FINAL PROJECT:DIABETES

CENG 3521, DATA MINING

Söz Tuana KURŞUN
Mert Furkan ERGÜDEN
Mehmet Kubilay ELÜSTÜ

Wednesday 20th January, 2021

Abstract

In this course and project we learned to write code with the help of some libraries and made it easier.The aim of this course is to classify data and transform it into meaningful data and to reveal valuable data for systems called decision support mechanisms in institutions.With this project, we learned to organize your data set and repair damaged data.Finally,we have observed the most accurate classification with many classification types.In short, we expect the machine to learn.In this way, when new patients arrive, they can be diagnosis by machine learning without examination.

1 Introduction

In this project we are going to realize, we will predict whether a diabetic patient will be sick or not. Based on the data kept in Excel; it is aimed to segment, group and organize patients.Some information is available in our data set to make this prediction.These are; Pregnancies, Glucose,Blood Pressure,Skin Thickness,Insulin,BMI,Diabetes Pedigree Function,Age and outcome output. This means that if outcome is 1 so patient is diabetic, 0 if it is not.

2 Assignments

If we roughly describe the tasks given in the homework, We will perform various classifications and observe which one gives better results.But with every step we take we have to dig deeper so keep reading :)

2.1 Assignment1: Editing Data

First we choose any project, after downloading the project we want, we call it on the command line and we can see the output and go into the details.We see a classification problem because our output consists of 1s and 0s.After calling our set, we call the df.info command to see more detail, so we can see the types and whether they are empty values.Then we check our values with the .isna().sum() command to see if we have missing values.Then we look at our data set to see if there is really missing data? and some things get caught in our eyes, like insulin and glucose values being 0.These values can not be 0 because if they are 0, it means that human is not performing life functions.So what does 0 mean? It means either wrong data or missing data.

Since we check the wrong data, these 0's are missing data and to check this, we can look at our data set by calling the .eq(0).sum() command.In order to correct these 0 values, we will write the code in the image and assign NaN values instead of 0s.Thus, we changed our table and then we assigned the average of those values by taking the average of each row instead of NaN values and then rewrite the code we used to check the 0's, so we check our new table and at the end of the part of organizing the data, we wanted to write the code you see in the picture to create a visual feast, so that it appeals to the eye.

```
#We called our dataset and visualized it with the data.head () command.
data=pd.read_csv('diabetes.csv')
data.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Figure 1: We called our dataset and visualized it with the data.head () command.

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Figure 2: We decided whether it is true or false by looking at the types in the data set.

```
data.isna().sum()
```

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

```
[ ] #Values like glucose,bloodpressure etc. can not be 0, we have to regulate them.
    data.eq(0).sum()
```

```
Pregnancies                 111
Glucose                       5
BloodPressure                35
SkinThickness               227
Insulin                     374
BMI                          11
DiabetesPedigreeFunction      0
Age                           0
Outcome                     500
dtype: int64
```

here we checked if there was an empty data. In the other, we checked the rows whose value is 0.Values like glucose,blood pressure etc. can not be 0, we have to regulate them.

```
[ ] #Of the 768 patients, 500 are not sick.
    data.shape
```

```
(768, 9)
```

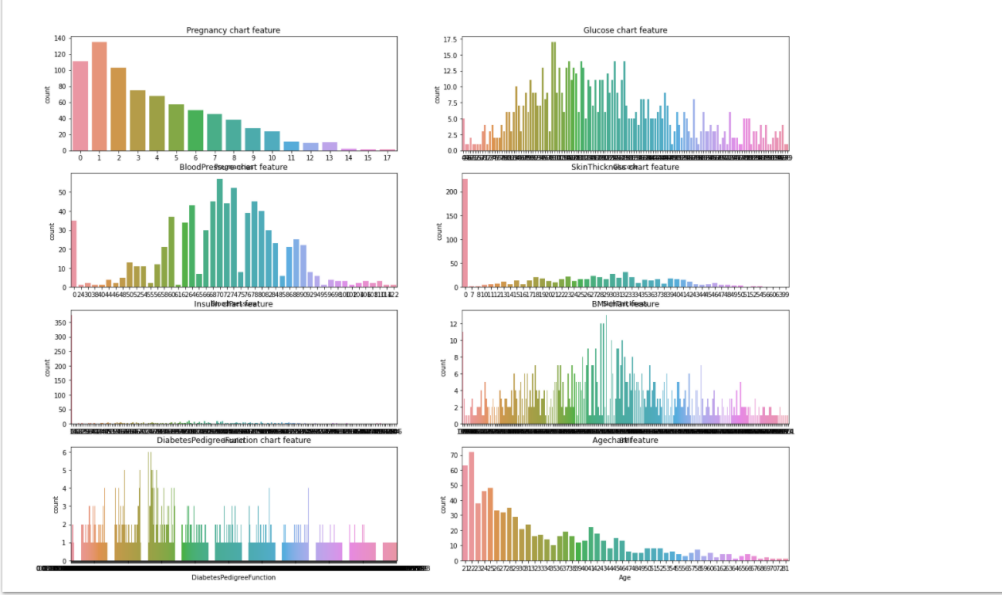
```
[ ] #scatter plots of features
    def scatter(ax,axis,name,title):
      sns.countplot(name,data=data,ax=ax[axis[0]][axis[1]])
      ax[axis[0],axis[1]].set_title(title)

    f,ax=plt.subplots(4,2,figsize=(20,15))
    plt.suptitle("Scatter plots of features")

    features = (((0,0),"Pregnancies","Pregnancy chart feature"),((0,1),"Glucose","Glucose chart feature"),((1,0),"BloodPressure","BloodPressure char
    ,((2,0),"Insulin","Insulin chart feature"),((2,1),"BMI","BMI chart feature"),((3,0),"DiabetesPedigreeFunction","DiabetesPedigreeFunction chart f

    for axis, name, title in features:
      scatter(ax,axis,name,title)
```

we showed the distribution plots of the features.And the last features for editing data data.shape,this code means we have the 768 patients and 500 are not sick.



[12]

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148.0	72.0	35.00000	155.548223	33.6	0.627	50	1
1	1	85.0	66.0	29.00000	155.548223	26.6	0.351	31	0
2	8	183.0	64.0	29.15342	155.548223	23.3	0.672	32	1
3	1	89.0	66.0	23.00000	94.000000	28.1	0.167	21	0
4	0	137.0	40.0	35.00000	168.000000	43.1	2.288	33	1

[13]
```
#we have no missing value.
data.isnull().sum()
```

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

[14]
```
#We have now organized our data set.
data.eq(0).sum()
```

```
Pregnancies                 111
Glucose                       0
BloodPressure                 0
SkinThickness                 0
Insulin                       0
BMI                           0
DiabetesPedigreeFunction      0
Age                           0
Outcome                     500
```

We organized all our data and 0 values are filled with means.

2.2 Assignment2: Correlation Analysis

Here we will examine the correlation analysis.What was the correlation? In short, it was the measure of the relationship between two variables.Here we will look at how much the goal affects the output, our outcome value. We can easily call this with a function inside the ”pandas.” We can see which variables are related to each other through the table. We use the heat map to show it better. We use the ”seaborn” library.

```
#The measure of the relationship between them.
data.corr()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.000000	0.127911	0.208522	0.082989	0.056027	0.021565	-0.033523	0.544341	0.221898
Glucose	0.127911	1.000000	0.218367	0.192991	0.420157	0.230941	0.137060	0.266534	0.492928
BloodPressure	0.208522	0.218367	1.000000	0.192816	0.072517	0.281268	-0.002763	0.324595	0.166074
SkinThickness	0.082989	0.192991	0.192816	1.000000	0.158139	0.542398	0.100966	0.127872	0.215299
Insulin	0.056027	0.420157	0.072517	0.158139	1.000000	0.166586	0.098634	0.136734	0.214411
BMI	0.021565	0.230941	0.281268	0.542398	0.166586	1.000000	0.153400	0.025519	0.311924
DiabetesPedigreeFunction	-0.033523	0.137060	-0.002763	0.100966	0.098634	0.153400	1.000000	0.033561	0.173844
Age	0.544341	0.266534	0.324595	0.127872	0.136734	0.025519	0.033561	1.000000	0.238356
Outcome	0.221898	0.492928	0.166074	0.215299	0.214411	0.311924	0.173844	0.238356	1.000000

Figure 3: In this table we expect high outcome values.

```
#We tried to show each feature among itself.
sns.heatmap(data.corr(),annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f587f0874a8>
```

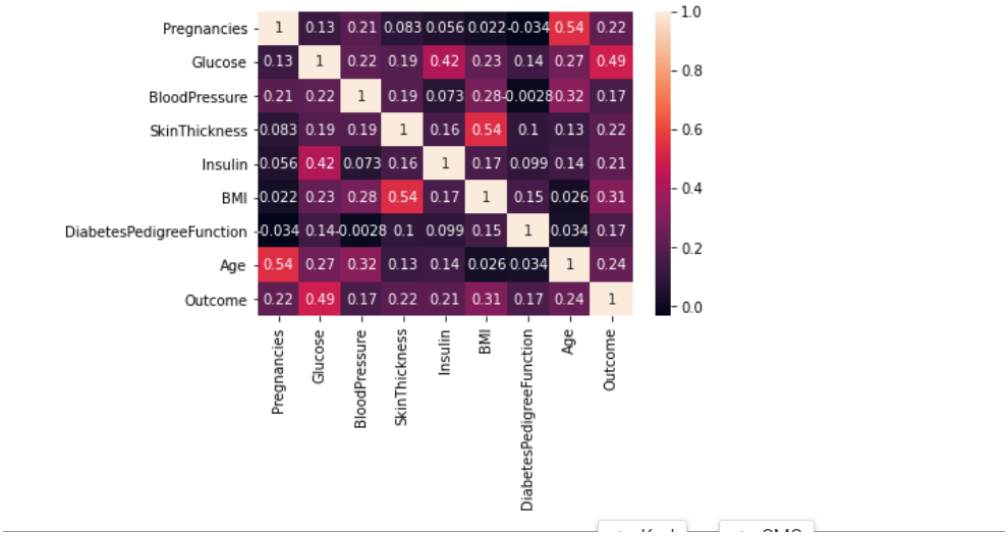


Figure 4: We tried to show each feature among itself.

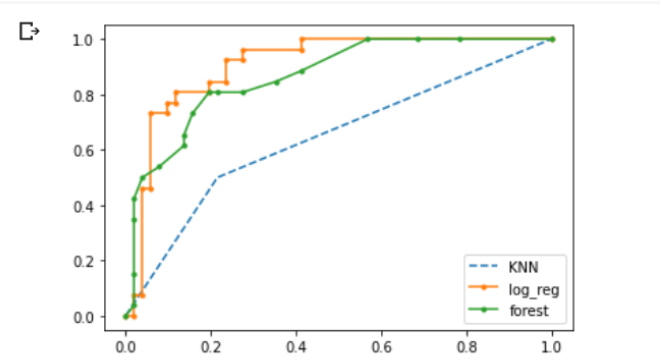
2.3 Assignment3: Classification Task

In the stage, we create the algorithms.We used 3 different types of classification. These are logistic regression,KNN and random forest.Using these three, we chose the classification that gave the

closest results and made a performance comparison.We create x and y, and our x values are what we enter and y is our output, that is, outcome.We apply the logistic regression random forest and KNN classifications to our data set, respectively.and we ”fit” and ”predict” each of them.Finally, we call the score and print the accuracy results on the screen.The more output in the score, the higher our prediction rate.
NOTE: Operations will appear in more detail in the code we write.

2.4 Assignment4: Compare Performance

In the last task, we compared performance between classifiers and saw which one gave more positive results.We had it plotted with pyplot.plot and we saw the result.



This is the most efficient analysis method, as the most area is under logistic regression.

2.5 Assignment5: Sub-tasks Requested From Us

First, we made clustering in our project, the purpose of which is that we made predictions about outcome, that is, diabetes or not, by connecting with other features without knowing the output,and then we printed the accuracy score. And then we write the codes for the importance order of the features that trigger diabetes and then drew them.Thus, we see which feature affects diabetes more.(is glucose)The last sub-task was to replace miss data, ie: data with 0 output, with linear regression.By doing Linear regression, we filled the 0 values once with mean, and this time we changed it by regression.

3 Results

As a result, we chose the health area, which is a frequently used area in our project. And we edited our data before taking action for forecast results.Since all our data is numerical, it has been easier and more efficient for us.We scanned missing or incorrectly entered data and corrected them, if any.After all our compilation processes were completed, we also had the heatmap drawn and visualized the relationships between them.After these processes were finished, we came to the actual process and created the algorithms and using the information we learned at the beginning of the data mining lesson here we expect the machine to learn.In this way, when new patients arrive, they can be diagnosis by machine learning without examination.

4 Conclusion

As a result, at the end of this assignment, as I mentioned at all stages,everything gets easier with machine learning. We make our lives easier with machine learning, especially in areas that are important such as health and where processes take time.Instead of doing a lot of tests as in the past, we can find out whether people are diabetes or not by looking at a few features, at least what we say is valid for this project.