here come the insurance which play a crucial role in life. Where many of the companies offers different types of insurance. In which one of the insurance offered by one company is Car insurance same company also offers some of other insurance. Here comes the problem. The Insurance provided company want if the their customer is interested in the car insurance offered by them who is also their customer for health, Life, House ect insurance. From the data they have collected by those who took insurance previously or asked if they are interested we need to get guess the clients who are also interested in taking this car insurance. We need to provide a solution when clients gives certain amount of data about the customer we need give a categorical output yes if the customer is interested, No, if one is not interested.

I. IMPORTANT DEFINITIONS

A. *Data*

Dataset employed from Kaggle

Dataset: https://www.kaggle.com/datasets/anmolkumar/health-insurance-cross-sell-prediction?select=train.csv

B. Prediction Target

Whether a customer is willing to buy vehicle insurance or not using prediction with health insurance

C. Variables

Variable	Definition
id	Unique ID for the customer
Gender	Gender of the customer
Age	Age of the customer
Driving_License	0 : Customer does not have DL, 1 : Customer already has DL
Region_Code	Unique code for the region of the customer
Previously_Insured	1 : Customer already has Vehicle Insurance, 0 : Customer doesn't have Vehicle Insurance
Vehicle_Age	Age of the Vehicle
Vehicle_Damage	1 : Customer got his/her vehicle damaged in the past. 0 : Customer didn't get his/her vehicle damaged in the past.
Annual_Premium	The amount customer needs to pay as premium in the year
Policy*Sales*Channel	Anonymized Code for the channel of outreaching to the customer ie. Different Agents, Over Mail, Over Phone, In Person, etc.
Vintage	Number of Days, Customer has been associated with the company
Response	1 : Customer is interested, 0 : Customer is not interested

II. OVERVIEW OF PROPOSED APPROACH / SYSTEM

Health Insurance Cross-sell Prediction data was used to conduct the research, and Kaggle, an open-source site, provided the data. The data used are data related to past car history or current car insurance in addition to personal information of existing health insurance customers. We planned to work on multiple existing algorithms to train for the dataset. The hyperparameter tuning and metrics evaluation for model building is really necessary. So, for the proposed approach, we planned to try various methods for tuning and evaluating. The proposed model using an independent variable that directly affects the dependent variable, it is possible to forecast the recurrence of a specific event. Through parameter inference, logistic regression analysis can assess the impact of explanatory variables on response values and describe the patterns of linkages and interconnections caused by the model structure. We planned to compare with other algorithms after tuning.

III. TECHNICAL DETAILS OF PROPOSED APPROACH / SYSTEM

A. Data Description

We used the data obtained from Kaggle.com. It contains information about the 381,109 policyholders' demographics, vehicles, and policies in 12 columns without any missing data. The full description and the type of variables are

Variables	Description	Type
id	Unique customer ID	int64
Gender	=Male if the customer is Male, Female otherwise	object
Age	Age of the customer	int64
Driving_License	=1 if customer has driving license, 0 otherwise	int64
Region_Code	Unique code for the region of the customer	float64
Previously_Insured	=1 if customer has vehicls insurance, 0 otherwise	int64
Vehicle_Age	Age of the vehicle, <1 yr, 1-2yr, >2yr	object
Vehicle_Damage	=Yes if customer had vehicle damage in the past, No otherwise	object
Annual_Premium	Annual premium that customer needs to pay	float64
Policy_Sales_Channel	Anonymized code for the different sales channels	float64
Vintage	Number of days the customer has been associated with the company	int64
Response	=1 if customer is interested in vehicle insurance, 0 otherwise	int64

B. Evaluation Metrics

To evaluate our model and to obtain the accuracy and error rate of our models before and after hyperparameter tuning. We used some metric evaluation techniques. They are ROC-AUC Score, Accuracy, Confusion Matrix, Recall, Precision, Log Loss

After testing across various models and evaluation, we have compared different evaluation metrics and ROC Curve where the ROC Scores and Parallel Coordinates Plot shows all the combinations of hyper-parameters used for tuning the model to get the best parameters.

ROC-AUC Curve

- A measurement tool for binary classifier issues is the Receiver Operator Characteristic curve. It is a probabilistic curve which distinguishes the "signal" from the "noise" by plotting the TPR against FPR at different criteria. The capacity of a classifier to differentiate between classes is measured by the Area Under the Curve, which is used to summarize the ROC curve.

C. Hyper-Parameter Tuning

It is necessary for building model to avoid overfitting and better accuracy. We used techniques HalvingRandomizedSearchCV, GridSearchCV, RandomizedSearchCV. Every method gave the same result, but GridSearchCV and RandomizedSearchCV took a huge amount of time to train the models. HalvingRandomizedSearchCV took the least time to train the models and predict the output. So, we have used HalvingRandomizedSearchCV for Hyper-Parameter Tuning.

```
**********************************************
Best Score for DecisionTreeClassifier : 0.8781552930630498
---
Best Parameters for DecisionTreeClassifier : {'splitter': 'random', 'random_state': 23, 'min_weight_fraction_
----------------------------------------------
Elapsed Time: 00:02:26
==============================

Evaluation of DecisionTreeClassifier after tuning:
----------------------------------------------
   Accuracy_Score  Precision  Recall  F1_Score  ROC_AUC_Score  Log_Loss
0        0.878172        0.0     0.0       0.0            0.5  4.207802
```

D. Baseline Methods for Comparison

We have applied different Machine Learning Models to our data set and see how each of them performs. Firstly, We will tune the hyper-parameters of those models and then we will compare and choose the best model among them, based on Elapsed Time and Evaluation Metrics of the best parameters.

List of Machine Learning Models we are going to train and evaluate our data set on Decision Tree, KNN(k-nearest neighbors), Bagging Classifier, Logistic Regression, Bagging Classifier, LightGBM, Logistic Regression

E. Overall Performances

i. Decision Tree: The decision-making process is frequently represented as decision trees, which have a branching, tree-like form. The method involves branching decisions that lead to results, creating a structure or visualization that resembles a tree. The database attributes will be divided using a decision tree technique and a value function. Pruning, a method used to delete nodes that might use pointless features, is done on the decision tree prior to it being optimized.

Hyper-Parameter Tuning:
random_state: Estimator randomness is controlled.
splitter: The strategy used to choose the split at each node.
max_leaf_nodes: Grow a tree with max_leaf_nodes in best-first fashion.
max_depth: The maximum depth of the tree.
max_features: Amount of features that can be used to achieve better split

```
################################################################
     <<<< Tuning Model: Halving_Randomized_Search_CV >>>>
****************************************************************
------------------------------------------------
DecisionTreeClassifier
------------------------------------------------

Evaluation of DecisionTreeClassifier before tuning:
------------------------------------------------
   Accuracy_Score  Precision    Recall  F1_Score  ROC_AUC_Score  Log_Loss
0        0.824994    0.27807  0.273458  0.275745       0.587483  6.044574

 Evaluation of DecisionTreeClassifier after tuning:
 ------------------------------------------------
    Accuracy_Score  Precision  Recall  F1_Score  ROC_AUC_Score  Log_Loss
 0        0.878172        0.0     0.0       0.0            0.5  4.207802
```

ii. Gaussian Naive Bayes: An approach for probabilistic classification called Gaussian Naive Bayes is derived from sturdy independence requirements.

Hyper-Parameter Tuning:
var_smoothing: Portion of the largest variance of all features that is added to variances for calculation stability.

```
################################################################
     <<<< Tuning Model: Halving_Randomized_Search_CV >>>>
****************************************************************
------------------------------------------------
GaussianNB
------------------------------------------------

Evaluation of GaussianNB before tuning:
------------------------------------------------
   Accuracy_Score  Precision    Recall  F1_Score  ROC_AUC_Score   Log_Loss
0        0.687571   0.268878  0.910044   0.41511       0.783375  10.791173

Evaluation of GaussianNB after tuning:
------------------------------------------------
   Accuracy_Score  Precision    Recall  F1_Score  ROC_AUC_Score   Log_Loss
0        0.689337   0.269544  0.906454  0.415527       0.782835  10.730149
```

iii. AdaBoost Classifier: A method called AdaBoost seeks to construct a single strong classifier by merging several weak classifiers. A single classifier might not be able to predict an object's class with sufficient accuracy, but by grouping several weak classifiers and having each one gradually learns from the incorrectly classified items of the others, we can generate a very strong model.

Hyper-Parameter Tuning:

n_estimators: The maximum number of estimators at which boosting is terminated. In case of perfect fit, the learning procedure is stopped early.
random_state: Estimator randomness is controlled.
learning_rate: Weight applied to each classifier at each boosting iteration.

```
######################################################################
     <<<< Tuning Model: Halving_Randomized_Search_CV >>>>
**********************************************************************
--------------------------------------------------
AdaBoostClassifier
--------------------------------------------------

Evaluation of AdaBoostClassifier before tuning:
--------------------------------------------------
   Accuracy_Score  Precision  Recall  F1_Score  ROC_AUC_Score  Log_Loss
0        0.878172        0.0     0.0       0.0            0.5  4.207802
Evaluation of AdaBoostClassifier after tuning:
--------------------------------------------------
   Accuracy_Score  Precision  Recall  F1_Score  ROC_AUC_Score  Log_Loss
0        0.878172        0.0     0.0       0.0            0.5  4.207802
```

iv. Bagging Classifier: An ensemble meta-estimator known as a bagging classifier fits base classifiers one at a time to random subsets of the original dataset, and then combines each prediction (either by voting or by averaging) to get the final prediction.
Hyper-Parameter Tuning:
n_estimators: The maximum number of estimators at which boosting is terminated. random_state: Estimator randomness is controlled.

```
######################################################################
     <<<< Tuning Model: Halving_Randomized_Search_CV >>>>
**********************************************************************
--------------------------------------------------
BaggingClassifier
--------------------------------------------------

Evaluation of BaggingClassifier before tuning:
--------------------------------------------------
   Accuracy_Score  Precision    Recall  F1_Score  ROC_AUC_Score  Log_Loss
0        0.853647    0.30408  0.156221  0.206403       0.553311  5.054895
Evaluation of BaggingClassifier after tuning:
--------------------------------------------------
   Accuracy_Score  Precision   Recall  F1_Score  ROC_AUC_Score  Log_Loss
0        0.853349   0.303517  0.15737  0.207272       0.553636  5.065167
```

v. LightGBM Classifier: For ranking, classification, and many other machine learning problems, LightGBM is a quick, distributed, high performance gradient boosting framework based on decision tree techniques.

Hyper-Parameter Tuning:
n_estimators: Number of Boosting iterations.
learning_rate: This setting is used for reducing the gradient step. It affects the overall time of training: the smaller the value, the more iterations are required for training. min_data_in_leaf: Minimal number of data in one leaf. Can be used to deal with over-fitting random_state: Estimator randomness is controlled.

```
######################################################################
     <<<< Tuning Model: Halving_Randomized_Search_CV >>>>
**********************************************************************
--------------------------------------------------
LGBMClassifier
--------------------------------------------------

Evaluation of LGBMClassifier before tuning:
--------------------------------------------------
   Accuracy_Score  Precision    Recall  F1_Score  ROC_AUC_Score  Log_Loss
0        0.878215   0.545455  0.002154  0.004291       0.500952  4.206292
```

```
Evaluation of LGBMClassifier after tuning:
--------------------------------------------------
   Accuracy_Score  Precision  Recall  F1_Score  ROC_AUC_Score  Log_Loss
0        0.878172        0.0     0.0       0.0            0.5  4.207802
```

vi. Logistic Regression:A statistical model called logistic regression makes use of the logistic function to model the conditional probability. We determine the conditional probability of the dependent variable Y for binary regression given the independent variable X. P(Y=1|X) or P(Y=0|X) can be used to represent it.

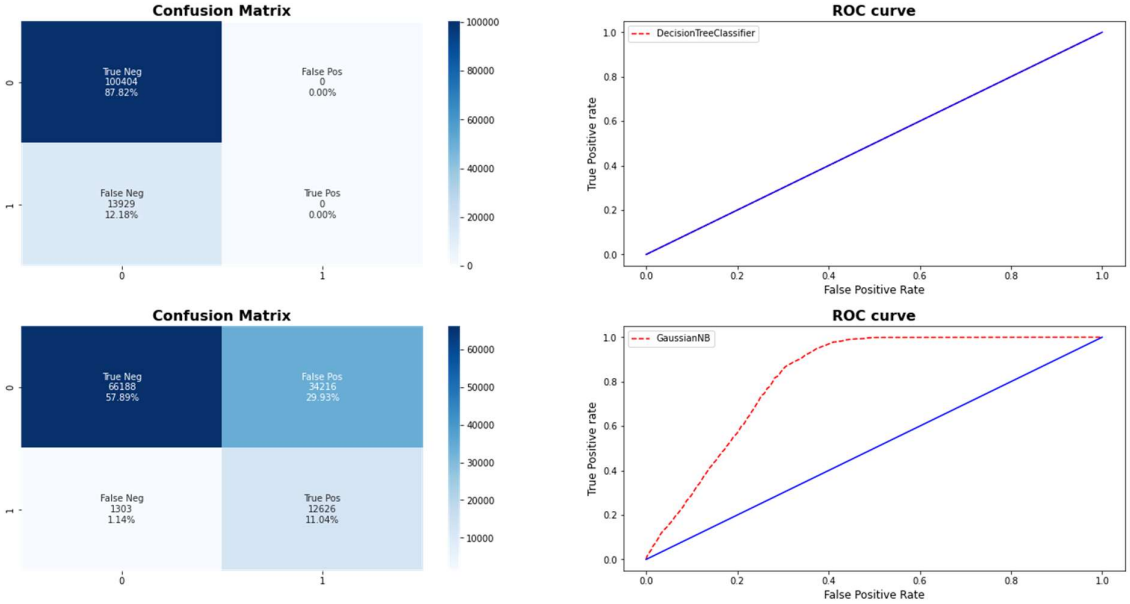
Hyper-Parameter Tuning:
penalty: Specify the norm of the penalty.
solver: Algorithm to use in the optimization problem.
C: Inverse of regularization strength
random state: Estimator randomness is controlled.

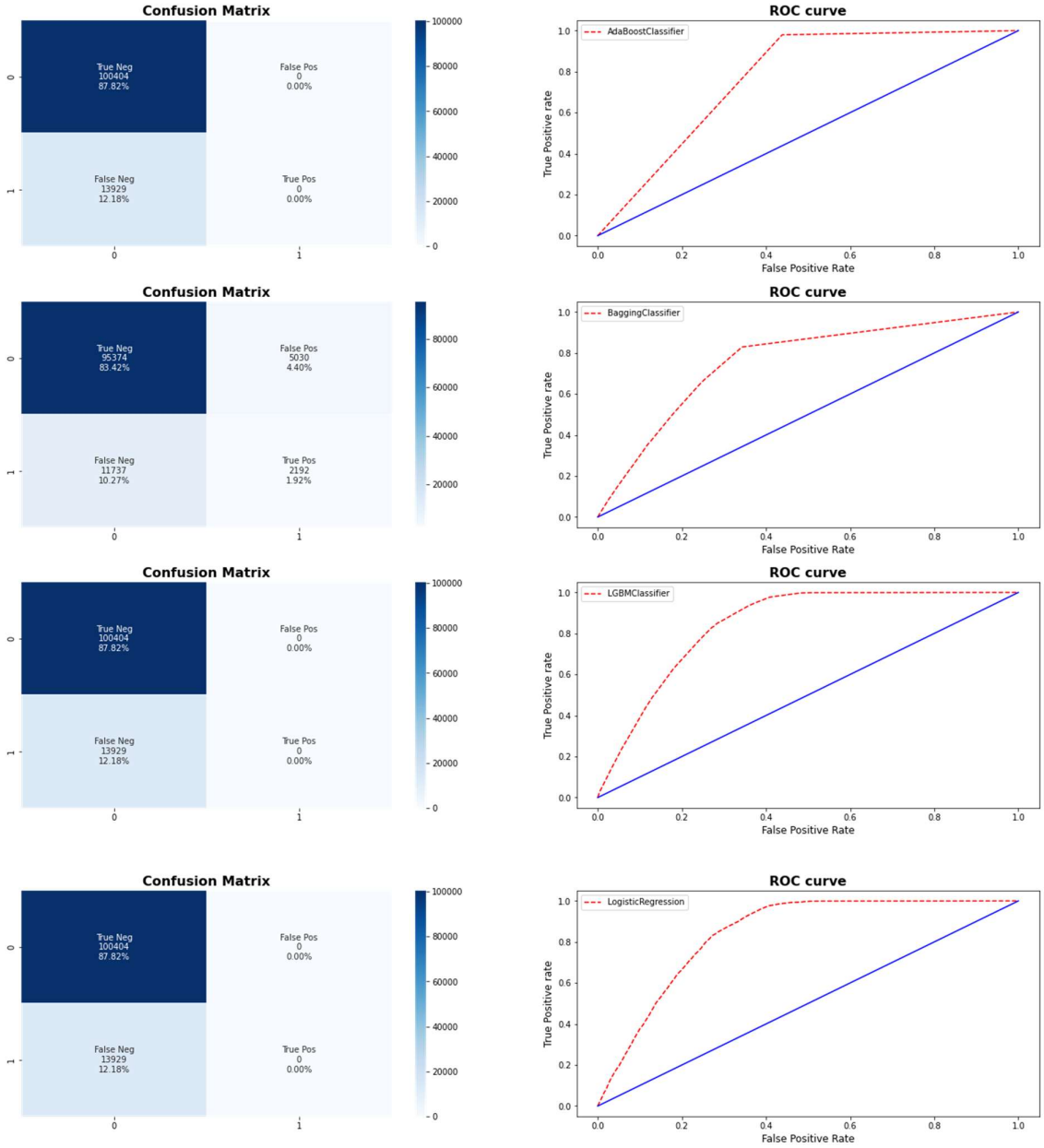
```
################################################################
     <<<< Tuning Model: Halving_Randomized_Search_CV >>>>
****************************************************************
--------------------------------------------------
LogisticRegression
--------------------------------------------------

Evaluation of LogisticRegression before tuning:
--------------------------------------------------
   Accuracy_Score  Precision  Recall  F1_Score  ROC_AUC_Score  Log_Loss
0        0.878172        0.0     0.0       0.0            0.5  4.207802

Evaluation of LogisticRegression after tuning:
--------------------------------------------------
   Accuracy_Score  Precision  Recall  F1_Score  ROC_AUC_Score  Log_Loss
0        0.878172        0.0     0.0       0.0            0.5  4.207802
```

F. EXPERIMENTS





G. RELATED WORK

Recently, various approaches for prediction systems have been created that may make use of various baseline algorithms [1,] [5,] [6]. The hyperparameters and metrics evaluation used by many of the prediction systems mentioned above are searches. The model tuning grid and random search. The weights acquired during the training of a linear regression model are parameters, whereas the number of trees in a random forest is a model hyperparameter. Hyperparameters can be compared to model choices. Using association rules and decision tree to predict cross selling opportunities [10]. There are models built on multiple algorithms using two class decision tree.

H. CONCLUCSION

As we worked on a prediction system which proposes a different method to solve the same problem of insurance prediction. The Data wrangling after loading the dataset applied a null check and treated outliers using a quantile method. Being the outlier's treatment done, the initial observations showed us the dataset has multiple numeric columns with different scales and so we applied a minmax scalar technique for

normalization of data and feature scaling. During Exploratory data analysis, we focused on 4 variables and categorized them for initial analysis based on age, region code, vintage and policy sales. We proceeded further by encoding categorical values for categorical columns using one hot encoding. So, we can extract information from the columns and understand them. The feature engineering applied on the dataset was extracted and measured how one variable can tell us about other variables through feature importance. As for the model fitting, the baseline algorithms we have used have performed well. Before adjustment, the accuracy scores for all models ranged from 68% to 85%. Following model adjustment, we were able to achieve an accuracy of about 87%. Where LightBGM Classifier outperformed during the experiment after using hyperparameter training and metrics evaluation.

REFERENCES

[1] https://support.sas.com/resources/papers/proceedings17/0941-2017.pdf
[2] https://gutentagworld.wordpress.com/2020/12/13/health-insurance-cross-sell-prediction/
[3] https://www.irjmets.com/uploadedfiles/paper/issue_7_july_2022/27508/final/fin_irjmets1656843849.pdf
[4] https://towardsdatascience.com/faster-hyperparameter-tuning-with-scikit-learn-71aa76d06f12
[5] https://fadilah.webflow.io/works/insurance-cross-selling-february-2021
[6] https://ieeexplore.ieee.org/abstract/document/4620698

Link for code: https://github.com/SanthanRatakonda/Prediction_System