FINAL REPORT

Categorized Red Hat Customer Potential to drive business value

TEAM – 21

CHARAN PATNAIK AVS - 002762551

RISHABH JAIN - 002769493

# **Project Topic**

Categorized Red Hat Customer Potential to drive business value

**Team Number**: 21

**Members**:

1. Venkata Sai Charan Patnaik Amiti
2. Rishabh Jain

# **Introduction**

**Background:**

Red Hat, Inc. is an American software corporation that sells open-source software to businesses. The company develops, maintains, and contributes to a wide range of open-source projects. Through business mergers and acquisitions, it acquired various proprietary software product codebases and published such software under open-source licenses. After Intel, Red Hat is the second largest corporate donor to the Linux kernel version 4.14 as of March 2016.

**Motivation:**

Customers are the Gods who can make or break a business. If the behavior of customers can be predicted, it will greatly aid an organization's business planning. Red Hat, like most businesses, can gather a lot of information about the behavior of people who interact with them over time. We are finding better ways to use this behavioral data to predict which people to approach and even when and how to approach them.

The provided input data contains two major data sets, namely people data and activity data. People dataset comprises all people ids, defined characteristics, and conducted activities across time. Activity data comprises all the unique activities and activity features that each person has completed over time. Each entry in the activity file indicates a distinct activity carried out by a person on a specific day.

In this project take step by step approach which includes data-preprocessing to AUC score calculation. Firstly, in data preprocessing we will address missing values and outliers followed by Exploratory data analysis to understand more about the data we will be working on. In feature engineering we would create some required meaningful features for our next step. Then, we execute multiple machine learning classification methods to fit and predict which activities incur to required outcome. Finally, we would validate our models using AUC score.

**Goal:**

Red Hat has provided masked data about its customers and their activities. We must predict whether a particular customer with certain characteristics have any business value based on these characteristics.

# **Methodology:**

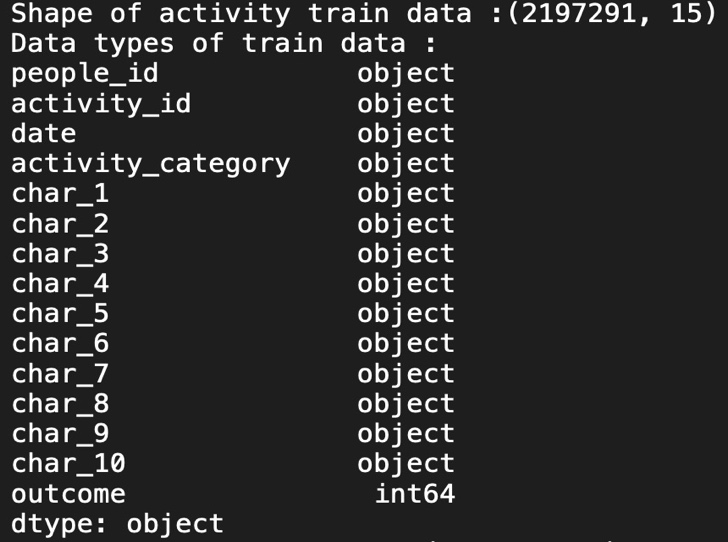
## **Data Extraction:**

The project has two main data sets namely **activity dataset and people dataset.** The activity data set was split into **train and test datasets** to execute and run the models.

Key Observations:

* There are 189k potential customers and 2.1M customer activities in the training set.
* The test set contains 498k customer activities (train/test split of activities 18.5% in test)
* We will have to perform EDA and then combine all the datasets ( expect for test data set ) to execute our models over.

Mentioned below are the size and datatypes of the variables in the **Train** **dataset:**



Mentioned below are the size and datatypes of the variables in the **Test** dataset:

Text

Description automatically generated

Mentioned below are the size and datatypes of the variables in the **Test** dataset:

Timeline

Description automatically generated

## **Data Preprocessing:**

* As seen above since all the datasets have ‘date’ variable we need to convert it into ‘datetime64[ns]’ format to ensure consistency and reliability.

Text

Description automatically generated

* Renaming the similarly named columns and keeping primary and foreign key i.e ‘people\_id’ consistent across.

Text

Description automatically generated

* Checking for null values: Using the below code we check if null values exist

Graphical user interface, text, application, chat or text message

Description automatically generated

Text

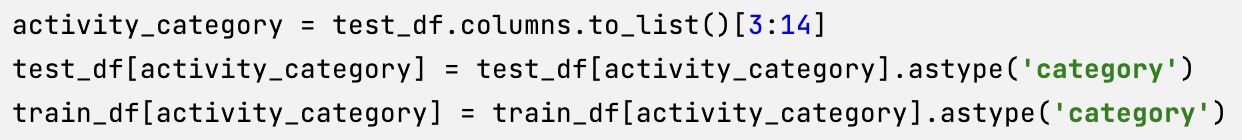
Description automatically generated

* Replacing the missing values with ‘type -1’

Text

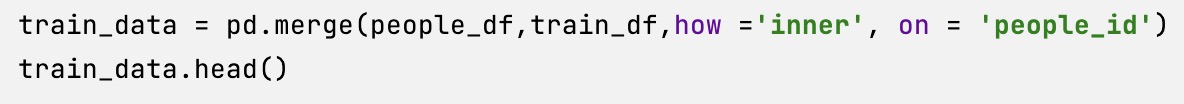
Description automatically generated

* Converting characteristics features into ‘category’ format to ensure classification:



## **Exploratory Data Analysis:**

* Merging both people dataset and activity test dataset over ‘people\_id’:



* Now we find the distribution of outcome values being ‘0’ or ‘1’ in the activity test data set to understand the data better.

Chart, pie chart

Description automatically generated

* We find and verify the distribution of people belonging to outcome value ‘1’

Graphical user interface, application, table, Excel

Description automatically generated

* We also find the distribution of the people belonging to the outcome value ‘0’

A picture containing graphical user interface

Description automatically generated

* We find the activity type which produce the outcome ‘1’

Chart, bar chart, funnel chart

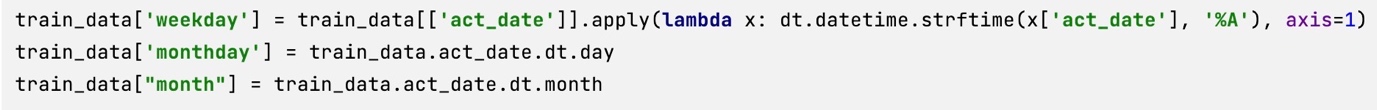
Description automatically generated

* We also find and validate activity type which produce the outcome ‘0’

Chart

Description automatically generated

* We need to extract weekday, day, and month values from date column for further analysis.



* Distribution of number of activities over each weekday

Chart, box and whisker chart

Description automatically generated

* Distribution of number of activities over each of the month

Chart

Description automatically generated

* Distribution of activities over each month of the year

Chart, histogram

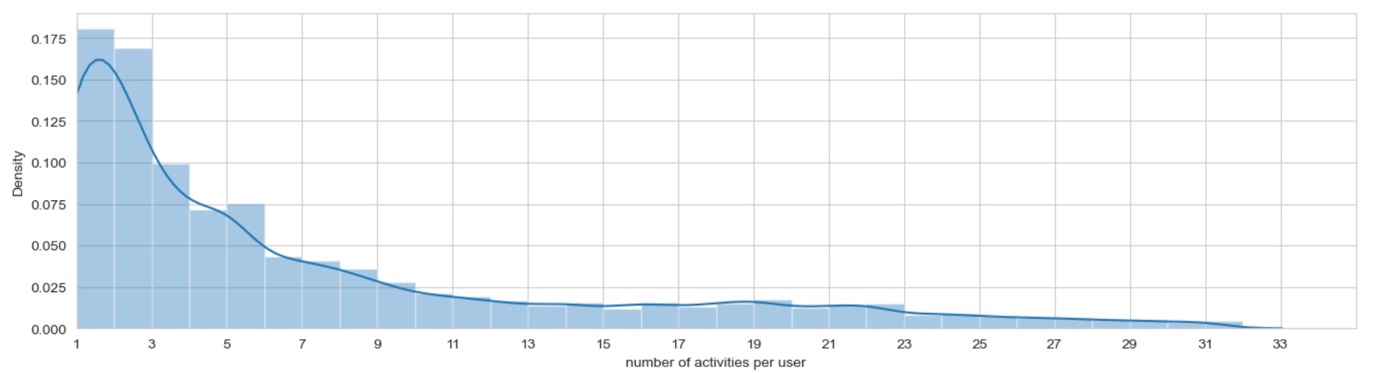
Description automatically generated

* **Outlier detection:** We need to find if there are any people who have outlying value for total number of activities. For this we are setting and finding the value of the outlier for the distribution curve using below code**.**

**Text, letter

Description automatically generated**

* Outlier Distribution graph:

****

* **Verdict:** There are very less people performing activities over the outlier value, hence this dataset does not require processing of the outliers.

## **Feature Engineering:**

* Creating new columns named ‘is\_train’ and ‘outcome’ in train and test dataset.

**Text

Description automatically generated with medium confidence**

* Combine newly created train and test data:

****

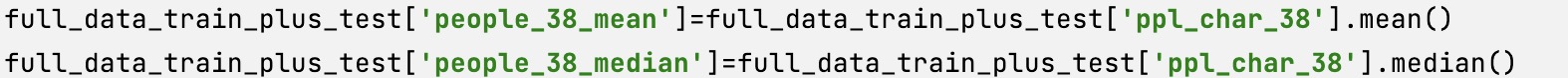
### **group\_ppl\_cnt** :Number of people per each group



### **days\_from\_min\_dt\_grp**: Days passed from Min ppl date per group



* **Mean and Median for ‘char\_38’ variable:**

****

### **group\_activity\_cnt** : Number of activities per group



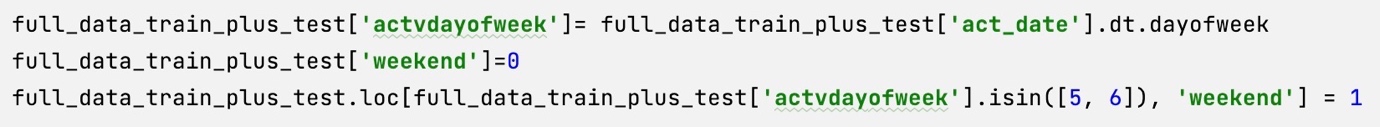
### **activt\_per\_ppl\_cnt**: Number of activities per people

****

### **ppl\_actv\_date\_diff**: Absolute difference between ppl data and activity date

****

* Check if **weekend** or not:

****

### **actv\_date\_qurtr:** Quarter of the activit

### 

* Split the data into train and test to perform next steps:



* Resampling the data reduce the number of data rows in the sample:



* All the categorical variables converted into numerical values using one-hot encoding.

Example for feature : ‘activity\_category’

Text

Description automatically generated

* Finally all the data is normalized to be utilized in machine learning models :

Text

Description automatically generated

## **Classification:**

For this dataset multiple algorithms are used for training. The algorithms include Logistic regression, Decision Tree and LightGBM. Hyperparameter tuning was done for each algorithm.

* **Logistic Regression**:

The probability of a target variable is predicted using the supervised learning classification algorithm known as logistic regression. Since the dependent variable's nature is dichotomous, there are only two viable classes.

Simply said, the dependent variable is a binary variable, with data recorded as either 1 (which represents success/yes) or 0 (which represents failure/no).

A logistic regression model makes mathematical predictions about P(Y=1) as a function of X. One of the most basic machine learning algorithms, it may be applied to a number of categorization issues, including spam identification, diabetes prediction, cancer diagnosis, etc.

**Reason of Utilization:**

We use this model as it is the starting point for any classification ML model because of its simplicity and speed. We will also use SGDclassifier with log loss for faster and efficient results.

* Now we execute the and plot graph for AUC score to get the **right hyperparameter values** for logistic regression

Text

Description automatically generated

Chart, line chart

Description automatically generated

* Fit and predict the SGDC classifier given below:

Text

Description automatically generated

* **Decision Tree** **:**

The most effective and well-known technique for categorization and prediction is the decision tree. A decision tree is a type of tree structure that resembles a flowchart, where each internal node represents a test on an attribute, each branch a test result, and each leaf node (terminal node) a class label.

**Reason for Utilization:**

We leverage this algorithm to efficiently improve our accuracy score as this Decision Tree is more complex in comparison to Logistic regression without much gain in execution time. They boost predictive models with accuracy, ease of interpretation, and stability.

**Execute and plot AUC :**

**Text

Description automatically generated**

**Output Plot :**

**Chart, line chart

Description automatically generated**

* **Fit and predict the model with the training data and hyperparameters**

**Graphical user interface, text

Description automatically generated with medium confidence**

* **Light Gradient Boosted:**

Light Gradient Boosted Machine, or LightGBM for short, is an open-source library that offers a practical and efficient implementation of the gradient boosting technique.

By including a sort of autonomous feature selection and concentrating on boosting cases with greater gradients, LightGBM expands the gradient boosting technique. This can hasten training significantly and enhance prediction performance.

**Reason for Utilization:**

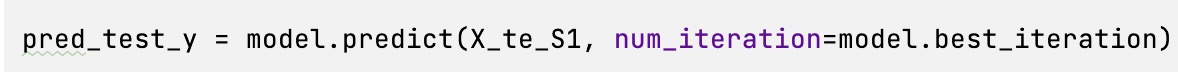
We will utilize the LightGBM algorithm for classification, given its ability to provide accurate classification results on large datasets, while avoiding overfitting. This will improve a lot in terms of accuracy when compared to the other two models but *drastically increases the execution time*.

Hence after multiple iteration we have chosen the below parameters, to get both high accuracy and save execution time.

**Text

Description automatically generated with low confidence**

* **Fit and Predict the model :**

****

# **Datasets:**

The dataset consists of three csv files:

* ***act\_train.csv***and***act\_test.csv***

These two constitute the ‘main’ dataset, representing the train and test dataset respectively. The data consists of an ID label, followed by **13 features**, and for the train dataset, the target label outcome as well.

There are **2197291** instances of train data and **498687** instances of test data.

* ***people.csv***

people.csv contains **189118** instances of data, each corresponding to a single user. Each user is identified based on a user id, which corresponds to an instance within test/train\_users.csv, followed by **40 features**.

* We will **combine** all three data sets to predict the required customer value.

The activity file contains several different categories of activities. Type 1 activities are different from type 2-7 activities because there are more known characteristics associated with type 1 activities (nine in total) than type 2-7 activities (which have only one associated characteristic).

To develop a predictive model with this data, you will likely need to join the files together into a single data set. The two files can be joined together using person\_id as the common key. All variables are categorical, with the exception of 'char\_38' in the people file, which is a continuous numerical variable.

LINK TO THE DATA SET :

<https://drive.google.com/drive/folders/1Lnj3fwMF0DjvEy0Om1LmKnJRH2XnwP8b?usp=share_link>

# **Conclusion:**

**Final AUC score to better understand accuracy of classification :**

**Text

Description automatically generated**

**Output Impressions:**

* As seen above the best accuracy value is achieved using Light GBM with AUC score of approximately **0.99, which also suggests overfitting.**
* Though, in terms of speed of execution and resources saved, Decision tree / Logistic regression are better models with very little impact on accuracy.

**Which model to choose ?**

* Two major factors in model selection is **Input data set and it’s domain understanding.**
* As seen from the data and premise of the whole project, we believe that the data provided has **Selection Bias** for the reasons mentioned below:

1. The provided data as very minimal anomalies. The outcome distribution between 0 and 1 is approximately equal in the given data set, therefore avoiding the necessity of oversampling and under-sampling techniques.
2. Furthermore, data set provided has no outliers when employed with statistical techniques to determine the outlier.
3. Our extensive research about the data set online revealed there is strong chance the data provided has **log loss**, that means Red hat has handpicked required rows & columns from another data set ( Raw data ) , processed it and has provided data, this also explains why there is **overfitting** even when using less complex ML models i.e Logistic Regression and Decision Tree.

* Hence, we can conclude the below mentioned insights:
  1. Red hat company has provided with handpicked processed data which might be the schema for all the input data which will be used for classification in the future.
  2. Thus, they are only looking to improve accuracy of the model.

**Result:**

Therefore from the above reasoning, overfitting is not a deterring factor since new data sets using the model in the future would still have the same log loss, processing and therefore the data schema.

Hence, even though there is overfitting and high execution time for **LightGBM**, it is the best model since, it has the highest accuracy value which directly adds to the business value of the company.

# **References:**

1. <https://medium.com/geekculture/prediction-for-redhat-business-value-6a714df469ac>
2. <https://scikit-learn.org/stable/modules/tree.html>
3. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
4. <https://xgboost.readthedocs.io/en/stable/python/python_api.html>
5. <https://bhavya0x83b.wordpress.com/2017/04/20/predicting-business-value-on-kaggle-for-red-hat/>
6. <https://www.kaggle.com/code/christianhritter/predictbusinessvalue-eda>
7. <https://www.kaggle.com/code/gvm1999/red-hat-prediction>
8. <https://www.kaggle.com/code/marinovik/predicting-business-value-with-lightgbm-red-hat>
9. <https://www.kaggle.com/code/abriosi/raddar-0-98-xgboost-sparse-matrix-python>