Design and Development of Disease Prediction

SUBMITTED BY:

AKSHITA AGARWAL (1601410012)
LAVI THUKRAL (1601410060)
PRIYANSHI KOTNALA (1601410081)

PROJECT GUIDE:

DR. L.S MAURYA
DEPARTMENT- CSE
SRMSCET

CONTENTS

- What/Why Diseases Detection ?
- Objective
- Diseases Studied :
  - Diabetes
  - Heart
  - Breast Cancer
- Research Aspects :
  - Dataset
  - Algorithms

    Naïve Bayes
    Logistic Regression
    Random Forest
    K Nearest Neighbour
- Web App
- Future Scope
- References

What/Why Disease Detection?

- A screening test is done to detect potential health disorders or diseases in people who do not have any symptoms of disease.
- Detecting diseases at early stage to treat them appropriately.
- Saves Time and Money.

Objective

- To develop a web app which predict major disease with user friendly and functionality rich web interface.
- Optimize the accuracy of prediction models and make the overall product valuable and efficient.
- Last but not the least,

  Help people and make their lives better.

Disease Studied

1. Diabetes

- It's a common chronic disease and lead to chronic damage and dysfunction of various tissues, especially eyes, kidneys, heart, blood vessels and nerves.

- The characteristic of diabetes is that the blood glucose is higher than the normal level, which is caused by defective insulin secretion or its impaired biological effects, or both.

- Diabetes can be divided into two categories:
  - Type 1 diabetes (T1D)
  - Type 2 diabetes (T2D)

- **Type 1 diabetes** :
  - Increased thirst and frequent urination
  - High blood glucose levels
  - Patients are mostly less than 30 years old.
  - Can’t be cured effectively with oral medications alone and the patients are required insulin therapy.

- **Type 2 diabetes:**
  - Occurs in middle-aged and elderly people
  - Occurrence of obesity, hypertension, dyslipidemia, arteriosclerosis, and other diseases.

- **Dataset Fields:**

  - Pregnancies
  - Glucose
  - Blood Pressure
  - Skin Thickness
  - Insulin
  - BMI (Body Mass Index)
  - Diabetes Pedigree Function
  - Age

Disease Studied

2. Heart

- **Dataset Fields :**

  - Oldpeak (depression induced by exercise relative to rest)
  - Slope (the slope of the peak exercise ST segment)
  - CA (number of major vessels colored by flourosopy)
  - Trestbps (resting blood pressure)
  - Fbs (fasting blood sugar)
  - Restecg (resting electrocardiographic results)
  - Thalach (maximum heart rate achieved)
  - Exang (exercise induced angina)
  - Age
  - Sex
  - Chest Pain
  - Cholestrol
  - THAL

Disease Studied

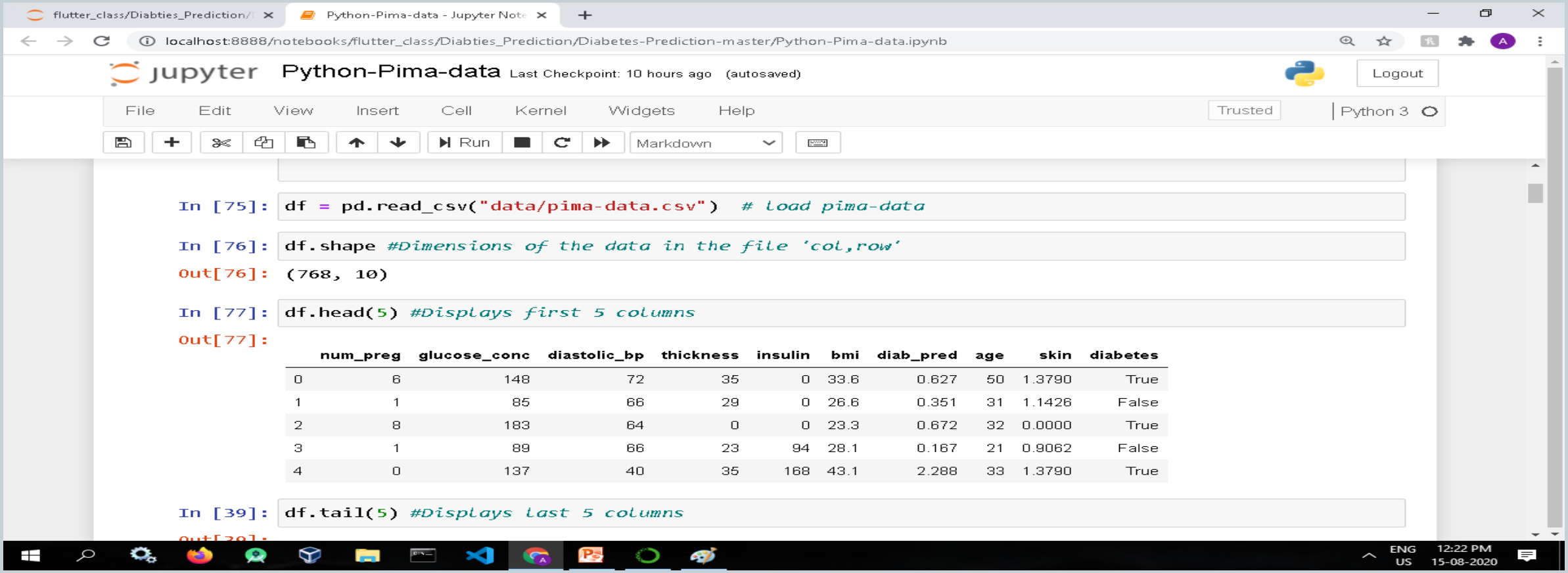
3. Breast Cancer

- It's a disease in which cells in the breast grow out of control. There are different kinds of breast cancer. The kind of breast cancer depends on which cells in the breast turn into cancer.

- **Dataset Fields :**
  - Mean Radius
  - Mean Texture
  - Mean Perimeter
  - Mean Area
  - Mean Smoothness

Dataset

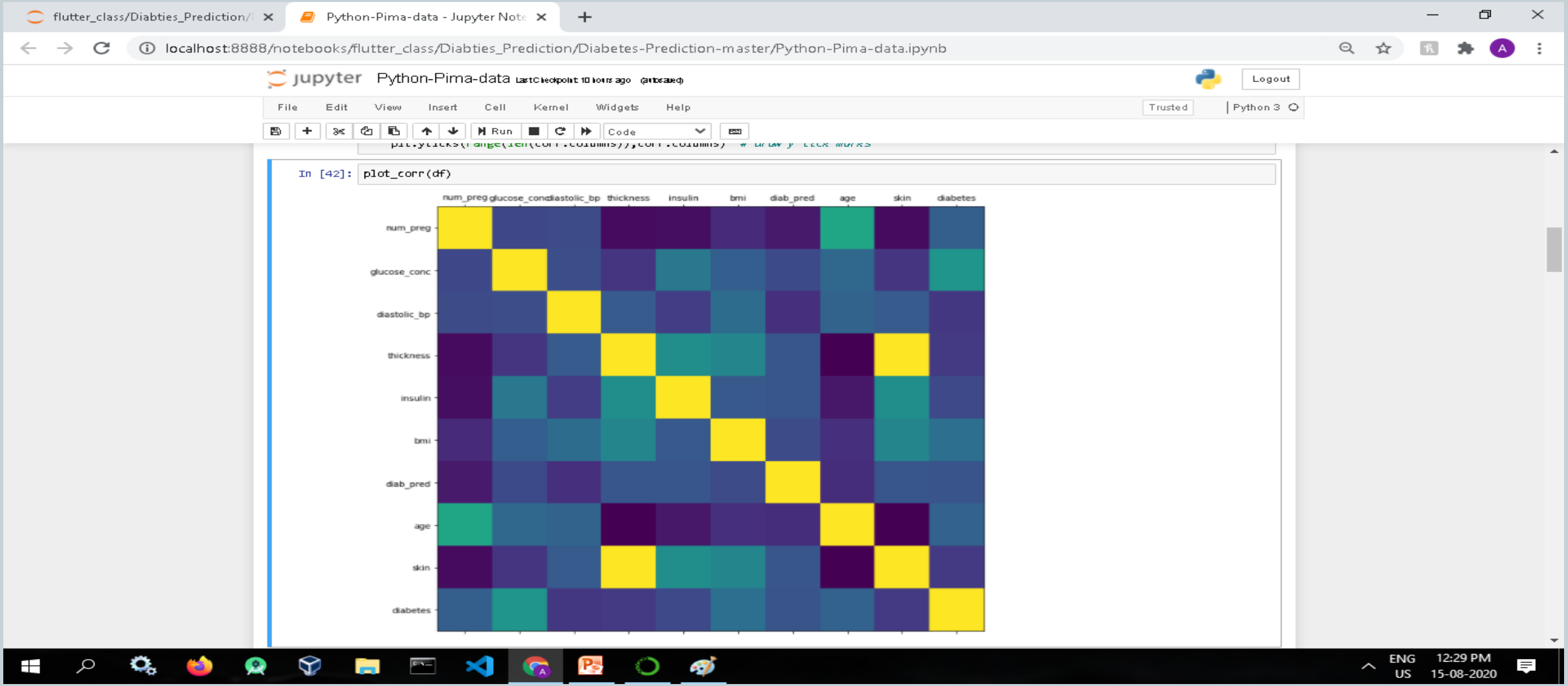
- The ‘Pima’ and ‘Kaggle’ dataset is chosen.



	num_preg	glucose_conc	diastolic_bp	thickness	insulin	bmi	diab_pred	age	skin	diabetes
0	6	148	72	35	0	33.6	0.627	50	1.3790	True
1	1	85	66	29	0	26.6	0.351	31	1.1426	False
2	8	183	64	0	0	23.3	0.672	32	0.0000	True
3	1	89	66	23	94	28.1	0.167	21	0.9062	False
4	0	137	40	35	168	43.1	2.288	33	1.3790	True

- **Data Preprocessing**:

  - Checking correlation among the dataset fields



- Handling Categorical Data

flutter_class/Diabties_Prediction/ | Python-Pima-data - Jupyter Note

localhost:8888/notebooks/flutter_class/Diabties_Prediction/Diabetes-Prediction-master/Python-Pima-data.ipynb#Deleting-correlated-columns

jupyter Python-Pima-data Last Checkpoint: 10 hours ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

Change true to 1 and false to 0

In [43]:
```
diabetes_map = {True : 1, False : 0} #Storing the column to be changed in a variable
```

In [44]:
```
df['diabetes'] = df['diabetes'].map(diabetes_map)
```

In [45]:
```
df.head(2)
```

Out[45]:

	num_preg	glucose_conc	diastolic_bp	thickness	insulin	bmi	diab_pred	age	skin	diabetes
0	6	148	72	35	0	33.6	0.627	50	1.3790	1
1	1	85	66	29	0	26.6	0.351	31	1.1426	0

In [46]:
```
del df['skin']
```

ENG IN 12:33 PM 15-08-2020

- Dealing with unexpected/missing values (replacing by column mean)

flutter_class/Diabties_Prediction/ | Python-Pima-data - Jupyter Note

localhost:8888/notebooks/flutter_class/Diabties_Prediction/Diabetes-Prediction-master/Python-Pima-data.ipynb#Deleting-correlated-columns

jupyter Python-Pima-data Last Checkpoint: 10 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Rows with unexpected 0 value

In [53]:
```
print("# rows in dataframe {0}".format(len(df)))
print("# rows missing glucose_conc: {0}".format(len(df.loc[df['glucose_conc'] == 0])))
print("# rows missing diastolic_bp: {0}".format(len(df.loc[df['diastolic_bp'] == 0])))
print("# rows missing thickness: {0}".format(len(df.loc[df['thickness'] == 0])))
print("# rows missing insulin: {0}".format(len(df.loc[df['insulin'] == 0])))
print("# rows missing bmi: {0}".format(len(df.loc[df['bmi'] == 0])))
print("# rows missing diab_pred: {0}".format(len(df.loc[df['diab_pred'] == 0])))
print("# rows missing age: {0}".format(len(df.loc[df['age'] == 0])))
```

```
# rows in dataframe 768
# rows missing glucose_conc: 5
# rows missing diastolic_bp: 35
# rows missing thickness: 227
# rows missing insulin: 374
# rows missing bmi: 11
# rows missing diab_pred: 0
# rows missing age: 0
```

Impute missing data

In [56]:
```
from sklearn.preprocessing import Imputer

# Impute with mean all 0 readings

fill_0 = Imputer(missing_values= 0, strategy = "mean", axis=0)

X_train = fill_0.fit_transform(X_train)
X_test = fill_0.fit_transform(X_test)
```

ENG IN 12:35 PM 15-08-2020

- **Performance Testing:**
  - Tested accuracy on training and testing subsets of data (Cross Validation).

flutter_class/Diabties_Prediction/ | Python-Pima-data - Jupyter Note

localhost:8888/notebooks/flutter_class/Diabties_Prediction/Diabetes-Prediction-master/Python-Pima-data.ipynb#Deleting-correlated-columns

jupyter Python-Pima-data Last Checkpoint: 10 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

Performance on Training Data

In [58]:
```
# predict values using the training data
nb_predict_train = nb_model.predict(X_train)
# import the performance metrics library

from sklearn import metrics

# Accuracy
print("Accuracy: {0:.4f}".format(metrics.accuracy_score(y_train,nb_predict_train)))
print()
```

```
Accuracy: 0.7542
```

Performance on Testing Data

In [59]:
```
# predict values using the training data
nb_predict_test = nb_model.predict(X_test)
# import the performance metrics library

from sklearn import metrics

# Accuracy
print("Accuracy: {0:.4f}".format(metrics.accuracy_score(y_test,nb_predict_test)))
print()
```

```
Accuracy: 0.7359
```

Metrics

ENG IN 12:42 PM 15-08-2020

- Checked Metrics via confusion matrix

flutter_class/Diabties_Prediction/ | Python-Pima-data - Jupyter Note

localhost:8888/notebooks/flutter_class/Diabties_Prediction/Diabetes-Prediction-master/Python-Pima-data.ipynb#Deleting-correlated-columns

jupyter Python-Pima-data Last Checkpoint: 10 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Metrics

In [60]:
```
print("Confusion Matrix")
# Note the use of labels for 1 = true and 0 = false to lower right
print("{0}".format(metrics.confusion_matrix(y_test,nb_predict_test, labels=[1,0])))
# Classification report
print("")
print("Classification Report")
print("")

print(metrics.classification_report(y_test,nb_predict_test, labels=[1,0]))
```

```
Confusion Matrix
[[ 52  28]
 [ 33 118]]

Classification Report

              precision    recall  f1-score   support

           1       0.61      0.65      0.63        80
           0       0.81      0.78      0.79       151

    accuracy                           0.74       231
   macro avg       0.71      0.72      0.71       231
weighted avg       0.74      0.74      0.74       231
```

Random Forest

In [61]:
```
from sklearn.ensemble import RandomForestClassifier
```

ENG IN 12:45 PM 15-08-2020

Algorithms

1. Naïve Bayes

- Naive Bayes classifier is a straightforward and powerful algorithm for the classification task. (used in NLP a lot)

- To understand it we need to understand the **Bayes theorem**.

$$P(H \mid E) = \frac{P(E \mid H) * P(H)}{P(E)}$$

Where :

- P(H) probability of hypothesis being true.
- P(E) probability of the event.
- P(E|H) probability of the event given that hypothesis is true.
- P(H|E) probability of the hypothesis given the event.

Algorithms

2. Random Forest

- The fundamental concept used - **The wisdom of crowds**

  " A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models."

- Here we use multiple random decision trees for a better accuracy.

- Instead of using **information gain/gini index** for calculating the root node, the process will happen randomly.

- It reduces the variance of the individual decision trees by randomly selecting trees and then picking the class that gets the most vote.

Algorithms

3. Logistic Regression

- It's a supervised classification algorithm that measures the relationship between the dependent and one or more independent variables.
- We pre-assign the **scores** and **weights** for the independent variables (calculated over the training data set) and use them to compute the **Logits (** ** *Till here linear regression model*).
- The Logits will pass through the **softmax function**(logistic function), which will return the probabilities for each target class.
- The high probability target class will be the predicted target class.

Algorithms

4. K Nearest Neighbour

- It's a **non-parametric** supervised classification algorithm.
- Used to predict the target label by finding the nearest neighbour class. The closest class will be identified using the distance measures like **Euclidean distance**.
- Selecting the value of **K** is the most critical problem:
  - A small value of K means that noise will have a higher influence on the result i.e., the probability of overfitting is very high.
  - A large value of K makes it computationally expensive and defeats the basic idea
  - A simple approach to select k is k = n^(1/2).

Web App

(Time for some practical usecase)

Future Scope

- **Adding functionality for registered users :**
  - Nearest Doctor Recommendations
  - Personal drive for storing medical history records

- **Adding tests for other diseases :** (COVID19, etc)

References

- Cox M. E., Edelman D. (2009). Tests for screening and diagnosis of type 2 diabetes. *Clin. Diabetes* 27 132–138. 10.2337/diaclin.27.4.132
- https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/
- https://research.ijcaonline.org/volume95/number17/pxc3896801.pdf
- https://www.researchgate.net/publication/316432650_Diabetes_Prediction_Using_Medical_Data

Thank You!