University of Bradford

Artificial Intelligence
COS5028-B

Professor Rami Qahwaji

2022/2023

Abdulaziz Albeloushi	**21025497**
Almotasembelah Daya	**22038511**
Yazan Amro	**22038503**

Abstract

The purpose of this report is to develop multiple AI solutions using Weka tool and Python programming language for the classification of diabetes based on many attributes in diabetes dataset. The solutions try to give a future prediction for being under diabetes

Faculty of Engineering and informatics

Contents

1. Introduction:

With the proper medical treatment and services, diabetes, primarily type 1 diabetes, in younger age groups can be prevented mainly as a cause of death. One of the top five non-communicable diseases (NCDs) according to the UN and WHO's action Plan to Face the NCDs Challenge is diabetes.
In this report, a Python solution will be developed to help us create an AI system that may make those patients aware of dealing with diabetes, including patient self-management tools and clinical decision support based on many attribute values from patients' data sets.

Diabetes complications such as heart disease, vision loss, lower-limb amputation, and kidney disease are associated with chronically high levels of sugar remaining in the bloodstream. While there is no cure for diabetes, many patients can benefit from strategies such as losing weight, eating healthily, staying active, and receiving medical treatment. Early diagnosis can lead to lifestyle changes and more effective treatment, making diabetes risk prediction models valuable tools for the general public and public health officials.

2. Background:

In this project, we will use a dataset we found on Kaggle.com. It was shared on that platform by **Alex Teboul**. This dataset contains a variety of health indicators that help in training the AI system to detect patterns that lead to getting diabetes and thus warn the patient and advise them to make some changes in their lifestyle/diet. We chose this dataset because when the AI system interacts with it, it reduces the margin of error and coincidence and therefore provides more accurate and precise results, and that's all thanks to the relatively high number of attributes/variables/health indicators.

3. Methodology and Data:

1. Dataset

Diabetes is one of the most common chronic diseases in the US, affecting millions of people annually and placing a heavy financial strain on the economy. Diabetes is a significant chronic condition that impairs a person's capacity to control blood glucose levels, which can shorten life expectancy and lower quality of life. Sugars from various foods are converted during digestion and released into the bloodstream. The pancreas is prompted to secrete insulin as a result. Insulin assists in making it possible for body cells to use the carbohydrates in the bloodstream as fuel. In

general, diabetes is characterized by the body's inability to utilize the insulin produced or not producing enough of it
Chronically high amounts of sugar in the bloodstream for people with diabetes get linked to complications like heart disease, vision loss, lower limb amputation, and kidney illness. Although there is no known cure for diabetes, many individuals can lessen its adverse effects by adopting lifestyle changes such as decreasing weight, eating a healthy diet, exercising, and receiving medical care. Predictive models for diabetes risk are potent tools for the general population and public health officials since early diagnosis can result in lifestyle changes and more successful treatment.

The Behavioral Risk Factor Surveillance System (BRFSS) is an annual telephone survey conducted by the Centers for Disease Control and Prevention (CDC). The survey collects responses from over 400,000 Americans each year on health-related risk behaviours, chronic health conditions, and the use of preventive services. It has taken place every year since 1984. A CSV file of the dataset available on Kaggle for 2015 was used for this project. This original dataset has 330 features and responses from 441,455 people. These characteristics are questions posed to participants or calculated variables based on individual responses.

The dataset used for this analysis is from a web source Kaggle.com; the dataset contains about 22 columns and 253681 rows, and it includes all types of attributes, and their SAS variable name, description, and values (Table 1) below demonstrates all its details.

Diabetes health indicators/attributes

Table 1

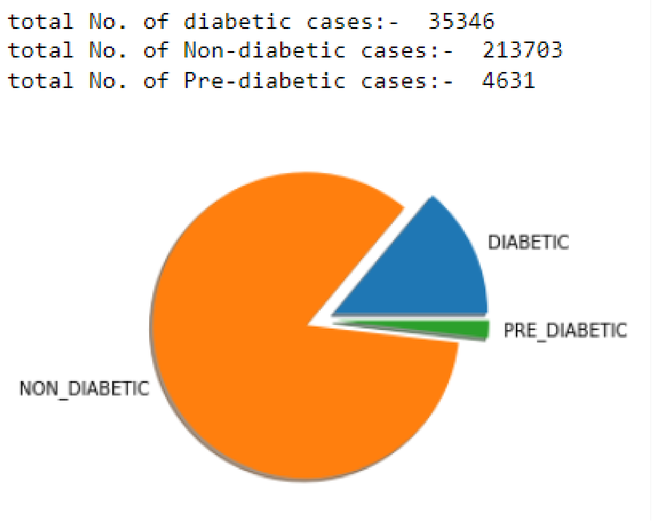
Diabetes_012	0 = no diabetes 1 = prediabetes 2 = diabetes
HighBP (blood pressure)	0 = no high BP 1 = high BP
HighChol	0 = no high cholesterol 1 = high cholesterol
CholCheck	0 = no cholesterol check in 5 years 1 = yes cholesterol check in 5 years
BMI (body mass index)	Body Mass Index
Smoker	Have you smoked at least 100 cigarettes in your entire life? [Note: 5 packs = 100 cigarettes] 0 = no 1 = yes
Stroke	(Ever told) you had a stroke. 0 = no 1 = yes
Heart Disease or Attack	coronary heart disease (CHD) or myocardial infarction (MI) 0 = no 1 = yes
Physical Activity	physical activity in the past 30 days - **not including job** 0 = no 1 = yes
Fruits	Consume Fruit 1 or more times per day 0 = no 1 = yes
Vegetables	Consume Vegetables 1 or more times per day 0 = no 1 = yes
Heavy Alcohol Consumption	Heavy Alcohol Consumption per day 0 = no 1 = yes
Any Healthcare	If the person receives good health care 0 = no 1 = yes
NoDocbcCost	Could Not See Doctor Because of Cost 0 = no 1 = yes
GenHlth	General health rate 1 = Excellent 2 = Very good 3 = good 4 = fair 5 = poor
MentHlth	Number of Days Mental Health Not Good 1 – 30 number of days Notes: -- the number of days
PhysHlth	Number of Days Physical Health Not Good 1 – 30 number of days Notes: -- the number of days

DiffWalk	Difficulty Walking or Climbing Stairs 0 = no 1 = yes
Sex	Gender of the person 0 = female 1 = male
Age	1 = Age 18 to 24 2 = Age 25 to 29 3 = Age 30 to 34 4 = Age 35 to 39 5 = Age 40 to 44 6 = Age 45 to 49 7 = Age 50 to 54 8 = Age 55 to 59 9 = Age 60 to 64 10 = Age 65 to 69 11 = Age 70 to 74 12 = Age 75 to 79 13 = Age 80 or older
Education	Has received a level of education 1 = Never attended school or only kindergarten 2 = Grades 1 through 8 (Elementary) 3 = Grades 9 through 11 (Some high school) 4 = Grade 12 or GED (High school graduate) 5 = College 1 year to 3 years (Some college or technical school) 6 = College 4 years or more (College graduate)
Income	Income level 1 = Less than $10,000 2 = Less than $15,000 ($10,000 to less than $15,000) 3 = Less than $20,000 ($15,000 to less than $20,000) 4 = Less than $25,000 ($20,000 to less than $25,000) 5 = Less than $35,000 ($25,000 to less than $35,000) 6 = Less than $50,000 ($35,000 to less than $50,000) 7 = Less than $75,000 ($50,000 to less than $75,000) 8 = $75,000 or more

As shown above, all the attributes help in detecting where diabetes is most common among groups of people. And by feeding that data to our AI system, it will be able to pinpoint the risks of getting diabetes and study the history of patients in order to predict future cases and prevent them.

4. Data Visualization:

1. The distribution between Diabetes cases



```
total No. of diabetic cases:-  35346
total No. of Non-diabetic cases:-  213703
total No. of Pre-diabetic cases:-  4631
```

Figure 1.0

As shown in figure 1.0, the pie chart illustrates the number of cases of people who don’t suffer from diabetes, the ones who have it and those who are predicted to have it. The ones which are coloured orange are the people that don’t suffer from diabetes, and the ones which are blue are the ones that suffer from diabetes according to the dataset, meanwhile, the ones on the green are the people who are predicted to suffer from diabetes according to the variables on the data; seeing the distribution of the classes, the people who don’t suffer from Diabetes are nearly three times more than the people who suffer from diabetes a calculation of difference has been taken between diabetic cases: 35346, the Non-diabetic cases: 213703 and the predicted diabetic cases: 4631, only **1.8%** of the people are predicted to suffer from diabetes
The calculations and the statistics have all been done using Python

2. The distribution of the age

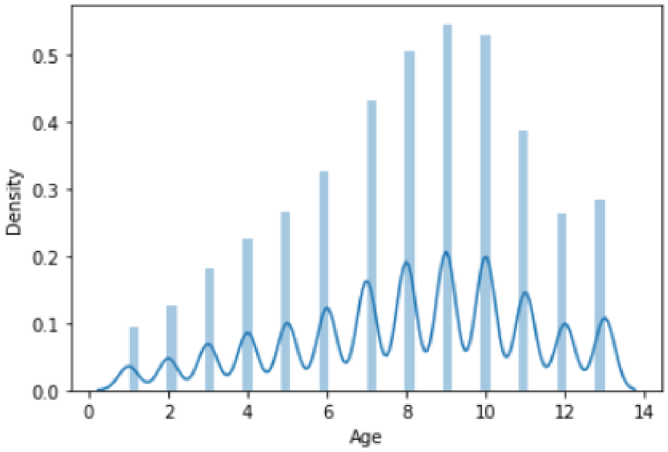
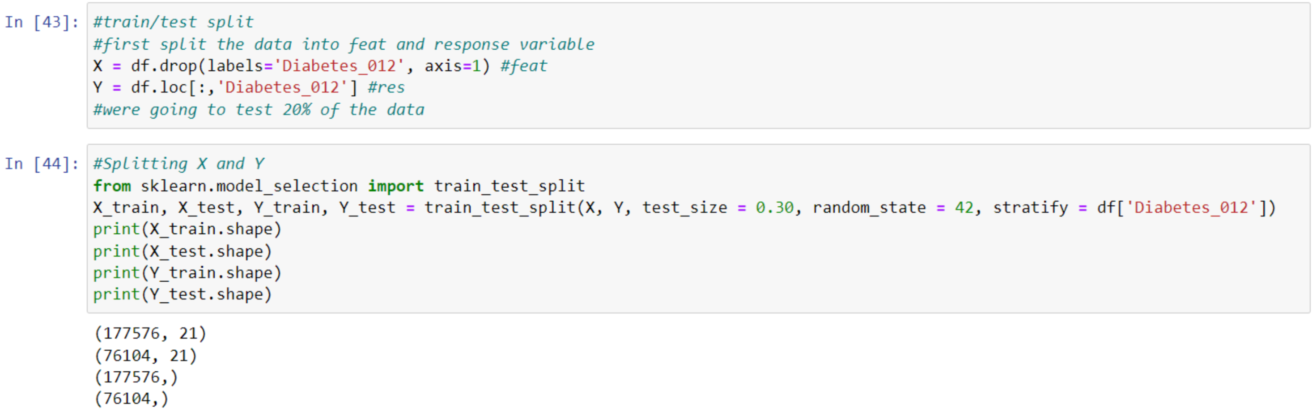


Figure 1.1

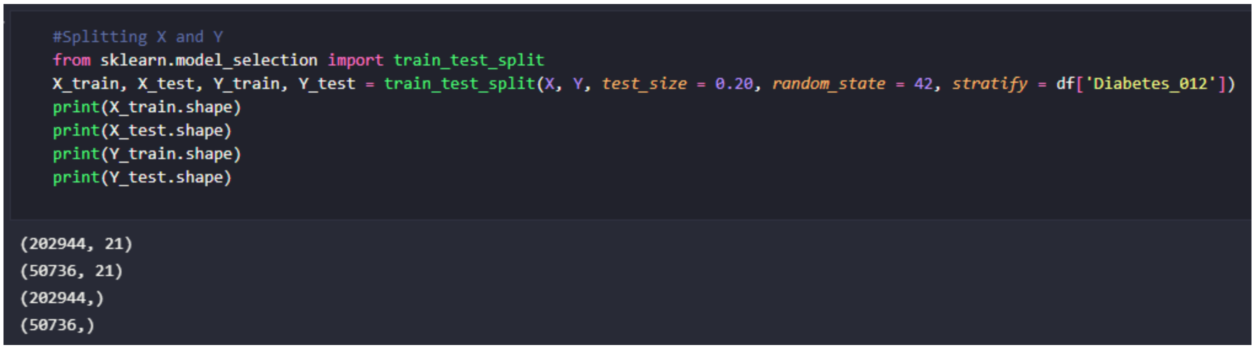
The column chart Above demonstrates the division of the ages, the top three ages that are shown are 9, which is between the age (60-64) which ranks first, 10 which is between (65-69) which ranks second and 8, which is between (55-59) which ranks third overall the persons who are above the age 50 that are mostly in the dataset there are at least 50% of them.

5. Data splitting training and testing:



```
In [43]: #train/test split
         #first split the data into feat and response variable
         X = df.drop(labels='Diabetes_012', axis=1) #feat
         Y = df.loc[:,'Diabetes_012'] #res
         #were going to test 20% of the data

In [44]: #Splitting X and Y
         from sklearn.model_selection import train_test_split
         X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.30, random_state = 42, stratify = df['Diabetes_012'])
         print(X_train.shape)
         print(X_test.shape)
         print(Y_train.shape)
         print(Y_test.shape)

         (177576, 21)
         (76104, 21)
         (177576,)
         (76104,)
```

Figure 1.2



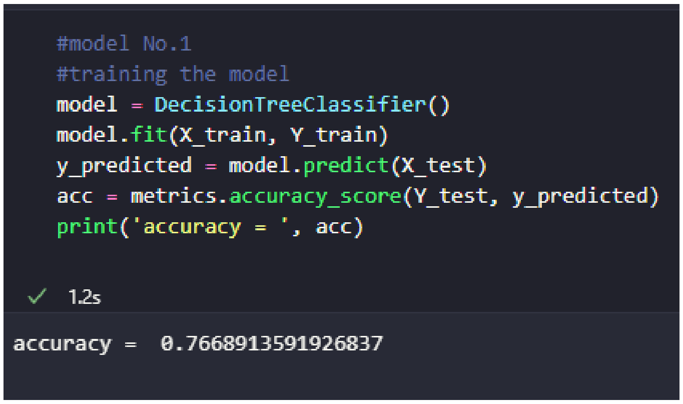
```
#Splitting X and Y
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.20, random_state = 42, stratify = df['Diabetes_012'])
print(X_train.shape)
print(X_test.shape)
print(Y_train.shape)
print(Y_test.shape)
```

```
(202944, 21)
(50736, 21)
(202944,)
(50736,)
```

Figure 1.3

Shows the procedure of how the data is set up. The data is stored in X and the target in Y; the method train_test_split was used, and the test size was set to 0.30 and the remainder of the training as shown in **figure 1.2**, the sections of the data has been split up using the random_state method, the same method has been used on **figure 1.3**, but the only changes were the test size, and it was changed to 0.20, and the results were different

6. Data Accuracy testing:



```
#model No.1
#training the model
model = DecisionTreeClassifier()
model.fit(X_train, Y_train)
y_predicted = model.predict(X_test)
acc = metrics.accuracy_score(Y_test, y_predicted)
print('accuracy = ', acc)
```

✓ 1.2s

```
accuracy =  0.7668913591926837
```

Figure 1.4

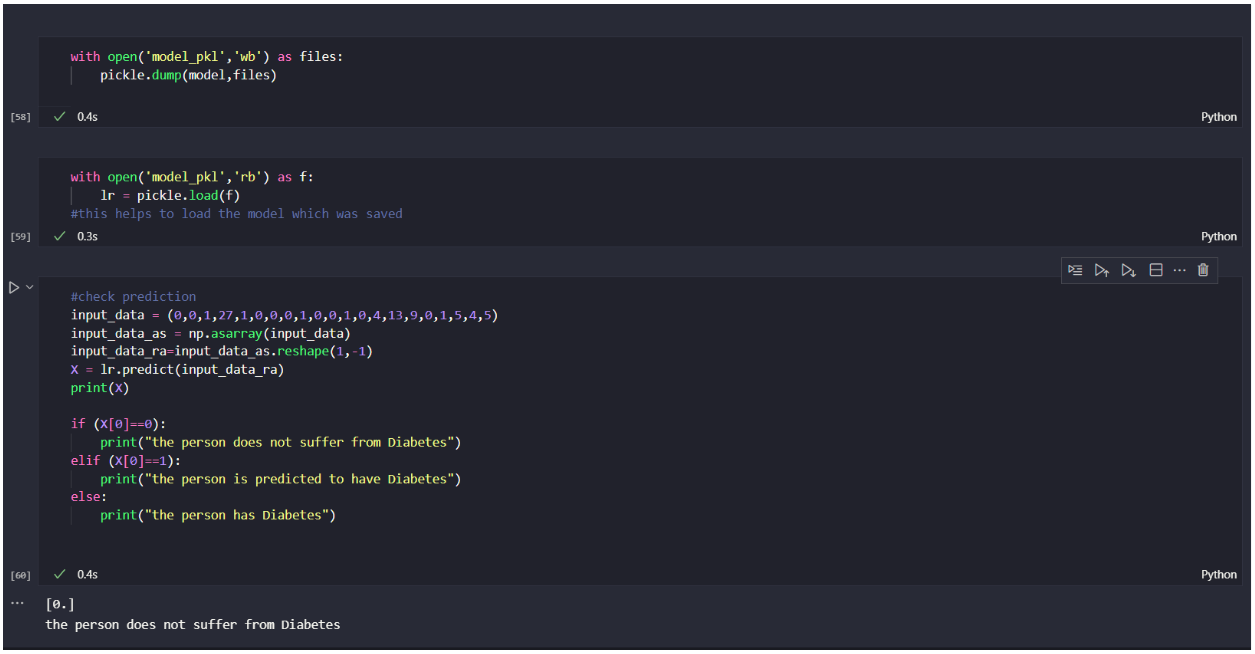
A test has been ran six times to calculate the average data accuracy score and the time, figure 1.4 highlights the method that has been used to calculate the score and the time, and the table (Table 2) shows the average time and score

Table 2

Time in seconds (s)	Accuracy score
1.9s	76.6%
1.4s	76.7%
1.3s	76.7%
1.9s	76.6%
1.2s	76.7%
1.2s	76.6%

as shown on the table the average time is between 1.2 seconds and 1.9 seconds while the accuracy score is 76%

7. Predictions



```
with open('model_pkl','wb') as files:
    pickle.dump(model,files)
```

```
with open('model_pkl','rb') as f:
    lr = pickle.load(f)
#this helps to load the model which was saved
```

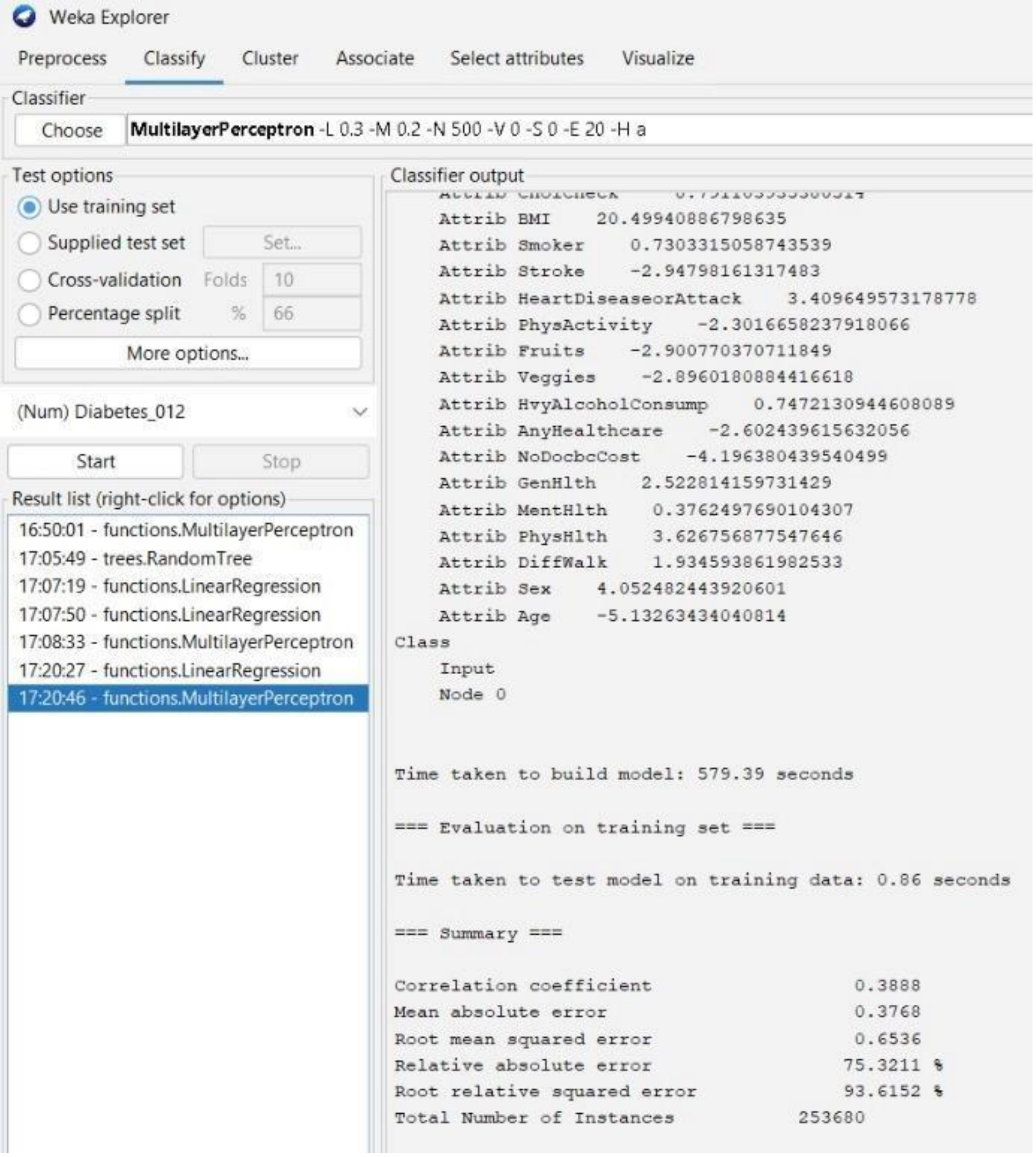
```
#check prediction
input_data = (0,0,1,27,1,0,0,0,1,0,0,1,0,4,13,9,0,1,5,4,5)
input_data_as = np.asarray(input_data)
input_data_ra=input_data_as.reshape(1,-1)
X = lr.predict(input_data_ra)
print(X)

if (X[0]==0):
    print("the person does not suffer from Diabetes")
elif (X[0]==1):
    print("the person is predicted to have Diabetes")
else:
    print("the person has Diabetes")
```

```
[0.]
the person does not suffer from Diabetes
```

Figure 1.5

This is the method that has been used to predict if the person suffers from Diabetes, or does not suffer from Diabetes, or is predicted to get the disease, using the input_data function it could help to calculate the output based on the dataset which was trained from, in the input section a number of variables/attributes values has been added to predict the outcome X on the 4th line in figure 1.4

8. Weka Tool solutions:



Classifier: MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

Test options:
- ● Use training set
- ○ Supplied test set
- ○ Cross-validation Folds 10
- ○ Percentage split % 66

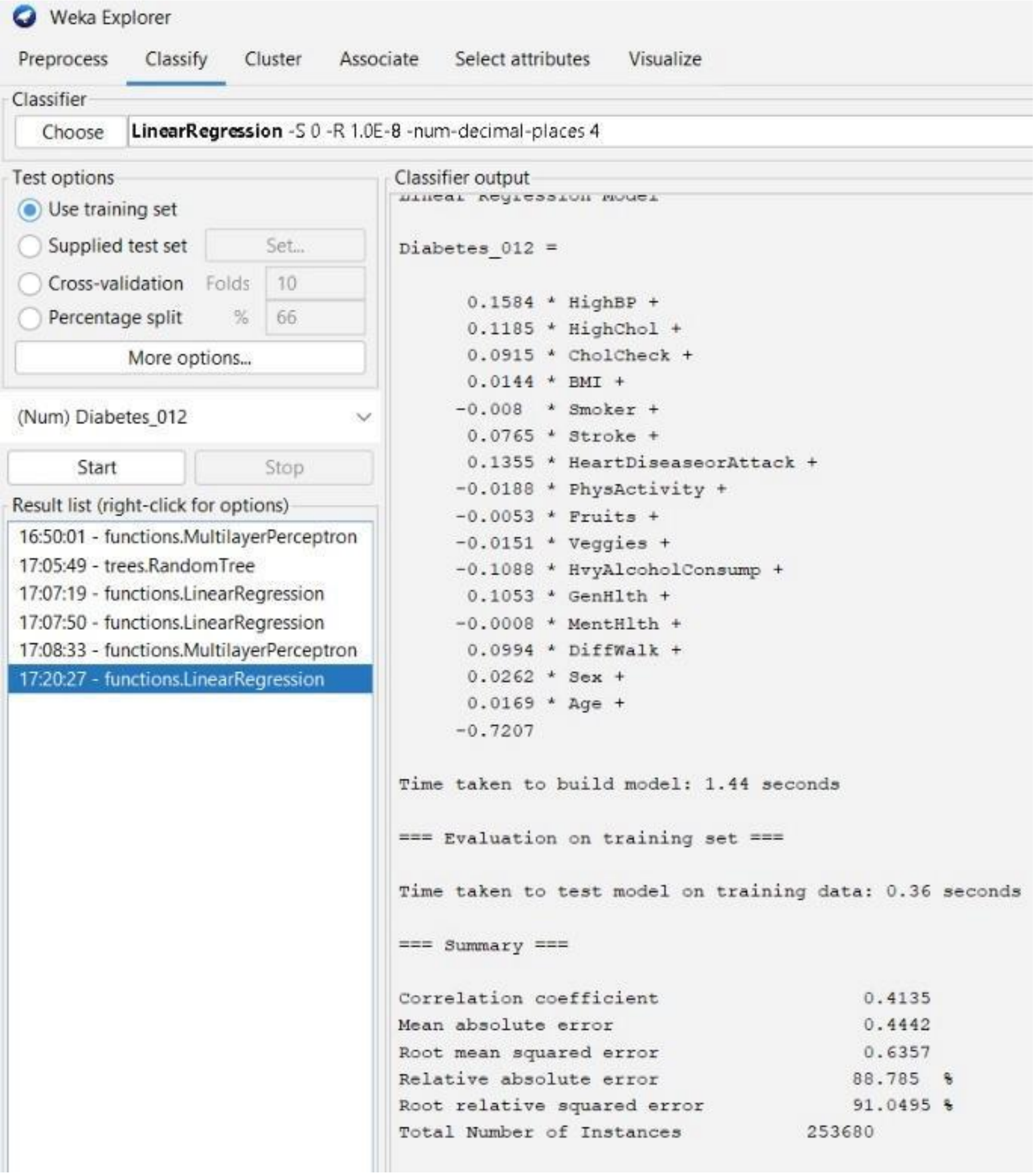
(Num) Diabetes_012

Result list (right-click for options)

```
16:50:01 - functions.MultilayerPerceptron
17:05:49 - trees.RandomTree
17:07:19 - functions.LinearRegression
17:07:50 - functions.LinearRegression
17:08:33 - functions.MultilayerPerceptron
17:20:27 - functions.LinearRegression
17:20:46 - functions.MultilayerPerceptron
```

```
    Attrib CholCheck    0.791103935360514
    Attrib BMI    20.49940886798635
    Attrib Smoker    0.7303315058743539
    Attrib Stroke    -2.94798161317483
    Attrib HeartDiseaseorAttack    3.409649573178778
    Attrib PhysActivity    -2.3016658237918066
    Attrib Fruits    -2.900770370711849
    Attrib Veggies    -2.8960180884416618
    Attrib HvyAlcoholConsump    0.7472130944608089
    Attrib AnyHealthcare    -2.602439615632056
    Attrib NoDocbcCost    -4.196380439540499
    Attrib GenHlth    2.522814159731429
    Attrib MentHlth    0.3762497690104307
    Attrib PhysHlth    3.626756877547646
    Attrib DiffWalk    1.934593861982533
    Attrib Sex    4.052482443920601
    Attrib Age    -5.13263434040814
Class
    Input
    Node 0


Time taken to build model: 579.39 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.86 seconds

=== Summary ===

Correlation coefficient                  0.3888
Mean absolute error                      0.3768
Root mean squared error                  0.6536
Relative absolute error                 75.3211 %
Root relative squared error             93.6152 %
Total Number of Instances           253680
```

We tried training and testing the data using Weka software by using the multilayer perception function taking Diabetes_012 as the main attribute However the result wasn't very promising, and the accuracy was low (roughly 38%).



Classifier

LinearRegression -S 0 -R 1.0E-8 -num-decimal-places 4

Test options

● Use training set

○ Supplied test set Set...

○ Cross-validation Folds 10

○ Percentage split % 66

(Num) Diabetes_012

Result list (right-click for options)

- 16:50:01 - functions.MultilayerPerceptron
- 17:05:49 - trees.RandomTree
- 17:07:19 - functions.LinearRegression
- 17:07:50 - functions.LinearRegression
- 17:08:33 - functions.MultilayerPerceptron
- 17:20:27 - functions.LinearRegression

Classifier output

```
Linear Regression Model

Diabetes_012 =

      0.1584 * HighBP +
      0.1185 * HighChol +
      0.0915 * CholCheck +
      0.0144 * BMI +
     -0.008  * Smoker +
      0.0765 * Stroke +
      0.1355 * HeartDiseaseorAttack +
     -0.0188 * PhysActivity +
     -0.0053 * Fruits +
     -0.0151 * Veggies +
     -0.1088 * HvyAlcoholConsump +
      0.1053 * GenHlth +
     -0.0008 * MentHlth +
      0.0994 * DiffWalk +
      0.0262 * Sex +
      0.0169 * Age +
     -0.7207

Time taken to build model: 1.44 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.36 seconds

=== Summary ===

Correlation coefficient                  0.4135
Mean absolute error                      0.4442
Root mean squared error                  0.6357
Relative absolute error                 88.785  %
Root relative squared error             91.0495 %
Total Number of Instances           253680
```

We also tried using a different function this time (linear regression) hoping to get a better result but unfortunately, we got an even worse result (roughly 41% accuracy). So we decided to stick with the python machine learning method because it was more accurate and within our knowledge and experience.

9. Proposed Ideas on possible solutions:

a) We cannot be completely safe from getting diabetes, but we can do a lot to try and avoid getting it by eating responsibly and monitoring our daily consumption of different nutrition values. And we can do that by developing a mobile application that can be used to track the user's nutrition and therefore giving them insights on where they can eat more or less of something. For example, when the user wants to eat a piece of cake they should get on the app and add it to their meal and then the application should give approximate values of the cake's nutrition and tell them if it's safe to eat it while taking into consideration the other meals that the user has consumed earlier that day.

b) The development of both types of diabetes mellitus depends on two main things: a genetic predisposition, and an environmental trigger .an essential factor that diabetes patients may face is the genetics factor, our idea is to build a prediction system consisting of a mobile application that gives a future prediction for any person if there is a Prospect of being under diabetes according to his/her family genetic history.

10.**References**:

1) Cousin, E., Duncan, B.B., Stein, C., Ong, K.L., Vos, T., Abbafati, C., Abbasi-Kangevari, M., Abdelmasseh, M., Abdoli, A., Abd-Rabu, R., Abolhassani, H., Abu-Gharbieh, E., Accrombessi, M.M.K., Adnani, Q.E.S., Afzal, M.S., Agarwal, G., Agrawaal, K.K., Agudelo-Botero, M., Ahinkorah, B.O. and Ahmad, S. (2022). Diabetes mortality and trends before 25 years of age: an analysis of the Global Burden of Disease Study 2019. *The Lancet Diabetes & Endocrinology*, 10(3). doi:10.1016/s2213-8587(21)00349-1.

2) diabetes.org. (n.d.). *Genetics of Diabetes | ADA*. [online] Available at: https://diabetes.org/diabetes/genetics-diabetes.

3) Maruch, S. and Maruch, A. (2006). *Python For Dummies*. [online] *Google Books*. John Wiley & Sons. Available at: https://books.google.co.uk/books?hl=en&lr=&id=LqmaDwAAQBAJ&oi=fnd&pg=PA3&dq=Python+for+dummies&ots=TM8LWsepnw&sig=oxS-EF6QvXX1B_47DZsU0EAx9Cw&redir_esc=y#v=onepage&q&f=false [Accessed 24 Nov. 2022].

4) ProjectPro. (n.d.). *How to split train test data using sklearn and python? -*. [online] Available at: https://www.projectpro.io/recipes/split-train-test-data-using-sklearn-and-python [Accessed 23 Nov. 2022].

5) TEBOUL, A. (2021). *Diabetes Health Indicators Dataset*. [online] www.kaggle.com. Available at: https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset?resource=download&select=diabetes_012_health_indicators_BRFSS2015.csv [Accessed 23 Nov. 2022].

6) Thomas, D.E. and Elliott, E.J. (2010). The use of low-glycaemic index diets in diabetes control. *British Journal of Nutrition*, 104(6), pp.797–802. doi:10.1017/s0007114510001534.

7) www.cdc.gov. (2019). *CDC - 2015 BRFSS Survey Data and Documentation*. [online] Available at: https://www.cdc.gov/brfss/annual_data/annual_2015.html