Automated System for prescription of drugs using machine learning

Dr.M.Malleswari ME,Ph.D [1], Selvakumar p[2]

[1] **Head of the department , Department of Information Technology, Veltech High Tech Dr RR Dr SR Engineering college . Email: drmalleswari79@gmail.com**

[2]**Department of Information Technology, Veltech High Tech Dr RR Dr SR Engineering college Email:selvatopper8@gmail.com**

Abstract- This project aims at providing the prescription of suitable drugs for the patient with particular condition. This is achieved by the use of machine learning algorithms to develop a Decision Support System which works by mimicking physician's intuition on analyzing the disease. The dataset involves symptoms experienced by a person, disease affected and Drugs prescribed which would be used for training a model based on the given Information the system will identify the type of Disease and suggest a medicinal drug.

Keywords- Data Modelling, Machine Learning, Random Forest Classifier, Support Vector Classifier,K-Neighbors Classifier

I. INTRODUCTION

Health is one of the major aspect which defines aesthetic needs of our life. In this fast moving world, where the nutrition of the people and working hours have changed drastically in the last two decades, we have seen dreadful impact of new complicated diseases, illness and other problems affecting the general working population and their lifestyle. People are also inclined to quick cures with comes along with different side-effects, instead of looking at the longer run. Time and money has played a diverse role in shrinking the diagnosis period and also it has been distancing the quality of diagnosis from the people by assigning a weight of higher cost since industrialization and rapid development came into picture.

Expert and immediate guidance of doctors is far beyond reach if the patients do not have enough money or time to get themselves up in the mid-night at the end of the month and start walking towards a global hospital.

Thus the probability of people going for second opinion and diagnosis is increasing exponentially these days. This leads to objectifying Patient's body like guinea-pig where different medicines and drugs to find out the right cure. Now with everything in the form of a Mobile App starting from ordering food till finding your life partner and along with age of artificial intelligence and machine learning, It is possible to mimic the diagnosis of expert doctor and create the first-aid for the people to trust and follow in case of emergency situations and as well as in situations where people do not trust medical services available in their locality.

II. DATA ANALYSIS AND MODELLING

The data collected was briefly described and talks about diagnosis of disease. It will also explain the different types of drugs that was prescribed. The data was analyzed in such a way that helps to build a suitable machine learning model. Various packages are used in python for the graphical representation of data

A. Physician's Intuition in the prescription of Drugs

The objective was to provide a prescription of drugs that uses machine learning algorithms that work partially (or) similar to a doctor's intuition in diagnosing a patient. The discussion with experts provides following inference which they use to handle to prescribe drugs to the patient.

1. The symptoms are associated with particular disease.

2. Initial diagnosis is completely based on the symptoms and the patient was questioned for the same.

3. The drug prescription varies based on the particular types such as vitamin suppliment for vision problems and aldosterone inhibitors for cardiac problem.

B. Symptoms listing: To identify the kind of drug to be prescribed various symptoms suggested are:

Optical symptoms:

Vision problems : the most common vision problems are refractive errors.

Cardiac symptoms*:*

Chest Discomfort : It's the most common sign heart danger.

Abdominal discomfort: feel of discomfort in digestion, dizziness.

Obesity Symptoms:

Increased fat content : (Increased BMI rate)

Lack of physical activity: Inability to cope up with sudden activity.

Common Cold Symptoms*:*

Running nose : Accompanied by sneezing and congestion.

Mild fever : Slight increase in body temperature.

Ortho Symptoms:

Crunching feeling : sound of bone rubbing on bone.

Stiffness in bone : Difficulty in moving bone.

Fever Symptoms:

Temperature Increase: Increase in body temperature than normal .

The above mentioned symptoms are taken in an account for the Initial diagnosis of disease and moving for further diagnosis .Later intrinsic interrogation with the patient is done.

C. Types of Drugs

According to the Inference and discussion with the experts the Inference comes out as follows: Drugs are broadly classified in to several types some of them are,

Vitamin Supplement:

Vitamin A plays vital role in vision by maintaining a clear cornea.

Most important vitamins for Eye health (vitamin A, vitamin E, vitamin c

Vitamin B6,B9 and B12, Riboflavin, Niacin, Luttein and Zeaxanthin.

Contact lens are also prescribed.

Ace Inhibitors:

An angiotensin-converting-enzyme inhibitor is a pharmaceutical drug used primarily for the treatment of hypertension and congestive heart failure. Examples of Ace Inhibitors include

Accupril (quinapril), Aceon (perindopril),

Altace (ramipril), capoten (captopril),etc.

Aldosterone Inhibitors:

An Aldosterone antagonist , is a diuretic drug which antagonize the action of aldosterone at mineralocorticoid receptors. This group of drug is often used as adjunctive therapy in combination with other drugs for chronic heart failure.

Buproppion/Naltrexone:

It's a combination drug used for weight loss in those that are either obese (or) over weight with some-weight related illnesses.

Corticosteroids and Antidepressant: Both were used for ortho problems.

Paracetamol:

It is also known as acetaminophen and APAP, is a medication used to treat pain and fever.

Ibuprofen:

It is a non-steroidal and anti-inflammatory drug class used for treating pain and fever.

D. Data overview

The data set was collected and It contains the record of 890 patients with different Symptoms, Disease affected, and drugs category prescribed by the physician. It is in the form of".CSV " used to train the model and to build a decision support system.

E. Data Preprocessing

The process involves transformation of raw data in to understandable format. The training of machine learning algorithm requires a data in certain format hence data should be preprocessed .Data preprocessing involves

These are the following major types of processing:

• **Cleaning**: Removing outliers, filling missing NA values

• **Transformation**: Normalising or standard scaling the data

• **Reduction**: Binning, dummy variable reduction.

The following concepts of data preprocessing will be used to this dataset ready for modelling:

• **Dummy variables**: Data Transformation:

The process of splitting the variables with multiple categories into individual mutually exclusive features to ensure the smoothness of the model performance.

E.g.: Cough has 3 categories, The Feature Cough is transformed into Dry Cough, Sore Throat and Cough with Sputum as Individual columns (Features) Representing (1/0) scenario.

• **Encoding**: Data Reduction:

This is done to convert a string representation of (Yes/No) to 1/0 for features like Allergy, Urinary Infection etc.

The following are the snippets of the data preprocessing done in Python in the Jupyter Notebook.

```
In [194]: import pandas as pd

In [195]: train = pd.read_csv("patient train3.csv")

In [196]: train.head()
```

```
In [3]: train = pd.read_csv("patient train2.csv")

In [4]: train.head()

Out[4]:
```

	Patient ID	Name	Sex	Age	Disease	Symptom1	Symptom2	Prescription	Prescription2
0	1.0	Braund, Mr. Owen Harris	male	22.0	optical	Vision problems	myopia or hypermetropia	contact lens	vitamin supplement
1	2.0	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	Cardiac	Chest Discomfort	Heart burn	Ace inhibitors	Aldosterone inhibitors
2	3.0	Heikkinen, Miss. Laina	female	26.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
3	4.0	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
4	5.0	Allen, Mr. William Henry	male	35.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone

Fig : Loading CSV file.

```
In [5]: train.isnull()
```

Out[5]:

	Patient ID	Name	Sex	Age	Disease	Symptom1	Symptom2	Prescription
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
5	False	False	False	False	False	False	False	False
6	False	False	False	False	False	False	False	False
7	False	False	False	False	False	False	False	False
8	False	False	False	False	False	False	False	False
9	False	False	False	False	False	False	False	False
10	False	False	False	False	False	False	False	False
11	False	False	False	False	False	False	False	False
12	False	False	False	False	False	False	False	False
13	False	False	False	False	False	False	False	False
14	False	False	False	False	False	False	False	False
15	False	False	False	False	False	False	False	False
16	False	False	False	False	False	False	False	False
17	False	False	False	False	False	False	False	False

Fig : Checking for null values.

```
In [6]: train['Sex'] = train['Sex'].map({'male':1,'female':2})
```

```
In [7]: train
```

1	2.0	Cumings, Mrs. John Bradley (Florence Briggs Th...	2.0	38.0
2	3.0	Heikkinen, Miss. Laina	2.0	26.0
3	4.0	Futrelle, Mrs. Jacques Heath (Lily May Peel)	2.0	35.0
4	5.0	Allen, Mr. William Henry	1.0	35.0
5	6.0	Moran, Mr. James	1.0	45.0
6	7.0	McCarthy, Mr. Timothy J	1.0	54.0
7	8.0	Palsson, Master. Gosta Leonard	1.0	2.0
8	9.0	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	2.0	27.0
9	10.0	Nasser, Mrs. Nicholas (Adele Achem)	2.0	14.0

Fig : Gender Labelling.

Out[61]:

	Patient ID	Sex	Age	Disease	Symptom1	Symptom2	Prescription
0	1.0	1.0	22.0	5.0	6.0	5.0	0.0
1	2.0	2.0	38.0	1.0	1.0	2.0	1.0
2	3.0	2.0	26.0	4.0	2.0	3.0	2.0
3	4.0	2.0	35.0	4.0	2.0	3.0	2.0
4	5.0	1.0	35.0	4.0	2.0	3.0	2.0
5	6.0	1.0	45.0	1.0	1.0	2.0	1.0
6	7.0	1.0	54.0	6.0	3.0	1.0	6.0
7	8.0	1.0	2.0	2.0	4.0	4.0	5.0
8	9.0	2.0	27.0	4.0	2.0	3.0	2.0
9	10.0	2.0	14.0	3.0	5.0	0.0	4.0
10	11.0	2.0	4.0	2.0	4.0	4.0	5.0
11	12.0	2.0	58.0	6.0	3.0	1.0	6.0
12	13.0	1.0	20.0	5.0	6.0	5.0	0.0
13	14.0	1.0	39.0	1.0	1.0	2.0	1.0
14	15.0	2.0	14.0	3.0	5.0	0.0	4.0
15	16.0	2.0	55.0	6.0	3.0	1.0	6.0
16	17.0	1.0	2.0	2.0	4.0	4.0	5.0
17	18.0	1.0	45.0	1.0	1.0	2.0	1.0
18	19.0	2.0	31.0	4.0	2.0	3.0	2.0

Fig : Train Data after Labelling .

```
In [23]: train.head(5)
```

Out[23]:

	Patient ID	Sex	Age	Disease	Symptom1	Symptom2	Prescription	Prescription2
0	1.0	1.0	22.0	5.0	6.0	5.0	0.0	0.0
1	2.0	2.0	38.0	1.0	1.0	2.0	1.0	1.0
2	3.0	2.0	26.0	4.0	2.0	3.0	2.0	2.0
3	4.0	2.0	35.0	4.0	2.0	3.0	2.0	2.0
4	5.0	1.0	35.0	4.0	2.0	3.0	2.0	2.0

Fig : Labelled Data

```
In [10]: train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 892 entries, 0 to 891
Data columns (total 8 columns):
Patient ID       891 non-null float64
Sex              891 non-null float64
Age              891 non-null float64
Disease          891 non-null object
Symptom1         891 non-null object
Symptom2         891 non-null object
Prescription     891 non-null object
Prescription2    891 non-null object
dtypes: float64(3), object(5)
memory usage: 38.4+ KB
```

F. Exploratory Data Analysis

Exploratory Data Analysis (EDA) is done in order understand the distribution, nature and patterns present in the given data. It is the most important step in the analytics due to the fact it gives the idea about the further data transformation that has to be done for the machine learning algorithms give high results.

```
In [26]: import matplotlib.pyplot as plt
         %matplotlib inline
         import seaborn as sns
         sns.set() # setting seaborn default for plots
```

Fig : Matplot Library import

```
In [27]: def bar_chart(feature):
             male = train[train['Sex']==1][feature].value_count
             female = train[train['Sex']==2][feature].value_cou
             df = pd.DataFrame([male,female])
             df.index = ['male','female']
             df.plot(kind='bar',stacked=True, figsize=(10,5))
```

Fig : Barchart feature

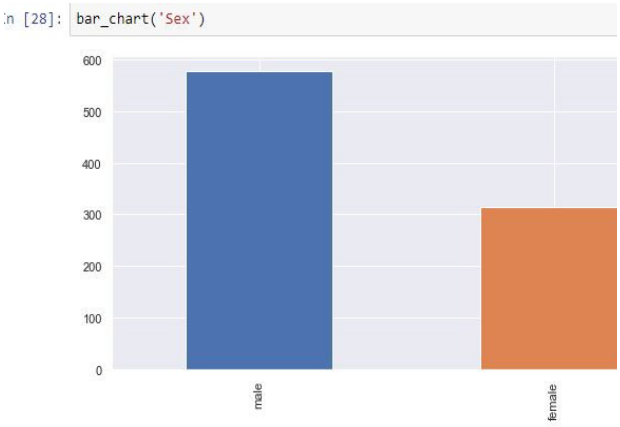


Fig : Barchart of Gender.

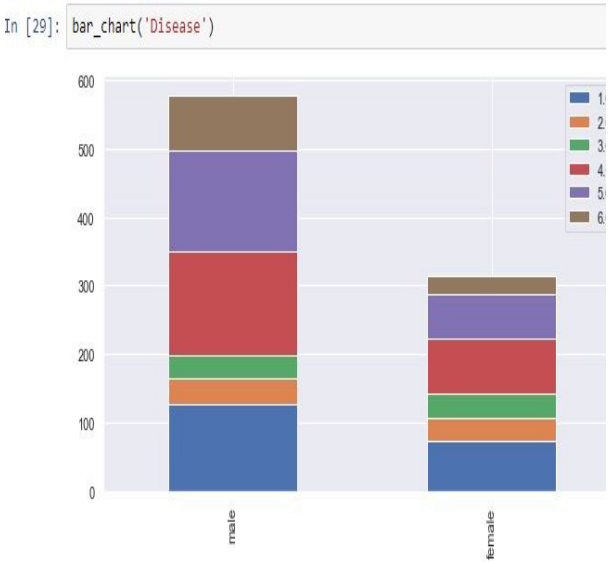
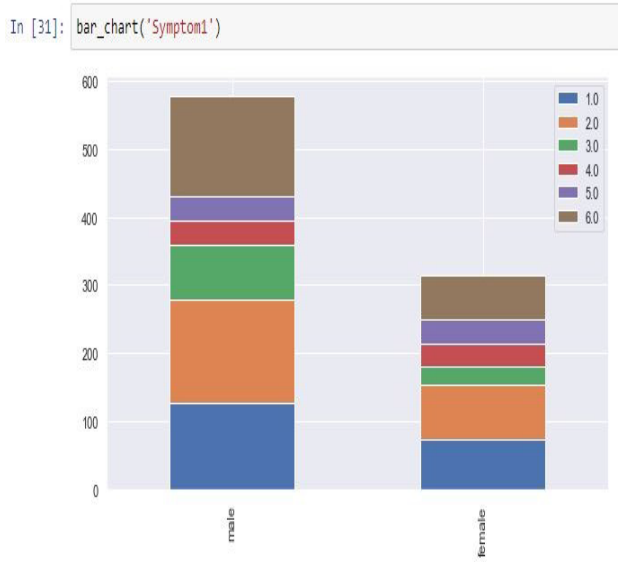


Fig : Barchart of Disease affected.



```
In [32]: bar_chart('Symptom2')
```

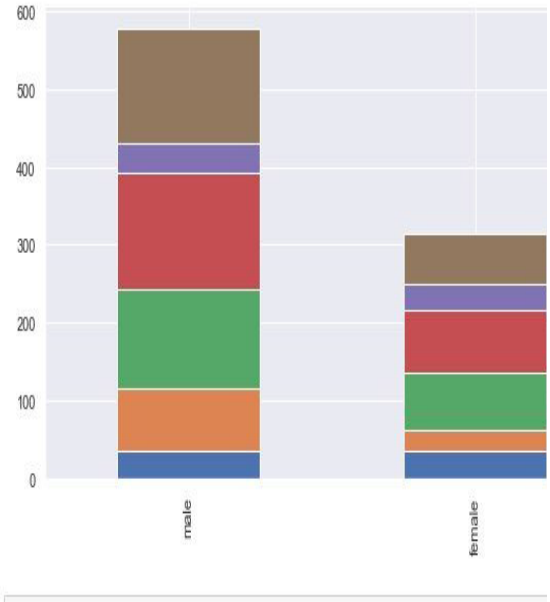


Fig : Barchart of symptoms.

```
In [33]: bar_chart('Prescription')
```

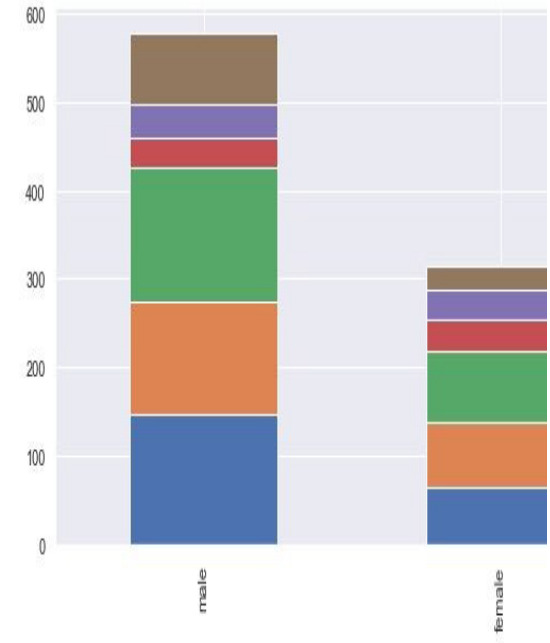
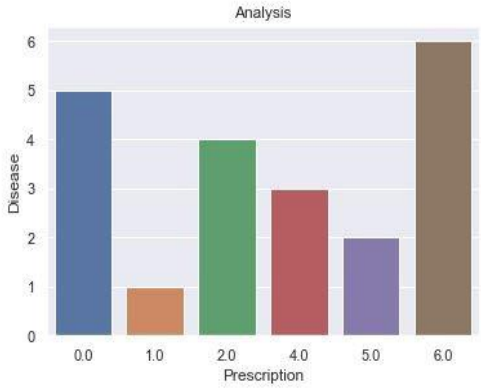


Fig : Barchart of Prescriptions

```
In [36]: sns.barplot(train["Prescription"],train["Disease"])
         plt.title("Analysis")
Out[36]: Text(0.5, 1.0, 'Analysis')
```




```
In [37]: sns.barplot(train["Prescription2"],train["Disease"])
         plt.title("Analysis")
Out[37]: Text(0.5, 1.0, 'Analysis')
```

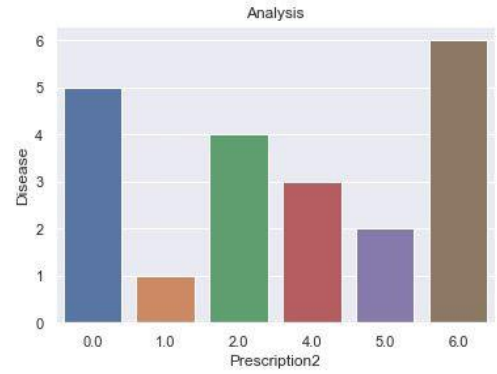


Fig : Barplot

```
In [35]: plt.scatter(train['Age'],train['Disease'],color = 'blue')
         plt.title("Age vs Disease")
         plt.xlabel("Age")
         plt.ylabel("Disease")
         plt.show()
```

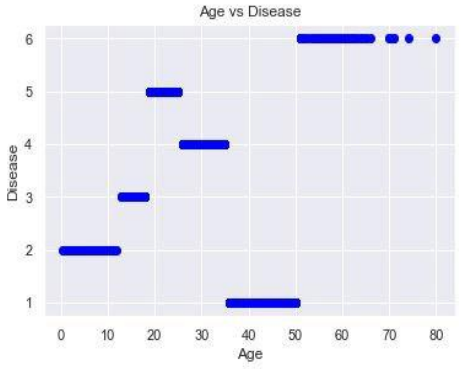


Fig : Scatter plot

Preparation of Train and Test data

```
In [39]: # Importing Classifier Modules
         from sklearn.neighbors import KNeighborsClassifier
         from sklearn.tree import DecisionTreeClassifier
         from sklearn.ensemble import RandomForestClassifier
         from sklearn.naive_bayes import GaussianNB
         from sklearn.svm import SVC

         import numpy as np

In [40]: from sklearn.model_selection import KFold
         from sklearn.model_selection import cross_val_score
         k_fold = KFold(n_splits=10, shuffle=True, random_state=0)

In [41]: # import dependencies
         import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         import seaborn as sns

         # other dependencies that you might not need
         # just for publishing image in notebook
         from IPython.display import Image
         from IPython.core.display import HTML
         from sklearn import datasets

In [42]: target = train['Prescription2']

In [43]: target

Out[43]: 0      0.0
         1      1.0
         2      2.0
         3      2.0
         4      2.0
         5      1.0
         6      6.0
         7      5.0
         8      2.0
         9      4.0
         10     5.0
         11     6.0
```

Fig : Target to predict

Accuracy of different models

```
In [45]: clf = KNeighborsClassifier(n_neighbors = 13)
         scoring = 'accuracy'
         score = cross_val_score(clf, train, target, cv=k_fold, n_jobs=1, scori
         print(score)

         [0.54444444 0.59550562 0.51685393 0.61797753 0.5505618  0.65168539
          0.59550562 0.53932584 0.53932584 0.59550562]

In [46]: # kNN Score
         round(np.mean(score)*100, 2)

Out[46]: 57.47

In [47]: clf = SVC()
         scoring = 'accuracy'
         score = cross_val_score(clf, train, target, cv=k_fold, n_jobs=1, scori
         print(score)

         [0.62222222 0.56179775 0.56179775 0.5505618  0.61797753 0.58426966
          0.58426966 0.51685393 0.65168539 0.57303371]

In [48]: round(np.mean(score)*100,2)

Out[48]: 58.24

In [49]: clf = RandomForestClassifier(n_estimators=13)
         scoring = 'accuracy'
         score = cross_val_score(clf, train, target, cv=k_fold, n_jobs=1, scori
         print(score)

         [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]

In [50]:
         # Random Forest Score
         round(np.mean(score)*100, 2)

Out[50]: 100.0
```

III. MACHINE LEARNING MODELS AND VALIDATION METHODS

The objective is to predict the drug category using Machine Learning Algorithm .The models used must be able to work like the intuition of the doctors diagnosing the patients.

The following methodology was used to do data modelling and validate Machine Learning.

1. Define a clear set evaluation metrics which are suitable for class problems.

2. The machine learning models to be used Random forest Classifier, Support vector classifier, K-Neighbors Classifier, Decision tree classifier and Bayesian classifier.

3. Validate the results with K-FOLD cross validation.

4. Select the best performing model it product ready for deployment

A. Machine Learning Models:

Random Forest:

Random Forest is a supervised algorithm which builds weak learners aggregates them to form a strong learner. This method of learning is called Ensemble learning. It uses aggregating known as “Bootstrap Aggregating or Bagging”.

Bagging or Bootstrap Aggregating:

The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners.

Given a training set X = x1, ..., xn with responses.

Y = y1, ..., yn, bagging repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these

samples: For b = 1,..., B: Sample, with re- placement, n training examples from X, Y; call these Xb, Yb. Train a classification or regression tree fb on Xb, Yb.

After training, predictions for unseen samples x’ can be made by averaging the predictions from all the

individual regression trees on x' or by taking the majority vote in the case of classification trees.

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x')$$

Node Splitting Criterion- Information Gain:

Information gain is used to decide which feature to split on at each step in building the tree. Simplicity is best, so we want to keep our tree small. To do so, at each step we should choose the split that results in the purest daughter nodes.

For each node of the tree, the information value "represents the expected amount of information that would be needed to specify whether a new instance should be classified yes or no, given that the example reached that node."

$$\overbrace{IG(T,a)}^{\text{Information Gain}} = \overbrace{H(T)}^{\text{Entropy(parent)}} - \overbrace{H(T|a)}^{\text{Weighted Sum of Entropy}}$$

Where Entropy is given as:

$$H(T) = I_E(p_1, p_2, \ldots, p_J) = -\sum_{i=1}^{J} p_i \log$$

where p1, p2,.. are fractions that add up to 1 and represent the percentage of each class present in the child node that results from a split in the tree.

The Random Forest functions in two stages:

Stage 1: Classifier Building

The algorithm randomly selects "N" Features from total features available in the dataset Among the the selected features, the node split is done (Information gain: Entropy split) Multiple trees are created till maximum split is reached. The forest is built by aggregating all the built trees.

Stage 2: Prediction using Majority Voting

Each tree predicts the outcome class using the test feature vectors. The algorithm uses votes for each predicted target and the majority voted target is the final prediction that is returned by the algorithm.

Advantages of using Random Forest:

• Avoids overfitting in classification problems in machine learning due to the presence of optimal Decision tress

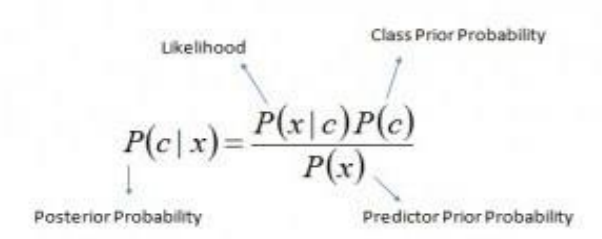
• It also calculates the Feature importance of each feature during training

• Can be scalable and deployed easily.

Bernoulli's Naive Bayes (NB) classifier :

This is a supervised machine learning algorithm which is built over Bayesian probability.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

• P(c|x) is the posterior probability of class (c, target) given predictor (x,attributes).

• P(c) is the prior probability of class.

• P(x|c) is the likelihood which is the probability of predictor given class.

• P(x) is the prior probability of predictor.

Advantages:

• Suitable for large dataset

• Suitable for multi class scenarios

• Performs well if most of the features are categorical variables

Disadvantages:

• The algorithm assumes all the given feature vectors are independent which is not possible in real-life. There is always dependencies.

• If a category is not in the test data or a pattern or observation the model will assign zero to the posterior probability calculated. The model will be unable to make a prediction. This phenomenon is known as "Zero Frequency error".

Bernoulli NB implements algorithms for the data whose distributions of the feature vectors are according to the multivariate Bernoulli Distributions i.e. each feature must be represented in binary. The decision rule for Bernoulli naive Bayes is based on:

$$P(x_i \mid y) = P(i \mid y)x_i + (1 - P(i \mid y))(1 - x_i$$

B. *Validation Metrics:*

Validation is the most important setup in data modelling to measure the performance of the model. It is also important to look in to different metrics to gain better insights of the performance of the model. The following metrics where looked into to understand how the above mentioned models performed in the test dataset.

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Confusion Matrix:

A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and bro- ken down by each class. This is the key to the confusion matrix.

True positives (TP): These are cases in which the model predicted yes and the actual value is yes.

True negatives (TN): The model predicted No and the actual value is No. False positives (FP): The model predicted yes but they actual value is No. False negatives (FN): The model predicted no, but the actually value is Yes.

Accuracy:

Overall, how often is the classifier correct?

(TP+TN)/total = (100+50)/165 = 0.91 Accuracy gives the overall performance of the algorithm, which will act as the base of comparison between multiple models.

Precision:

When it predicts yes, how often is it correct?

TP/predicted yes = 100/110 = 0.91 This metric gives us the percentage of True positives from the Total number of classifications made as positive. The less Precision, the more False positives and in the field of health care the threshold of False positives must be less than 10% in accordance with the Risk factors.

Recall:

Also known as True Positive Rate / Sensitivity . When it's actually yes, how often does it predict yes?

TP/actual yes = 100/105 = 0.95 This metric gives us the insight of the how much percentage of the total positive classes is predicted by the model.

The results of the above mentioned algorithms are explained in detail the Results and Analysis Section.

C . Model Deployment Methods

• After the identification of the best machine learning model which is production ready both in terms of deployment and Doctor‘s intuition, the model will be trained in the whole dataset.

• The trained model can be store as a serialized file using the Python Package “Pickle”

• The Serialized object file stored in the form of “.sav” extension can be loaded into the web application (Views Layer) and can be used for prediction .

```
In [271]: import pickle
          import requests
          import json

In [272]: mechmodel = 'submission.sav'

In [273]: from sklearn.externals import joblib
          joblib.dump(clf, 'mechmodel.pkl')
Out[273]: ['mechmodel.pkl']

In [274]: clf = joblib.load('mechmodel.pkl')

In [275]: model_columns = list(train.columns)
          joblib.dump(model_columns, 'mechmodel_columns.pkl'
          print("mechmodels columns dumped!")
          mechmodels columns dumped!

In [276]: test
Out[276]:
```

	Patient ID	Sex	Age	Disease	Symptom1	Symptom2	Pre
0	12	1	49	4	6	3	

```
In [277]: filename = 'submission.sav'

In [278]: filename

Out[278]: 'submission.sav'
```

Fig : Pickling the trained model.

IV . DJANGO CODE TO PROMPT INPUT FROM PATIENT

As the part of the questionnaire, We need to build a form for getting the User Data(i.e. Symptoms of the Patient). So using framework‘s form class we defined the our Form class for input. Here is a excerpt of the Code.

```
from django import forms



class hello(forms.Form):
    PatientID = forms.IntegerField()
    Sex =  forms.ChoiceField(widget=forms.Select, choices=((1, 'male'),
                                                            (2, 'female'),))

    Age = forms.IntegerField()


    Disease = forms.ChoiceField(widget=forms.Select,
                        choices= ((1, 'Cardiac'),
                                  (2, 'cold'),
                                  (3, 'fever'),
                                  (4, 'obesity'),
                                  (5, 'optical'),
                                  (6, 'Ortho'),))



    Symptom1 = forms.ChoiceField( widget=forms.Select, choices=((1, 'chest Discomfort'),
                                                                (2, 'fat'),
                                                                (3, 'Joint Stiffnesss'),
                                                                (4, 'Running nose'),
                                                                (4, 'Temperature increase'),
                                                                (5, 'Vision problems'),))
```

```
Symptom2 = forms.ChoiceField( widget=forms.Select, choices=((1, 'Crunching feelir
                                                             (2, 'Heart burn'),
                                                             (3, 'lack of physical
                                                             (4, 'mild fever'),
                                                             (5, 'myopia or hyperm
                                                             (0, 'nil'),))

Prescription = forms.ChoiceField( widget=forms.Select, choices=((1, 'Ace inhibitc
                                                                 (2, 'Physical act
                                                                 (3, 'contact lens
                                                                 (4, 'Paracetamol'
                                                                 (5, 'ibuprofen'),
                                                                 (6, 'Antidepressa
```

```
def output(request):

    form2 = hello(request.POST)
    if form2.is_valid():
        PatientID = form2.cleaned_data['PatientID']
        Sex = form2.cleaned_data['Sex']
        Age = form2.cleaned_data['Age']
        Disease = form2.cleaned_data['Disease']
        Symptom1 = form2.cleaned_data['Symptom1']
        Symptom2 = form2.cleaned_data['Symptom2']
        Prescription = form2.cleaned_data['Prescription']

        ml = Machine_Learning()
        ml.setValues(PatientID,Sex,Age,Symptom1,Symptom2,Prescription,Disease)
        ml.predictvalues()

    return render(request, 'prescrib/output.html',{'Prescription2':Prescription2})
```

Fig : views in prescrip(app)

We have a Views.py module in python which defines functions which render the appropriate View layer objects(web pages). We use this to handle requests which are dispatched by the URL dispatcher.

Here we are handling the View layer code for the Prediction App.

```
from django.urls import reverse
from django.http import HttpResponseRedirect
from django.shortcuts import render,HttpResponse
from django.views.generic.edit import CreateView, UpdateView,
from prescrib.predictions.pred import *
from .forms import hello
from .ML import Machine_Learning

def index(request):
    return HttpResponse("Hi there Dont worry u ill finish ur
def service(request):
    return render(request, 'prescrib/deletenow1.html')


def prescrib(request):
    form = hello()
    return render(request, 'prescrib/trydelete.html', {'form'


def output(request):

        form2 = hello(request.POST)
        if form2.is_valid():
            PatientID = form2.cleaned_data['PatientID']
            Sex = form2.cleaned_data['Sex']
            Age = form2.cleaned_data['Age']
            Disease = form2.cleaned_data['Disease']
            Symptom1 = form2.cleaned_data['Symptom1']
            Symptom2 = form2.cleaned_data['Symptom2']
```

URLs are handled by a dispatcher module. They reference the Views.py file and call the functions which need to be referenced to render appropriate web page. The url patterns is a list of url objects which maps patterns using regex code.

```
1  from django.urls import path
2
3  from . import views
4  urlpatterns = [
5
6          path('', views.index, name='index'),
7          path('service/', views.service, name='service'),
8          path('prescrib/', views.prescrib, name='prescrib'),
9          path('output/', views.output, name='output'),
0
1      ]
```

Fig: prescrib/urls.py

To store content in database, we configure the models.py File. This defines classes which can create the appropriate records in the database. Database can be configured using settings.py file in Django Admin App. We use the following Models.py file for Prediction data.

```python
from django.db import models

# Create your models here.

class Prescription2(models.Model):
    Prescription2 = models.CharField(max_length=140)
```

Fig : Models for Prescrip (app)

```python
import numpy as np
from sklearn.externals import joblib
import pickle
import os
script_dir = os.path.dirname(__file__)

def prescrib(PatientID,Sex,Age,Disease,Symptom1,Symptom2,Presc
    rel_path = "pres/mechmodel.pkl"
    abs_file_path = os.path.join(script_dir, rel_path)
    regressor = joblib.load(abs_file_path)

    rel_path = "pres/mechmodel.pkl"
    abs_file_path = os.path.join(script_dir, rel_path)
    enc = pickle.load( open( abs_file_path, "rb" ))

    X_test = [[PatientID,Sex,Age,Disease,Symptom1,Symptom2,Pre
    X_test = enc.transform(X_test).toarray()
    print(len(X_test[0]))
    y_pred = regressor.predict(X_test)
    print("our_prediction:",y_pred)
    return y_pred[0]
```

Fig : Prediction using symptoms (pred.py)

We can have custom python modules which render us help with some algorithms to handle business logic. They can be integrated easily with other django modules as the base language is Python. We have to created certain modules like for determining Prescriptions as well as for processing the Machine Learning Model from a pickle object .

If all works well, the server runs successfully and the application goes live.

```
You have 15 unapplied migration(s). Your project may not work properly until you apply the migrations for

Run 'python manage.py migrate' to apply them.

March 21, 2019 - 12:16:11

Django version 2.1.7, using settings 'prescription2.settings'

Starting development server at http://127.0.0.1:8000/

Quit the server with CTRL-BREAK.
```

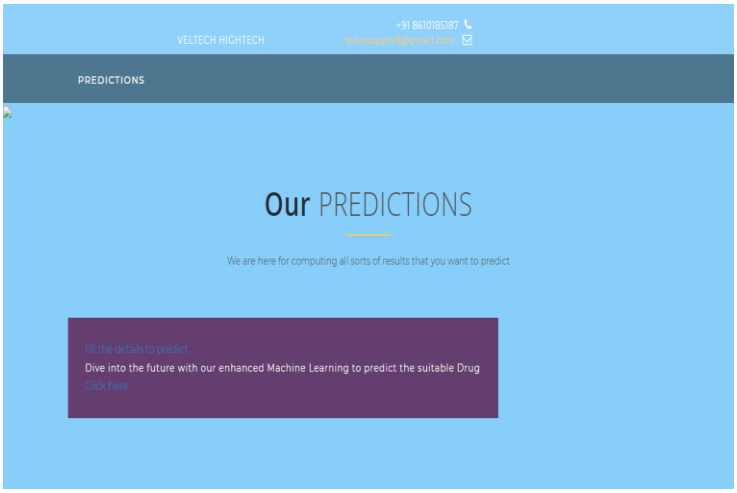
Fig : Django server running successfully

V . TESTING OF MODEL

Testing is an integral part of any development activity. Testing exposes any fault in the system and helps to continuously improve the performance and security of the application. Unit testing performed in each and every stage and any unexpected outputs are corrected then and there to develop a stable system.

Home-Page

Launching the web-app on the browser we get to see the homepage.The application shows the basic functionalities.“Click here” will enable user to launch the Prescrib app



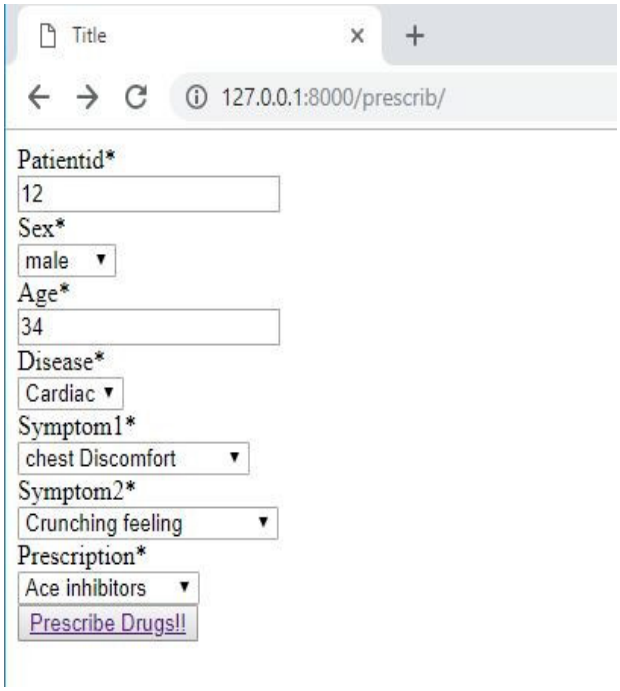


Fig : Form filling

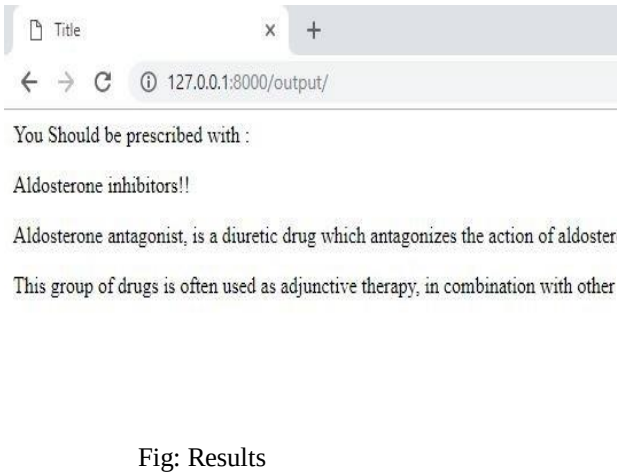


Fig: Results

Here the app prescribed Aldosterone Inhibitors .Which as successfully predicted based on the symptom provided.

V. RESULTS AND ANALYSIS

The following are the results of the models applied over the dataset and validated over the previously mentioned metrics.

KNEIGHBORS CLASSIFIER:

```
[45]: clf = KNeighborsClassifier(n_neighbors = 13)
      scoring = 'accuracy'
      score = cross_val_score(clf, train, target, cv=k_fold, n_jobs=1, scoring=scoring)
      print(score)

      [0.54444444 0.59550562 0.51685393 0.61797753 0.5505618  0.65168539
       0.59550562 0.53932584 0.53932584 0.59550562]
```

SUPPORT VECTOR CLASSIFIER:

```
In [47]: clf = SVC()
         scoring = 'accuracy'
         score = cross_val_score(clf, train, target, cv=k_fold, n_jobs=1, scoring=scoring)
         print(score)

         [0.62222222 0.56179775 0.56179775 0.5505618  0.61797753 0.58426966
          0.58426966 0.51685393 0.65168539 0.57303371]

In [48]: round(np.mean(score)*100,2)

Out[48]: 58.24
```

RANDOM FOREST CLASSIFIER:

```
In [76]: RandomForestClassifier(n_estimators=13)

Out[76]: RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                                max_depth=None, max_features='auto', max_leaf_nodes=None,
                                min_impurity_decrease=0.0, min_impurity_split=None,
                                min_samples_leaf=1, min_samples_split=2,
                                min_weight_fraction_leaf=0.0, n_estimators=13, n_jobs=None,
                                oob_score=False, random_state=None, verbose=0,
                                warm_start=False)
```

```
In [49]: clf = RandomForestClassifier(n_estimators=13)
         scoring = 'accuracy'
         score = cross_val_score(clf, train, target, cv=k_fold, n_jobs=1, scoring=scoring)
         print(score)

         [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]

In [50]:
         # Random Forest Score
         round(np.mean(score)*100, 2)
```

Accuracy of different machine learning models varies in such a way they are trained. K-Neighbors classifier and Support vector classifier seems to give average accuracy when compared to Random Forest classifier. Hence by choosing better model we could get accurate Decision Support Systems

VI . CONCLUSION AND FUTURE ENHANCEMENTS

Health analytics is one the most important and useful applications of analytics. With new age technologies such as Artificial Intelligence, machine learning and Distributed Application development, the industry is able to create highly efficient support systems in giving accurate diagnosis to the patients.

So accordingly we developed a user i.e. patient, personalized Health Consultation service that help identifies the type of disease from which they are suffering and also prescribes medicines to reduce the effects of the symptoms faced by the respective patients.

The system receives inputs of the symptoms from the users via a front end. The list of symptoms is sent to the back-end framework developed in Django to handle multiple requests and sessions of the users. Once the request is received by the server at the Application Programming Interface (API) endpoint, it predicts the type of fever with the help of trained machine learning algorithms and returns back the response along with list of prescriptions of medicine. The whole process of the First health consultation is automated for the users to access at any part of the day or night.

The current machine learning model uses a Random forest classifier to prescribe the type of drug with a given set of symptoms. In order to combine information gain, rule based mappings and probability of recent cases in a locality or an area, a hybrid ensemble model which predicts the type of drug using Majority voting between Random Forest, Apriori algorithms and Naive Bayes classifier can be built and dynamic training within intervals of seasons can be automated. The web-application runs on single server - multi-client model. The architecture of the model will be scaled up to run as a multi-server-multi-client model_using distributed application frameworks such as Redis, where recent and most frequent data can be cached, making the application perform faster. The system will also be able to track past prescriptions of patients and maintain Electronic health records and appointments taken by the user.

REFERENCES

1. Wang, T., Zhao, L., Cao, Y.F., Qu, Z.J. and Li, P.J. (2018) Medical Data Visualization Analysis and Processing Based on Machine Learning. Journal of Computer and Communications, 6, 299- 310.

https://doi.org/10.4236/jcc.2018.611027

2. Fatima, M. and Pasha, M. (2017) Survey of Machine Learning Algorithms for Disease Diagnostic. Journal of Intelligent Learning Systems and Applications, 9, 1-16.

https://doi.org/10.4236/jilsa.2017.91001

3. Dereck L. Hunt, MD; R. Brian Haynes, MD, PhD; Steven E. Hanna, MA, PhD; et al JAMA. 1998;280(15):1339-1346. doi:10.1001/jama.280.15.1339.

4.Safdari, R. , Farzi, J. , Ghazisaeidi, M. , Mirzaee, M. and Goodini, A. (2013) Healthcare intelligence risk detection systems. Open Journal of Preventive Medicine, 3, 461-469.

5. Lee, S. and Kim, E. (2015) Public Health Nurse Perceptions of Omaha System Data Visualization. International Journal of Medical Informatics, 84, 826-834.S

AUTHORS

First Author- selvakumar p, BTECH-Information Technology, Veltech High Tech Dr RR Dr SR Engineering college, selvatopper8@gmail.com

Project Guide- Dr.M.Malleswari ME.,Ph.D, Head of the Department, BTECH-Information Technology, Veltech High Tech Dr RR Dr SR engineering college, drmalleswari79@gmail.com