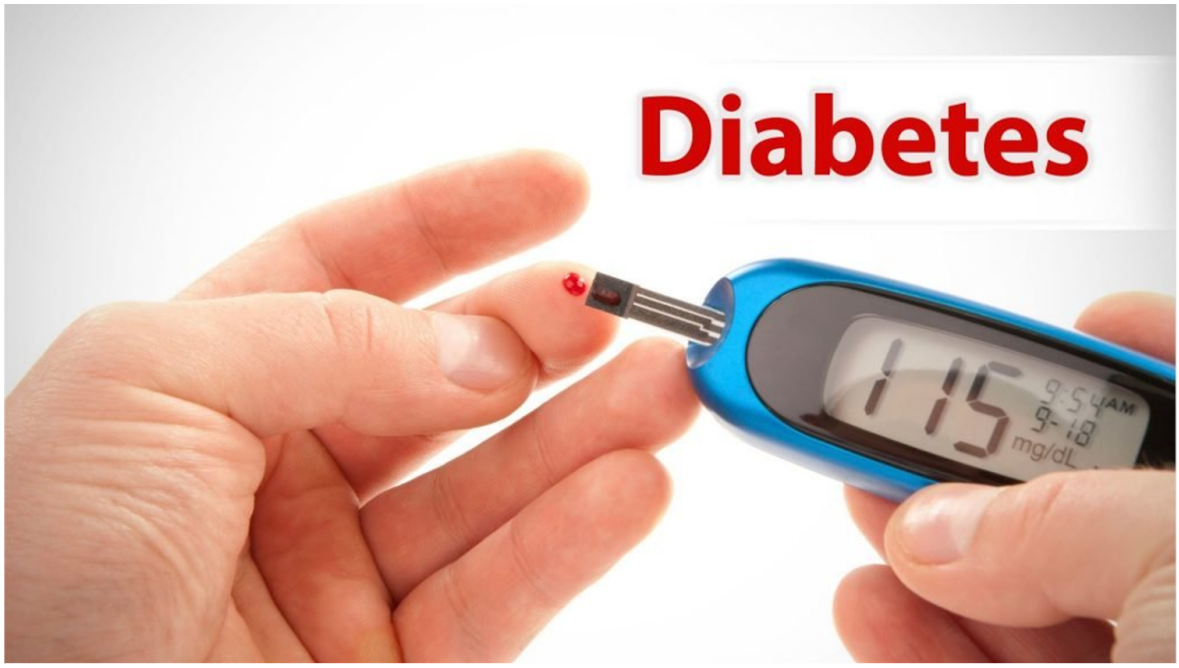


Detecting Diabetes Early

Daniel Kim

Problem Statement

Like any other disease, It is very important to detect the early symptoms of diabetes in order to be cured completely. Reportedly, there are two types of diabetes: type 1 and type 2.
The goal is to classify types of diabetes and their early symptoms so that people can cure diabetes, or prevent diabetes.

The Data

This data contains 17 attributes which include age, gender, and other symptoms.

1. **Target, y = 'Positive' in the column, 'class'**
2. **Other variables, Xs = all other features than y**
3. **Total number of rows = 521**
4. **Total number of columns = 17**

Data Wrangling

1. **Data Cleansing**

```
In [3]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 520 entries, 0 to 519
Data columns (total 17 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Age                 520 non-null    int64
 1   Gender              520 non-null    object
 2   Polyuria            520 non-null    object
 3   Polydipsia          520 non-null    object
 4   sudden weight loss  520 non-null    object
 5   weakness            520 non-null    object
 6   Polyphagia          520 non-null    object
 7   Genital thrush      520 non-null    object
 8   visual blurring     520 non-null    object
 9   Itching             520 non-null    object
 10  Irritability        520 non-null    object
 11  delayed healing     520 non-null    object
 12  partial paresis     520 non-null    object
 13  muscle stiffness    520 non-null    object
 14  Alopecia            520 non-null    object
 15  Obesity             520 non-null    object
 16  class               520 non-null    object
dtypes: int64(1), object(16)
memory usage: 69.2+ KB
```

   a. **Null Value, NaN**

There is no Null Values.

   b. **Head Samples**

```
In [4]: df.head()
```

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	muscle stiffness	Alopecia	Obesity	class
0	40	Male	No	Yes	No	Yes	No	No	No	Yes	No	Yes	No	Yes	Yes	Yes	Positive
1	58	Male	No	No	No	Yes	No	No	Yes	No	No	No	Yes	No	Yes	No	Positive
2	41	Male	Yes	No	No	Yes	Yes	No	No	Yes	No	Yes	No	Yes	Yes	No	Positive
3	45	Male	No	No	Yes	Yes	Yes	Yes	No	Yes	No	Yes	No	No	No	No	Positive
4	60	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Positive

As seen above, each columns shows binary responses like ‘Male’ or ‘Female’, or ‘Yes’ or ‘No’, or ‘Positive’ or ‘Negative’ except for age. The dataset seems so organized that it can be moved to the next step, one-hot encoding.

c. One-Hot Encoding

```
df.head()
```

	Age	Gender_Female	Gender_Male	Polyuria_No	Polyuria_Yes	Polydipsia_No	Polydipsia_Yes	sudden weight loss_No	sudden weight loss_Yes	weakness_No	weakness_Yes	Polyphagia_No	Polyphagia_Yes	Genital thrush_No	Genital thrush_Yes	visu blurring_
0	40	0	1	1	0	0	1	1	0	0	1	1	0	1	0	1
1	58	0	1	1	0	1	0	1	0	0	1	1	0	1	0	0
2	41	0	1	0	1	1	0	1	0	0	1	0	1	1	0	1
3	45	0	1	1	0	1	0	0	1	0	1	0	1	0	1	1
4	60	0	1	0	1	0	1	0	1	0	1	0	1	1	0	0

Such features from Gender to Class are binary (Male or Female, Yes or No, or Positive or Negative). Thus let's remove such all columns ending with No or Negative.

d. Removing the other binary columns

```
for col in df.columns:
    if '_No' in col:
        del df[col]
df
```

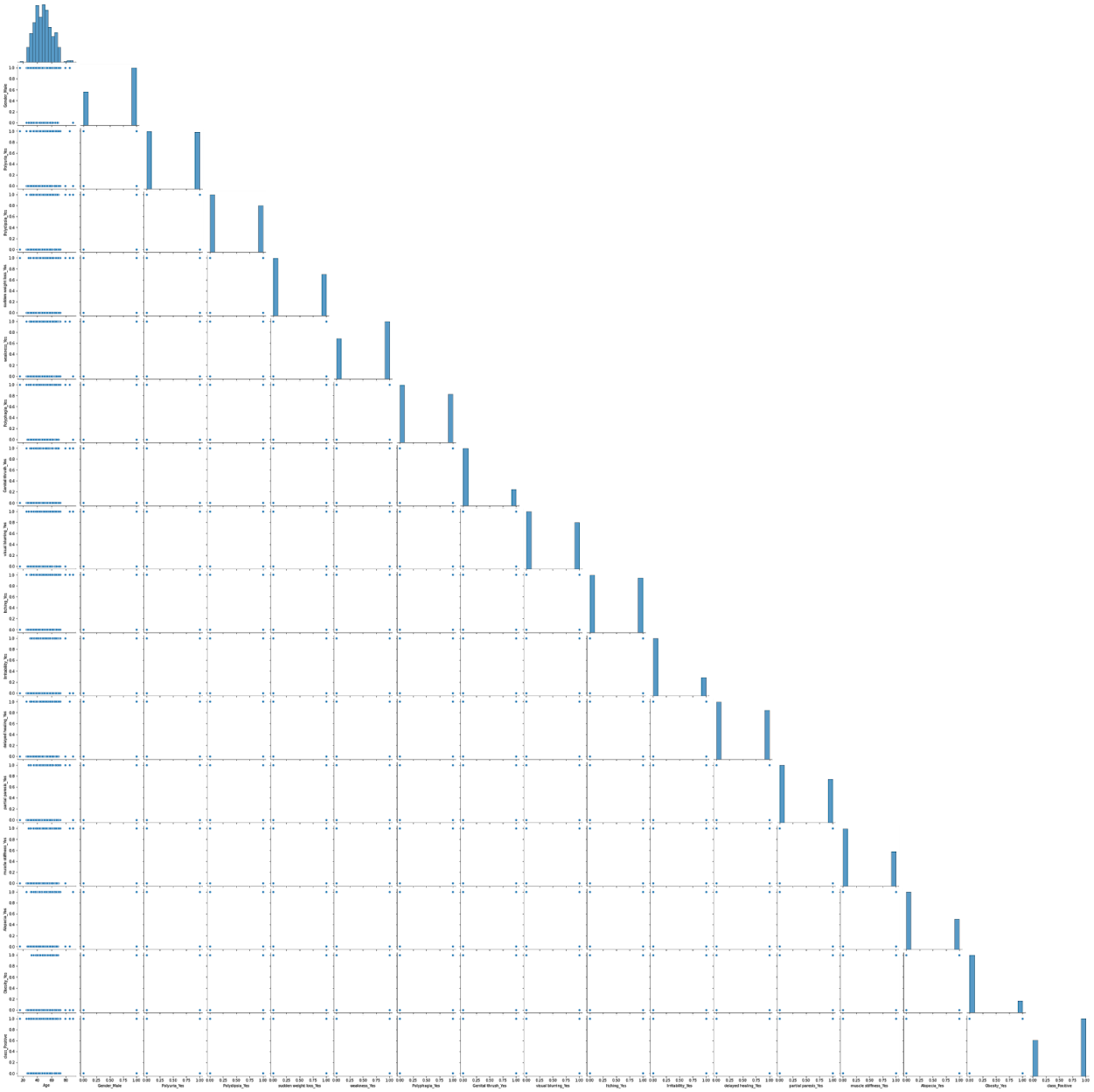
	Age	Gender_Female	Gender_Male	Polyuria_Yes	Polydipsia_Yes	sudden weight loss_Yes	weakness_Yes	Polyphagia_Yes	Genital thrush_Yes	visual blurring_Yes	Itching_Yes	Irritability_Yes	delayed healing_Yes	partial paresis_Yes	muscle stiffness_Yes	Alo
0	40	0	1	0	1	0	1	0	0	0	1	0	1	0	1	1
1	58	0	1	0	0	0	1	0	0	1	0	0	0	1	0	1
2	41	0	1	1	0	0	1	1	0	0	1	0	1	0	1	1
3	45	0	1	0	0	1	1	1	1	0	1	0	1	0	0	0
4	60	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
515	39	1	0	1	1	1	0	1	0	0	1	0	1	1	0	0
516	48	1	0	1	1	1	1	1	0	0	1	1	1	1	0	0
517	58	1	0	1	1	1	1	1	0	1	0	0	0	1	1	0
518	32	1	0	0	0	0	1	0	0	1	1	0	1	0	0	1
519	42	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

520 rows × 19 columns

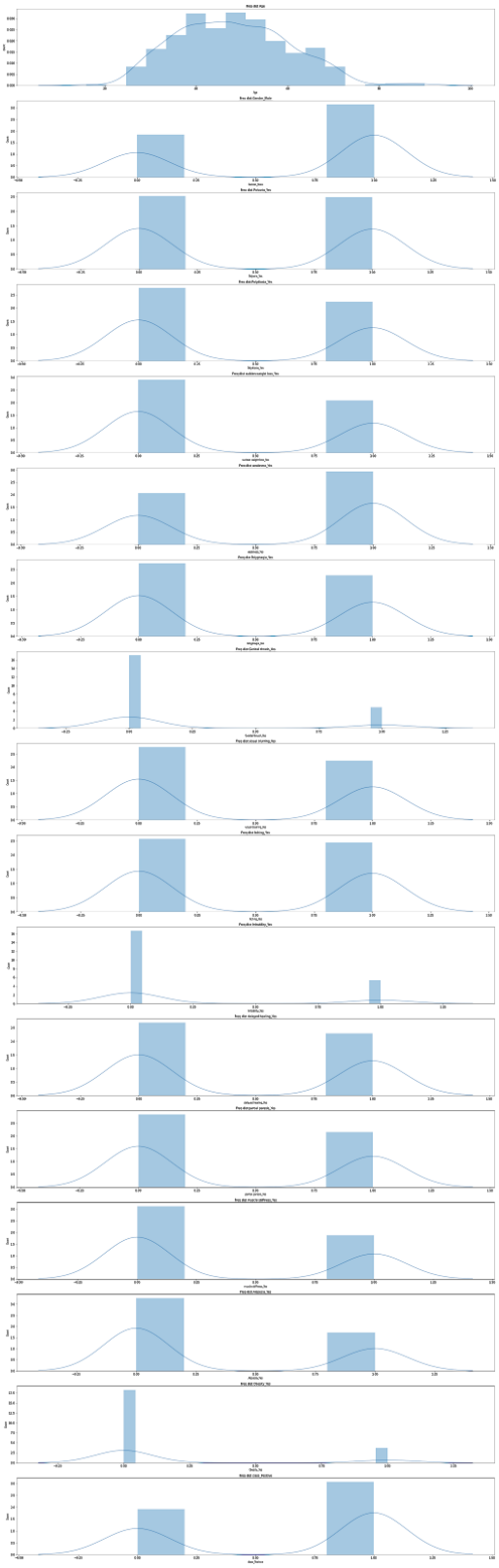
By using for loops as seen above, the other columns ending with '_Female', '_Negative', or '_No' were deleted not only for reducing data size but also for efficiency in running algorithms.

Exploratory Data Analysis

1. Pairplot

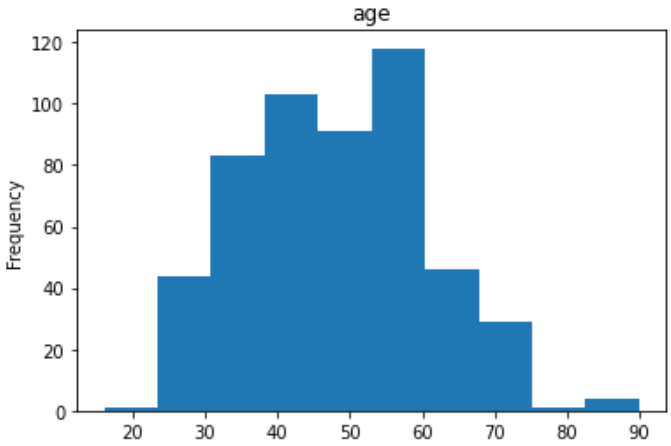


2. Histogram

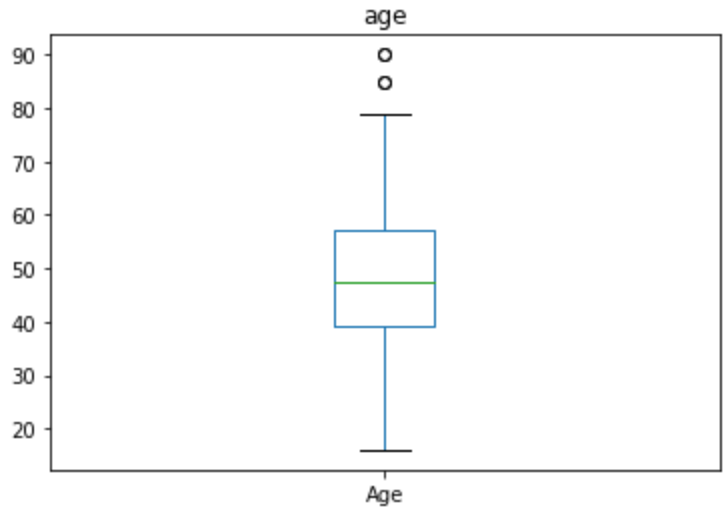


Except for the first feature, ‘age’, all the other do now show normal distribution because they are all binary-typed data.

‘Age’ seems normaly distributed.



3. **Boxplot for Age**



The boxplot of ‘Age’ shows that there are two outliers.

Surely, the outliers can be detected as outliers when the outlier detection algorithm is applied. So the next step will be the outlier detection by Isolation Forest.

Conclusion

- **Just for confirmation purpose, maybe, ‘Isolation Forest’ algorithm will be applied to remove outliers.**

- **By using different machine learning algorithms, compare important features, classification accuracies, and other insights.**