KONYA FOOD AND AGRICULTURE UNIVERSITY FACULTY OF ENGINEERING AND ARCHITECTURE DEPARTMENT OF COMPUTER ENGINEERING

CENG 4202

INTRODUCTION TO DATA SCIENCE

CLASSIFICATION ASSIGNMENT

Name Last Name

Mehmet Gürsel ARSLAN

GitHub: https://github.com/GurselArslan/classificationAssignment

ID Number

162010020006

Lecturer

Dr. Öğr. Üyesi Burak YILMAZ

Submission Date

2021

CONTENTS

1. SELECTING AND REVIEWING THE DATA SET 1

2. PREPROCESSING 2

3. SELECTION AND APPLICATION OF THE CLASSIFICATION METHOD 4

4. CONFUSION MATRIX 6

5. EVALUATION SCORES 6

REFERENCES 7

1. SELECTING AND REVIEWING THE DATA SET

Diabetes Dataset was chosen to generate a classification model. (Kaggle, 2021). After downloading the dataset, it was opened using the Python Pandas library. The purpose of this dataset is to predict within the information given whether a patient has diabetes. This dataset contains the following columns:

Pregnancies: The column containing the number of times the person is pregnant

Gender: The column containing the gender information of people. Men are indicated by M, Women by F.

Glucose: The column containing the amount of glucose in people.

BloodPressure: The column containing blood pressure information of people.

SkinThickness: The column containing the body thickness information of people.

Insulin: The column containing the insulin amount of the person.

BMI: The column containing the body mass indices of people.

DiabetesPedigreeFunction: The Column containing function results of diabetes patients in the person's family tree.

Age: The column containing the age information of people.

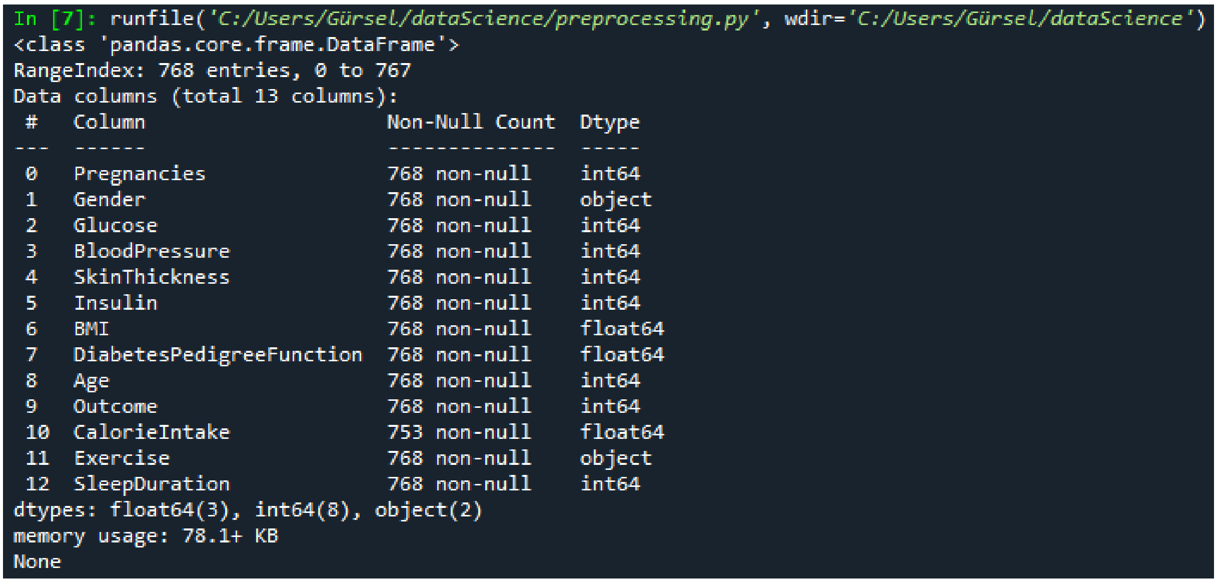
Outcome: The column containing information on whether people are at risk of diabetes. 1 for Diabetes, Not Diabetes 0

CalorieIntake: The column containing the daily calorie intake of people.

Exercise: The column containing the daily exercise times of the people. (Evening, Morning, Both or No)

SleepDuration: The column containing the daily sleep times of the people.

The data types of the information kept in this data set are shown in the figure 1 below.



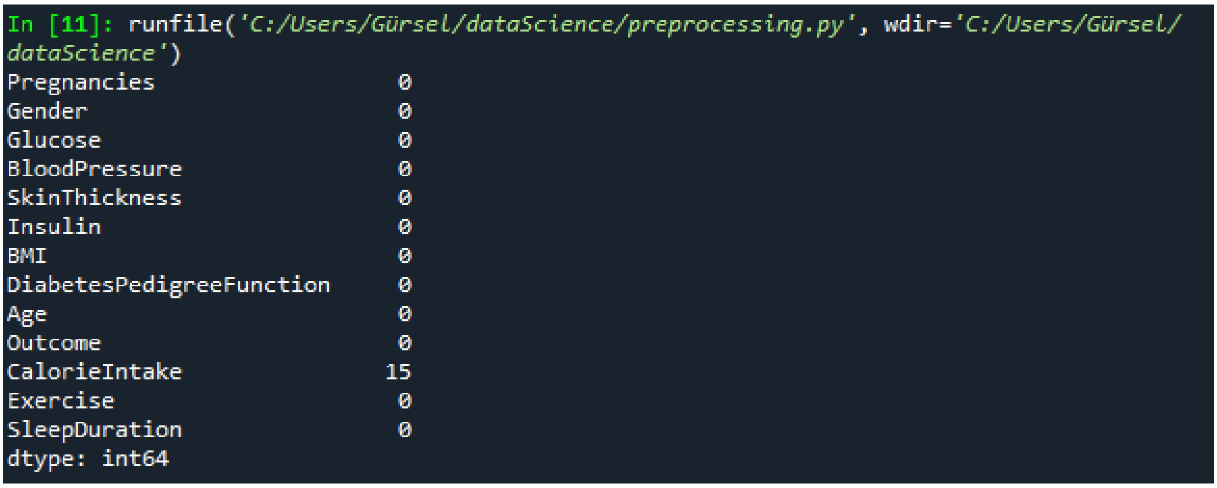
```
In [7]: runfile('C:/Users/Gürsel/dataScience/preprocessing.py', wdir='C:/Users/Gürsel/dataScience')
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 13 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Gender                    768 non-null    object
 2   Glucose                   768 non-null    int64
 3   BloodPressure             768 non-null    int64
 4   SkinThickness             768 non-null    int64
 5   Insulin                   768 non-null    int64
 6   BMI                       768 non-null    float64
 7   DiabetesPedigreeFunction  768 non-null    float64
 8   Age                       768 non-null    int64
 9   Outcome                   768 non-null    int64
 10  CalorieIntake             753 non-null    float64
 11  Exercise                  768 non-null    object
 12  SleepDuration             768 non-null    int64
dtypes: float64(3), int64(8), object(2)
memory usage: 78.1+ KB
None
```

Figure 1 Data types before pre-processing

2. PREPROCESSING

The “Gender” and “Exercise” columns in the data set were taking the objects as data types, as can be seen in figure 1. Converting such categorical data to numerical data is part of the pre-treatment process. It is more difficult for computers to understand categorical data and make calculations. Therefore, the data in these two columns were converted into numerical data. This process was done using the label encoding technique. Using the LabelEncoder class under the Sklearn library, Gender data has been converted to 1 for Male and 0 for Female. As a result of the same operation for Exercise data, it was changed as 0-> Both, 1-> Evening, 2-> Morning and 3-> No.

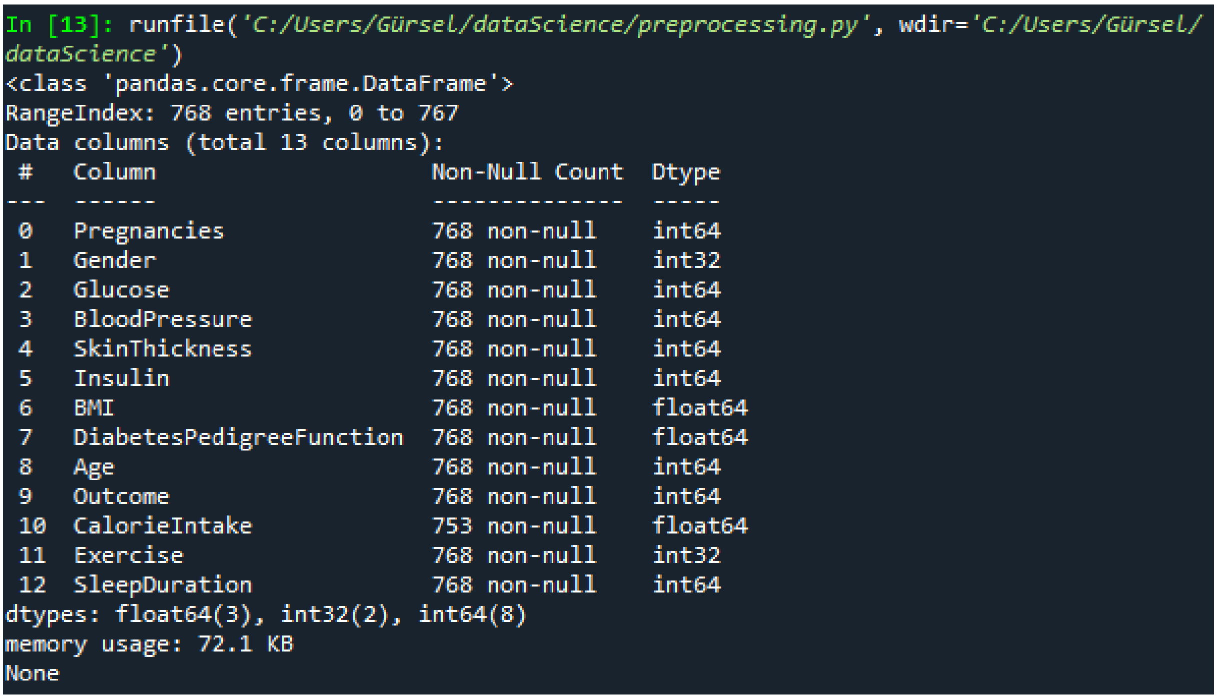
Next, the missing data in the data set was checked, another step of the preprocessing process, the missing data was detected and filled. Below is Figure 2 showing the missing data.



```
In [11]: runfile('C:/Users/Gürsel/dataScience/preprocessing.py', wdir='C:/Users/Gürsel/
dataScience')
Pregnancies                  0
Gender                       0
Glucose                      0
BloodPressure                0
SkinThickness                0
Insulin                      0
BMI                          0
DiabetesPedigreeFunction     0
Age                          0
Outcome                      0
CalorieIntake               15
Exercise                     0
SleepDuration                0
dtype: int64
```

Figure 2 Number of missing data by column

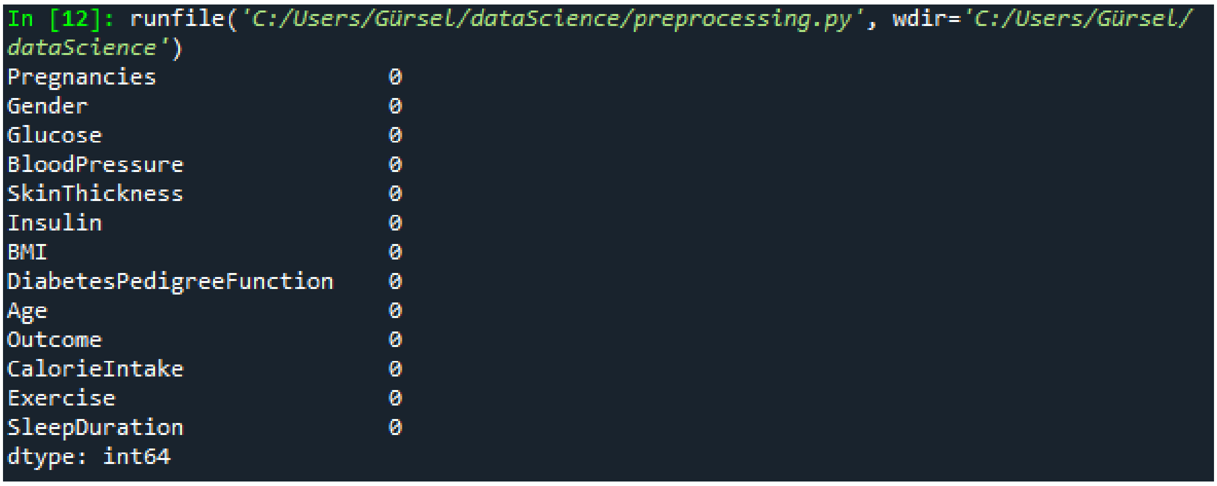
As can be seen in Figure 2, there are 15 NaN values in the CalorieIntake column. At this stage, it is necessary to fill in the missing data. Some of the methods that can be chosen are deleting rows with missing values, filling missing data with the mean value of this column, or filling missing data with the median value of this column. For the process of filling the missing data, the method of filling with the median value was chosen and applied.



```
In [13]: runfile('C:/Users/Gürsel/dataScience/preprocessing.py', wdir='C:/Users/Gürsel/
dataScience')
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 13 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               768 non-null    int64  
 1   Gender                    768 non-null    int32  
 2   Glucose                   768 non-null    int64  
 3   BloodPressure             768 non-null    int64  
 4   SkinThickness             768 non-null    int64  
 5   Insulin                   768 non-null    int64  
 6   BMI                       768 non-null    float64
 7   DiabetesPedigreeFunction  768 non-null    float64
 8   Age                       768 non-null    int64  
 9   Outcome                   768 non-null    int64  
 10  CalorieIntake             753 non-null    float64
 11  Exercise                  768 non-null    int32  
 12  SleepDuration             768 non-null    int64  
dtypes: float64(3), int32(2), int64(8)
memory usage: 72.1 KB
None
```

Figure 3 Data types after preprocessing

As can be seen in Figure 3, data types in all columns are numerical.



```
In [12]: runfile('C:/Users/Gürsel/dataScience/preprocessing.py', wdir='C:/Users/Gürsel/
dataScience')
Pregnancies                 0
Gender                      0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
CalorieIntake               0
Exercise                    0
SleepDuration               0
dtype: int64
```

Figure 4 Number of missing data by column after preprocessing

As can be seen in Figure 4, there is no NaN value in any column. As mentioned above, missing data are filled in with the median value of the column.

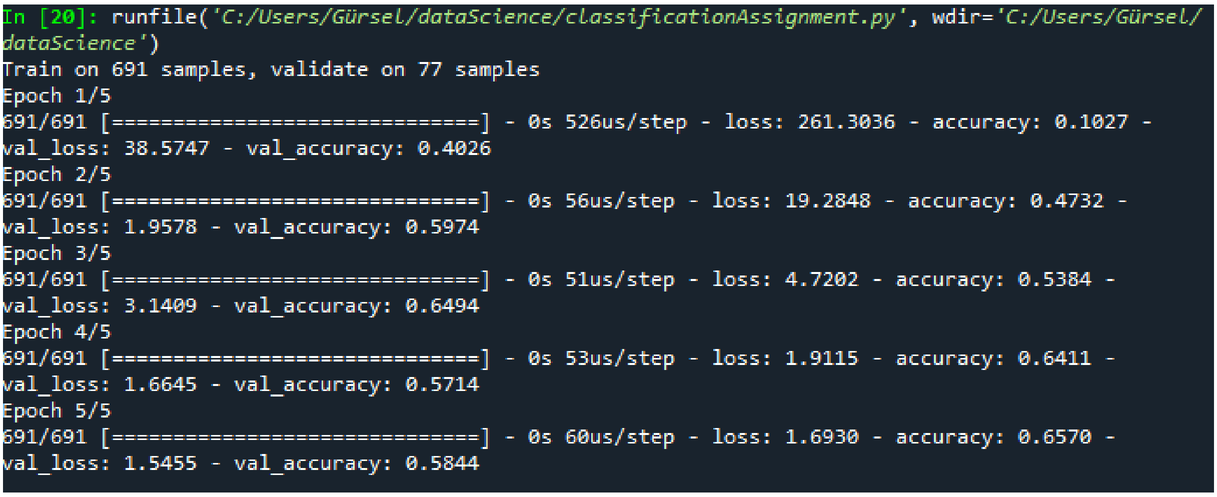
All columns in the data set contain information that should be taken into account when classifying. There are no columns that should be ignored when classifying, such as ID or name. As a result of the above operations, the data set has been made more suitable for classification.

3. SELECTION AND APPLICATION OF THE CLASSIFICATION METHOD

Once the data set has been selected, an appropriate classification method should be chosen to make a classification in this data set. In this assignment, the diabetes data set is selected. While choosing the appropriate classification method for this data set, the source given in the CENG 4202 Introduction to Data Science course was used (Bag, 2020). As a result of the research, the Neural Networks method has been deemed suitable for problems that need to be diagnosed a disease from medical screening.

Keras, an open-source neural network library, was used to classify this data set (Keras, n.d). For Keras, "Outcome" was parsed as a result and the other 12 columns as input. Thus, training data were defined. After this process, the model and layers were defined with the necessary parameters. Finally, the model was run with the data in the dataset and the classification process was applied.

As an example, the output obtained with the parameters determined as 5 repetitions and 90% training and 10% test is given below.



```
In [20]: runfile('C:/Users/Gürsel/dataScience/classificationAssignment.py', wdir='C:/Users/Gürsel/
dataScience')
Train on 691 samples, validate on 77 samples
Epoch 1/5
691/691 [==============================] - 0s 526us/step - loss: 261.3036 - accuracy: 0.1027 -
val_loss: 38.5747 - val_accuracy: 0.4026
Epoch 2/5
691/691 [==============================] - 0s 56us/step - loss: 19.2848 - accuracy: 0.4732 -
val_loss: 1.9578 - val_accuracy: 0.5974
Epoch 3/5
691/691 [==============================] - 0s 51us/step - loss: 4.7202 - accuracy: 0.5384 -
val_loss: 3.1409 - val_accuracy: 0.6494
Epoch 4/5
691/691 [==============================] - 0s 53us/step - loss: 1.9115 - accuracy: 0.6411 -
val_loss: 1.6645 - val_accuracy: 0.5714
Epoch 5/5
691/691 [==============================] - 0s 60us/step - loss: 1.6930 - accuracy: 0.6570 -
val_loss: 1.5455 - val_accuracy: 0.5844
```

Figure 5 Classification result with 5 epochs

In Figure 5, 5 repetitions are set to show, so it is normal for the accuracy percentages to be low. Accuracy percentages are higher in normal conditions run as 50 repetitions.

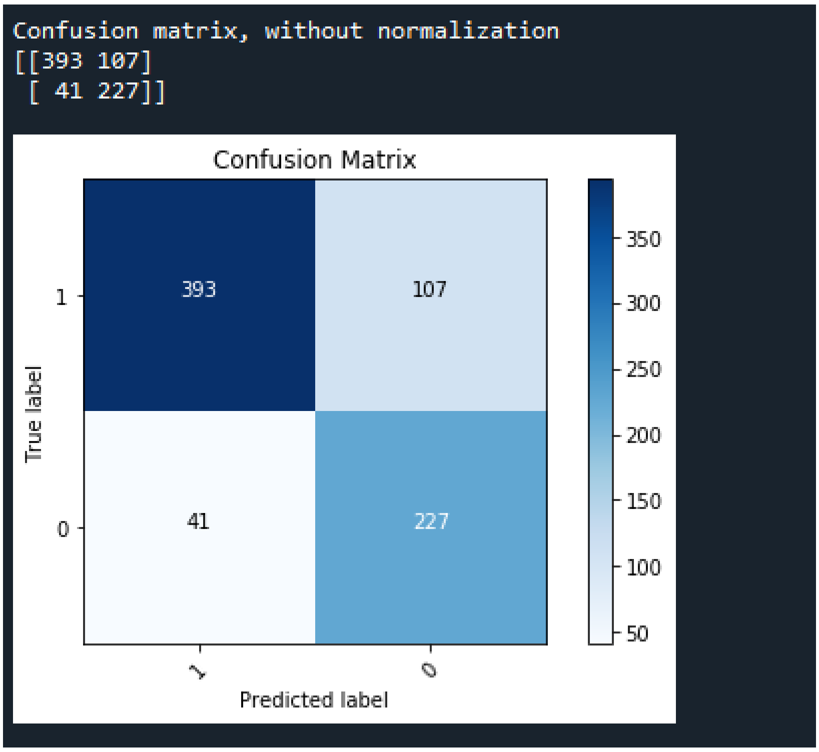
When running the model, the data percentage of data to be used for training and validation was defined using the “validation_split” parameter that Keras provides to the users. In this assignment, validation was defined as 10% and education as 90%. While the percentages were selected in this assignment, the most used percentages were investigated, and these percentages were selected(Draelos,2019).

As mentioned in part 1, all columns in this data set contain information that should have an impact on the classification process. Therefore, the columns other than the Outcome column are given as parameters to the classifier. The parameters given for the classifier are:

Pregnancies, Gender, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age, CalorieIntake, Exercise, SleepDuration

4. CONFUSION MATRIX

It can be said that confusion matrix is a table produced to see the performance of classification. The confusion matrix of the classification made in this assignment is given below. The function used for the visualization of the confusion matrix is taken from scikit learn (Scikit Learn, n.d).

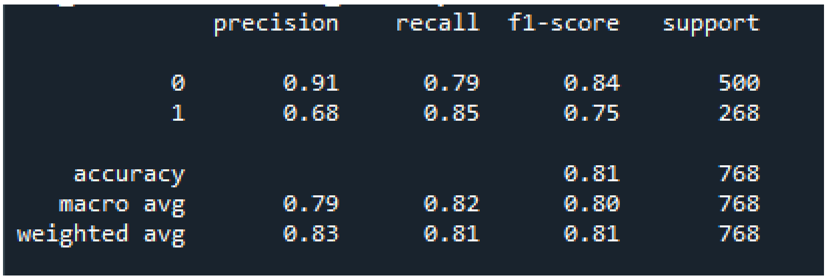


```
Confusion matrix, without normalization
[[393 107]
 [ 41 227]]
```

Figure 6 Confusion matrix

5. EVALUATION SCORES

Evaluation scores obtained at any time for the model produced are given in Figure 7.



```
              precision    recall  f1-score   support

           0       0.91      0.79      0.84       500
           1       0.68      0.85      0.75       268

    accuracy                           0.81       768
   macro avg       0.79      0.82      0.80       768
weighted avg       0.83      0.81      0.81       768
```

Figure 7 Evaluation scores

REFERENCES

Bag, S. (2020, October 6). *Which machine learning algorithm should you use by problem type?* Medium.

https://medium.com/analytics-vidhya/which-machine-learning-algorithm-should-you-use-by-problem-type-a53967326566

Draelos, R. (2019, September 15). *Best use of train/val/test splits, with tips for medical data*. Glass Box.

https://glassboxmedicine.com/2019/09/15/best-use-of-train-val-test-splits-with-tips-for-medical-data/#:~:text=Common%20ratios%20used%20are%3A,20%25%20val%2C%2020%25%20test

Kaggle (2021, April). *Diabetes Dataset.*

https://www.kaggle.com/dslearner0406/diabetes-dataset

Keras (n.d). *About keras.*

Retrieved May 24, 2021 from https://keras.io/about/

Scikit Learn (n.d) *Confusion matrix*

Retrieved May 24, 2021 from https://scikit-learn.org/0.18/auto_examples/model_selection/plot_confusion_matrix.html