PROJECT REPORT

Acknowledgement

I would like to express my sincere thanks to Mr. Prasad S, HOD of our Department who helped in getting an internship in TCS.

I extend my sincere gratitude to the Faculty members of the ICT Academy of Kerala for giving me proper guidance about the work that I need to do and understand the project.

I would also like to thank my family and friends for supporting for and while doing this internship.

Project Title

Classification Model: Build Model that Classifies the Side Effects of Drugs

Abstract

As the society become more exposed to changing environmental conditions people started developing different health issues as an impact of their lifestyle and other phenomenon, so consumption of drugs have become prevalent among people irrespective of their age, gender etc.. All drugs we consume have side effects, which can endanger the health of the patients. To classify drugs based on side effects we used machine learning models like Logistic Regression, Random Forest etc. and model of best accuracy is selected. The dataset we considered contained two relevant columns, 'Satisfaction and Side effects' under the assumption less side effects results more satisfaction we build two models. The first model was build using Reviews as independent variable and satisfaction as target variable and fitted Random forest model and obtained accuracy 0.73 percent. The second model considered all the columns except reviews and the accuracy obtained under Logistic regression is 0.86.

Introduction

Drugs have become a vital part in the life of mostly all people in today's world. Drugs consumption may endanger health conditions of people. As different drugs are available in the market that to consumed for different kind of diseases as the age gender and health conditions of patients results in different kind of side effects. Analysing a dataset that contains list of many commonly used drugs for different conditions and their side effects and building a classification model may help people who seek to consume them so that they could be aware of the side effects they may have after the consumption. Here I used Machine learning methods to predict the side effect and build a classification model to classify different drugs based on their side effects.

The Classification model is a subset of Supervised Machine Learning that is used to identify the category of new observations on the basis of training data. In Classification, a program learns from the given dataset or observations and then classifies new observation into a number of classes or groups. Such as Yes or No, 0 or 1, Spam or Not Spam etc. Classes can be called as targets/labels or categories.

Internship Activities

After enrolling in the TCS platform I chose the project topic 'Classification Model: Build Model that Classifies the Side Effects of Drugs'. Got myself familiarized with the platform and the watched the welcome kit videos provided in the. Pre Assessment test was cleared by me in the 2nd attempt and started search for dataset in Kaggle.

I found a suitable dataset in the kaggle titled 'WebMD Drug Review Dataset'. The link of the dataset is provided below;

https://www.kaggle.com/datasets/rohanharode07/webmd-drug-reviews-dataset

Gone through the webpage and read all the article and description about the selected dataset. Visited the links of the provided in the project references in the TCS ion platform.

The project environment I used was python. Used Google Colab; a web IDE for python, to enable Machine learning with storage on the cloud.

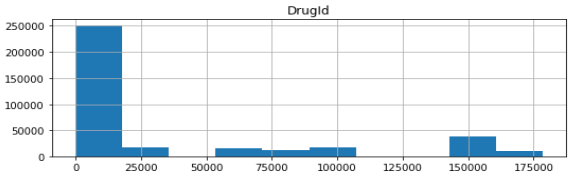
The methodology I followed is provided below

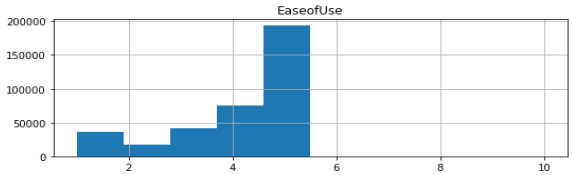
Methodology

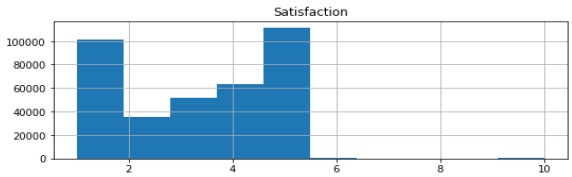
- Identify the problem
- Literature Survey
- Gathering Data
- Data Preparation
- Data Wrangling
- Analyse Data
- Train Model
- Test Model
- Deployment

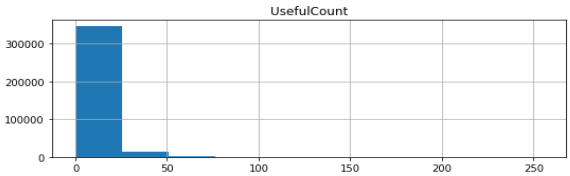
First I opened a notebook for EDA(Exploratory Data Analysis) and did some basic functions and plotted necessary diagrams the diagrams are given below;

Histogram;

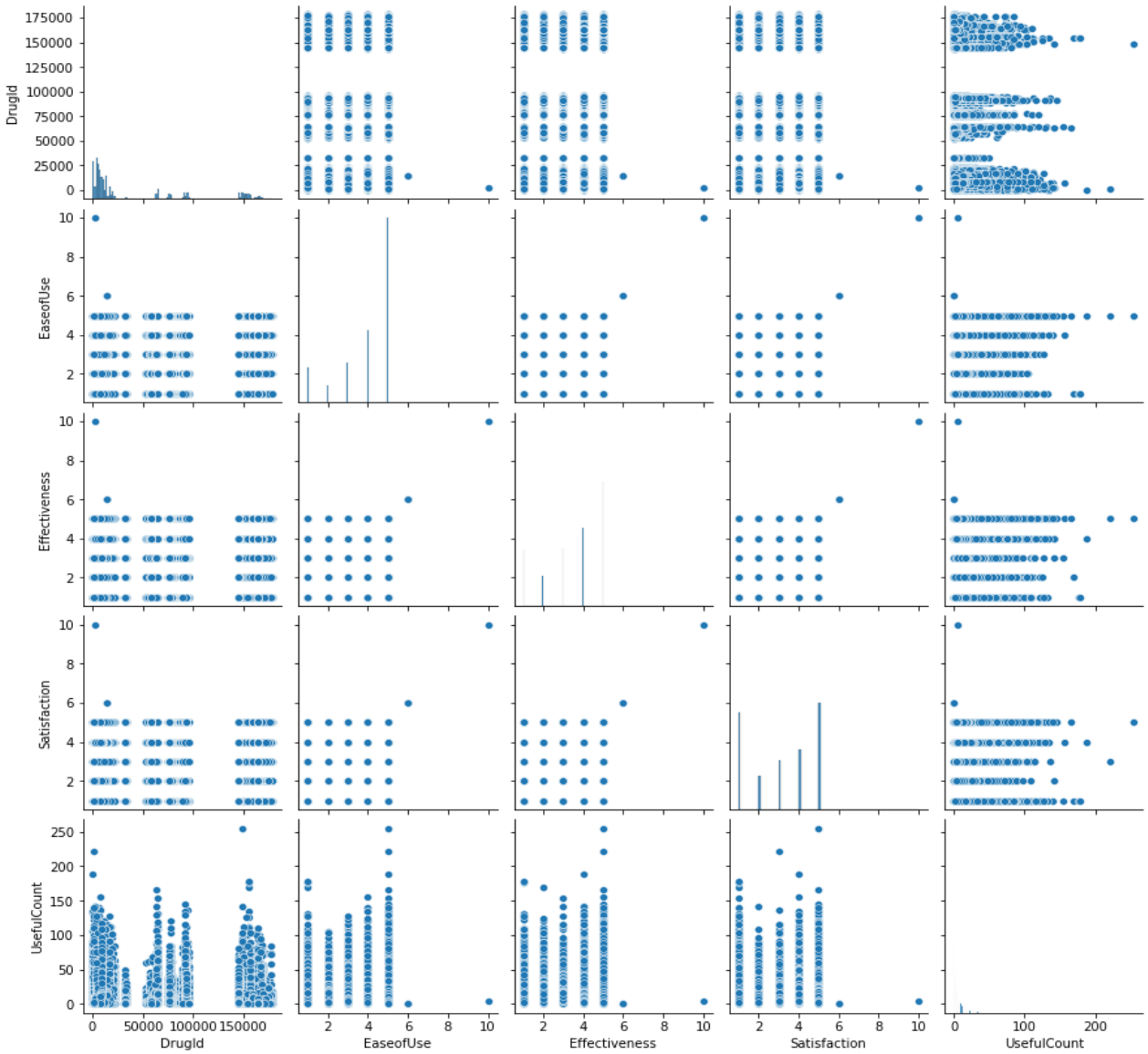








Scatter plot;



First Way of Approach

I opened a new notebook so that I could do the further steps in model fitting.

The steps I followed for Data preprocessing are as follows;

- Started working in my dataset, the initial target variable was set to be column of 'side effect' but it was changed to the column 'Satisfaction' in the assumption that less side effects means more satisfied.
- Satisfaction columns contained ratings in number 1-10.
- First considered only two columns 'Reviews' and 'Satisfaction' which had the dtypes object and int respectively.
- All other columns were dropped

Started to refer about functions to convert text in 'Reviews' column.

- Imported package 'nltk' which is a package used for building python programs. This package contains many libraries like tokenization, parsing, classification etc.
- Then download stopwords. Also import BeautifulSoup from bs4 and import 're'
- The following functions were defined using the keyword def to remove the html strips, Square brackets and noisy text

The fucnction I used is as follows;

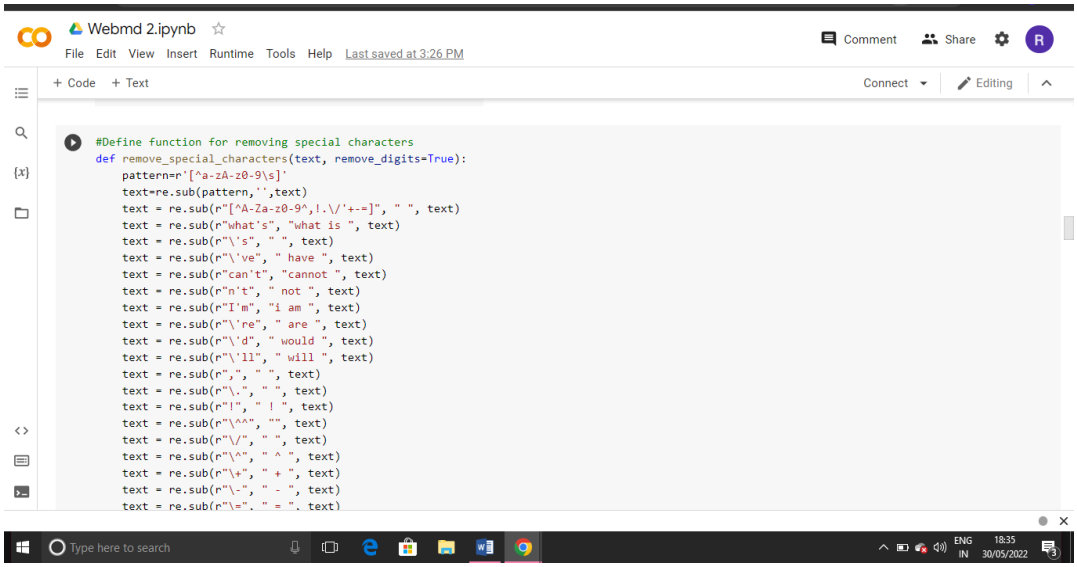
```
#Removing the html strips
def strip_html(text):
    soup = BeautifulSoup(text, "html.parser")
    return soup.get_text()

#Removing the square brackets

def remove_between_square_brackets(text):
    return re.sub('\[[^]]*\]', '', text)

#Removing the noisy text
def denoise_text(text):
    text = strip_html(text)
    text = remove_betBeautiween_square_brackets(text)
      return text
```

- The new dataset obtained was cleaned and didn't contained any html strips, square brackets and noisy text still that doesn't made a proper cleaned data still it contained special characters so.
- The functions to remove special characters were used the screenshot of the code is given below.



- Stemming of the text was also done (Stemming is a nlp technique that lowers inflection in words to their root forms).
- After applying all above functions then next I removed the stopwords for which I imported 'stopwords' from 'nltk.corpus'. Tokenization process was also done(Tokenization is a way of separating a piece of text into smaller units called tokens. Here, tokens can be either words, characters, or subwords. Hence, tokenization can be broadly classified into 3 types – word, character, and subword (n-gram characters) tokenization).
- After obtaining a cleaned "Review' column next I moved on to 'Satisfaction' column which is the target variable. The column satisfaction contained numbers from 1 to 10 which were ratings .There were three rows contained values more than 5 for convenience those three rows were dropped.
- The rest of the numerical data was changed o binary class (0 and 1) using the function given below

```
def relabel(z):
    if z<=3:
        return 0
    else:
        return 1
data1['Satisfaction']=data1['Satisfaction'].apply(re
```

- The text data was later transformed using Tfidifvectorizer which was imported from 'sklearn.feature_extraction.text'
- The data preprocessing was done fully now left was model fitting.
- The dataset was splitted into train and test data taking test_size=0.2 and random state=143.
- Referred and studied about each of the models through web one of the referred webpage is https://www.javatpoint.com/machine-learning-models.
- Decided to fit Logistic Regression first and gave the code.
- The model was fitted the accuracy score obtained is 0.7348182043719161

Second Way of Approach

As the accuracy was not upto 80 percent thought of doing it in considering only the columns which contained numerical values;

- A new notebook was opened in order to fit a model in different way.
- The libraries pandas , numpy, matplotlib.pyplot and seaborn were imported and the dataset was loaded into the notebook.
- Did explanatory data analysis found dtypes, shape , info etc,, the result obtained is as follows
    - Shape-`(362806, 12)`
    - Columns-(['Age', 'Condition', 'Date', 'Drug', 'DrugId', 'EaseofUse','Effectiveness', 'Reviews', 'Satisfaction', 'Sex', 'Sides' 'UsefulCount']
    - All the columns except 'DrugId', 'EaseofUse','Effectiveness' and 'Satisfaction' had dtype object

  Looked into the first column that is the 'Age', it is understood that it needed some processing the age was in the form of ranges as mentioned below;
  {array(['75 or over', '25-34', '65-74', '35-44', '55-64', '45-54', '19-24',' ', '13-18', '7-12', '0-2', '3-6'], dtype=object)}

  When I check the next column 'Condition' the it contained 1806 unique values and column 'Drug' contained 7093 unique values that implies this 0.36 million rows contained repeated drug names so thought of considering only 500 unique drug names, which repeated mostly. The rows were reduced by considering only these 500 drugs and the resulting output I obtained is as follows;
- The required drug was only considered that was named 'data2 'and I then moved on to the remaining columns and found out the unique values in it , the data considered was data2 which contained only 500 unique drugs.
- When it was considered the column 'sides' contained 266 the column 'Effectivenenss' contained 6 the column 'EaseofUse' contained 6 the column 'DrugID' contained 387 and the column 'Usefulcount' contained 147 unique values.
- The sex column contained

```
   Female    177838

   Male       71405
              17187
        reName: Sex, dtype:
int64
```

Of all the columns available in the dataset the required columns were chosen they were
`['Age', 'Drug', 'EaseofUse', 'Effectiveness', 'Sex', 'Useful Count','Satisfaction']`. The new dataset was named data3 and the processing was done in the new dataset.

- The target variable here was 'Satisfaction'. All columns except the target variable was given 'x' and I did *one hot encoding* using the function get_dummies
- Got a new form of dataset which contained only numerical values.
- Started transforming the column of target variable.
- The column of Target Variable which is the column of 'Satisfaction' ,I repeated the same process which I did in the *first way of approach*
- I fitted the first model and that is the decision tree[DecisionTreeClassifier(max_depth=32)]. The accuracy score I obtained is 0.8598693840783695
- Then I tried Random Forest and the accuracy obtained is 86.10329167135833
- For Logistic Regression, Accuracy is 0.8620463161055437

Description of the Algorithms I used for two approaches is given below;

Algorithms

Logistic Regression;

Logistic regression is a statistical method that is used for building machine learning models where the dependent variable is dichotomous: i.e. binary. Logistic regression is used to describe data and the relationship between one dependent variable and one or more independent variables. The independent variables can be nominal, ordinal, or of interval type.

Random Forest;

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the dataset is huge I planned to fit Classification models Logistic Regression and Random Forest keeping the following assumptions which generally assumed for both models

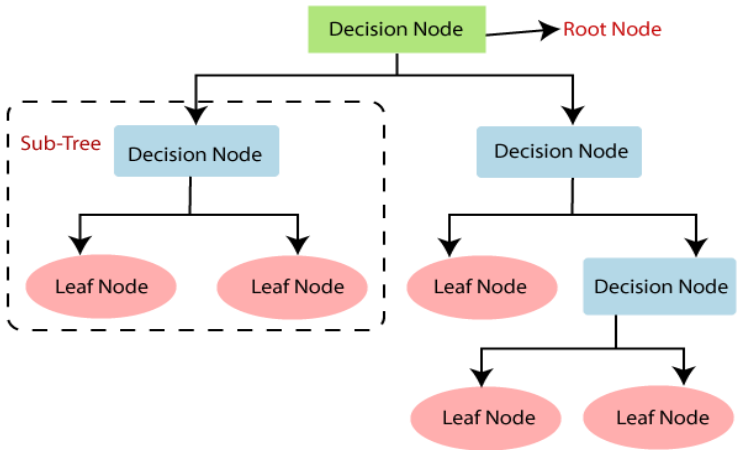
- Assume the observations to be independent of each other
- There is minimal or no multicollinearity
- Assumption of no formal distributions. Being a non-parametric model, it can handle skewed and multi-modal data.

Decision Tree;

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a

dataset, branches represent the decision rules and each leaf node represents the outcome.

The graph below represents the working of decision tree,



Results

First Way of Approach:

Model	Accuracy
Logistic Regression	73.04818
Random forest	73.059

Second Way of Approach

Model	Accuracy
Decision Tree	85.986
Logistic Regression	86.20
Random forest	86.103

Challenges and Opportunities

The dataset considered contains 0.36 million rows and 12 columns may take time in cleaning and processing and also mostly the data is in text format so needed application of text processing.

Deciding the target variable become a crisis as it was in text format and contained unique data.

As the dataset is of large size choosing a classification model is tough task Logistic Regression and Random forest will be more appropriate other models such as SVM and K-Nearest Neighborhood and Support Vector Machine may not work in this.

But if a model with more than 80 percent accuracy will classify the drugs based on whether the customer is satisfied or not as it is taken by keeping the assumption that more satisfied less side effects it helps people in knowing about the drugs and side effects.

Conclusion

As I conclude this project I was able to approach the problem assigned tome in two different way and build models two different ways. As the second approach which considered only the numerical columns gave more than 80 percent accuracy in three different models like Logistic Regression, Decision Tree and Random Forest that approach can be considered more convenient and effective in building further applications. Of all three models Logistic regression gave the maximum accuracy in the second approach and can be considered as the best model.

To sum up trying out different approaches and fitting different models helped me understand more about the topic and overall this internship become first step towards the career as a Data Analyst.

Link To Code;

The links of the three notebooks are given below;

https://colab.research.google.com/drive/1-AJbFuQudqnkZoOqgx_HE9HwWQCcvFbL?usp=sharing

https://colab.research.google.com/drive/1Yd7mnWqSmsZFBr7qvKtUw02PzAViVqzx?usp=sharing

https://colab.research.google.com/drive/1sTjxvqx6LaRfDtvpZDmqLBJ-oT8lgvdY?usp=sharing