Dataset Details

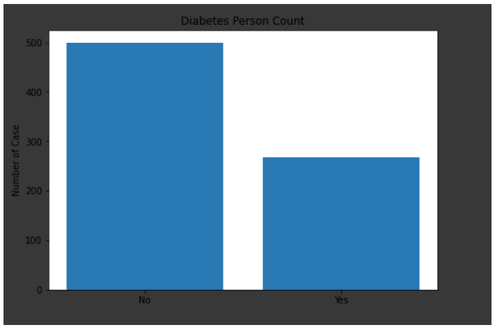
Pima Indian Diabetes Dataset
Based on the medical measurements used in the dataset, the purpose of the dataset is to diagnostically determine whether or not a person has diabetes. Multiple medical prediction variables and one outcome attribute compose datasets. Predictor factors include the patient's number of births, their BMI, level of insulin, age, and so on.This dataset consists of 768 rows and 9 column.

Digit Recognizer
Each image has a height of 28 pixels and a width of 28 pixels, for a total of 784 pixel. There is a single pixel-value correlated with each pixel, showing the pixel's lightness or darkness, with higher numbers suggesting darker. This pixel-value is an integer ranging from 0 to 255, inclusive.The data set for training, has 785 columns. The first column is the digit that was drawn by the individual, labeled "label". The remainder of the columns contain the corresponding picture pixel values.This dataset consists of 42000 rows and 784 column of pixel.

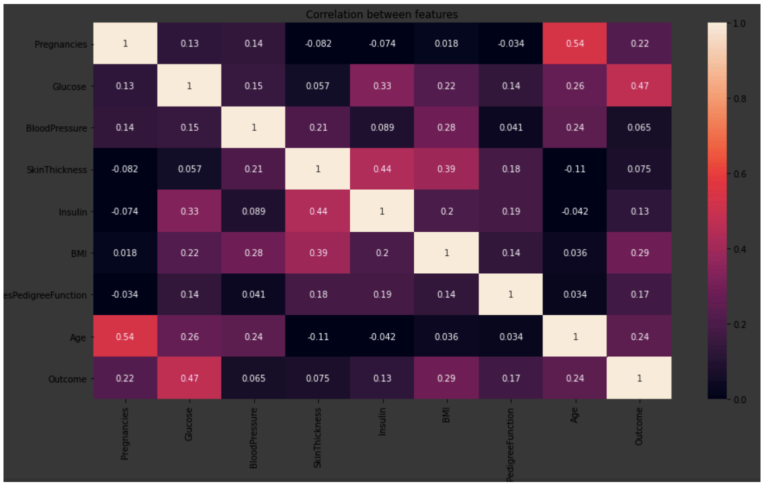
Data Visualization

Pima Indian Diabetes Dataset

Outcome
Class variable (0 or 1) 268 of 768 are 1, the others are 0.There are not as many people with diabetes as people with non-diabetes. Because of a limited number of cases, it is popular in medical diagnosis. We call it as Imbalanced data.

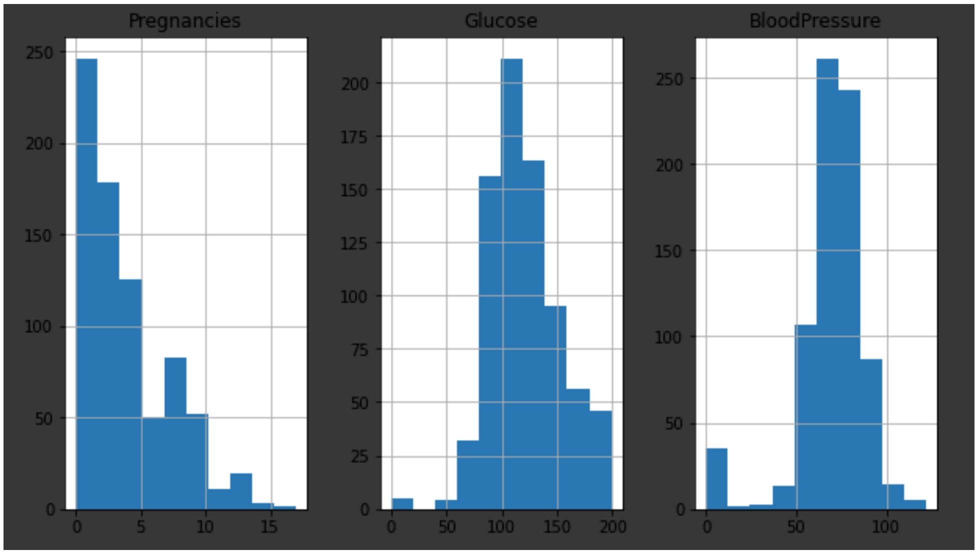


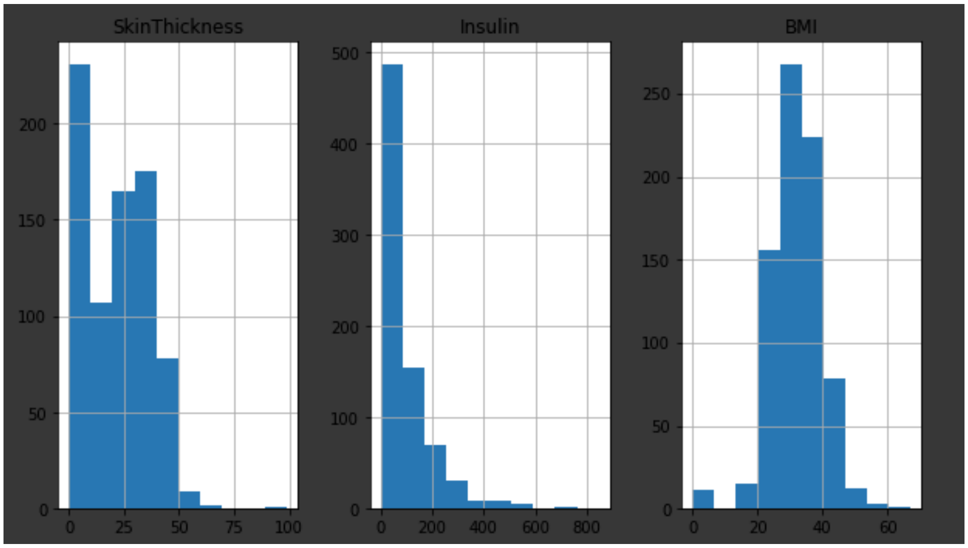
Correlation Between Features
An significant part of the method of data processing is the testing of associations. This analysis is one of the tools used to assess which attributes most influence the target variable, and is used in the prediction. Here we found that Glucose, BMI, Age and then Pregnancies are main feature which plays important role in prediction of outcome variable in this dataset.

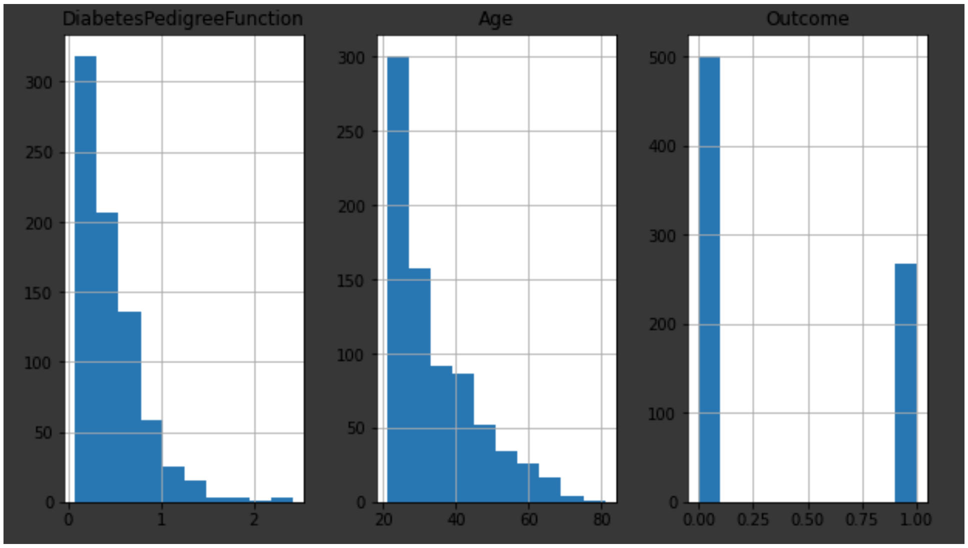


Complete Data Distribution

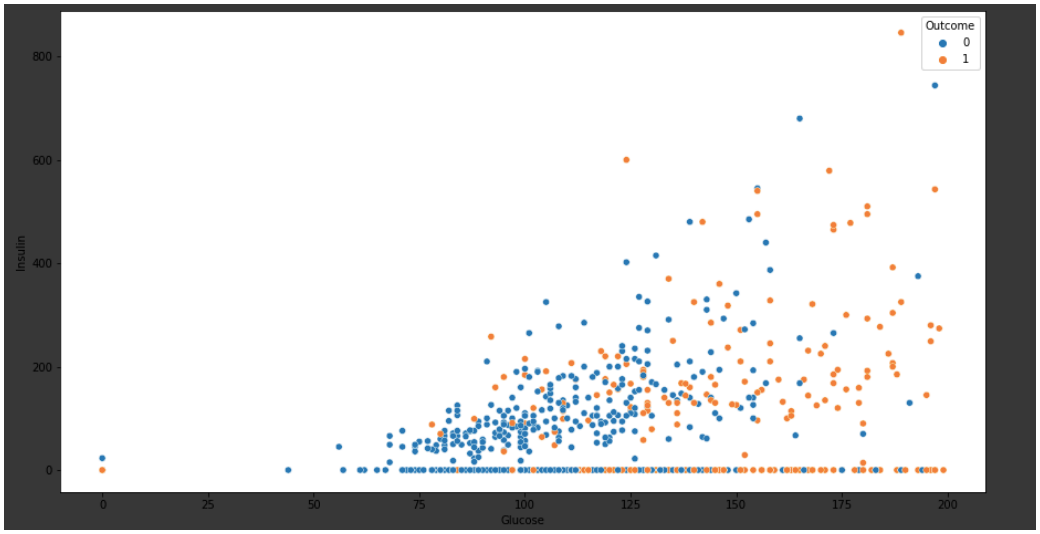
This is a complete representation of the distribution of data provided to us in histogram format



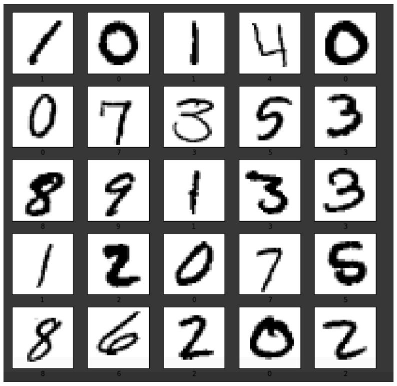




By the plot below we can see that the person whose glucose level is high are most likely to be diabetic. As well as we can also say that person having both insulin level and glucose level high are most likely to be diabetic



Digit Recognizer



This dataset consists of handwritten digits from 1-9.

Train-Test Split of Data

The train-test split is a technique for testing a machine learning algorithm's output. This involves a dataset being taken and divided into two parts. First part is used to train/fit the model and is referred to as the training dataset. The second part is not used to train the algorithm instead, it is used to give input element of the dataset, then predictions are made and the predicted values are compared. This second dataset is known as test dataset.

In this dataset I have choose a 20% split percentage. That is 80% would be my training set and 20% would be my test set. I have taken this split by considering several parameters like computational cost in both training and testing the model. As well as representativeness of training set.

Algorithm Description

KNN Algorithm
In K, meaning algorithm, we would look at the K nearest training data points for each test data point and take the most frequently occurring classes and assign the test data to that class. K reflects the sum of training data points that lie near the evaluation data point that we are going to use to locate the class.

1. Load the training and test data
2. Choose the value of K
3. For each point in test data:

```
        - find the distance to all training data points
        - store the distances in a list and sort it
        - choose the first k points
        - assign a class to the test point based on the majority of classes present in the chosen points
4. End
```

Feature Scaling

Standardization
This is used in PIMA dataset only
The algorithm should not be biased towards variables with higher magnitude. To overcome this problem, we can bring down all the variables to the same scale. One of the most common technique to do so is normalization where we calculate the mean and standard deviation of the variable. Then for each observation, we subtract the mean and then divide by the standard deviation of that variable:
x = (x-u)/sigma

$$X' = \frac{X - \mu}{\sigma}$$

Where x is the original feature vector, 'u' is the mean of that feature vector, and sigma is its standard deviation.

Distance Metrics

Euclidean distance (Used in both PIMA and Digit recognizer dataset)
The Euclidean distance between two points in either the plane or 3-dimensional space measures the length of a segment connecting the two points. It is the most obvious way of representing distance between two points.

The Pythagorean Theorem can be used to calculate the distance between two points, as shown in the figure below. If the points (x1,y1) and (x2,y2) are in 2-dimensional space, then the Euclidean distance between them

$$\begin{aligned} d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) &= \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} \\ &= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}. \end{aligned}$$

Manhattan Distance
The Manhattan distance between two vectors (city blocks) is equal to the one-norm of the distance between the vectors. The Manhattan distance as the sum of absolute differences
Manhattan Distance [{a, b, c}, {x, y, z}] = Abs [a − x] + Abs [b − y] + Abs [c − z]

Chebyshev distance
The Chebyshev distance calculation, commonly known as the "maximum metric" in mathematics, measures distance between two points as the maximum difference over any of their axis values. In a 2D grid, for instance, if we have two points (x1, y1), and (x2, y2), the Chebyshev distance between is max(y2 - y1, x2 - x1).

$$D_{\text{Chebyshev}}(x, y) := \max_i(|x_i - y_i|).$$

Algorithm Results

Pima Indian Diabetes Dataset

Results without doing Feature Engineering

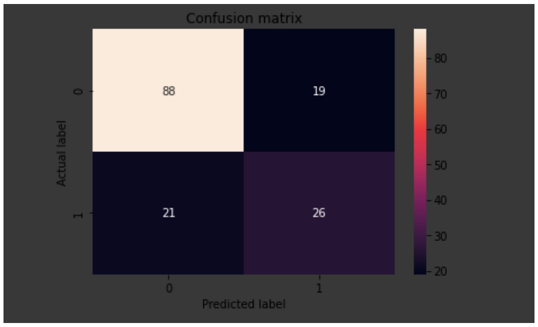
S.no.	Distance Metrics	True Positive	True Negatives	False Positive	False Negative	Accuracy
1.	Chebyshev	83	28	24	19	74.6%

2.	Manhattan	88	31	19	16	74.02%
3.	Euclidean	95	34	12	13	74.02%

Results after doing Feature Engineering

S.no.	Distance Metrics	True Positive	True Negatives	False Positive	False Negative	Accuracy
1.	Chebyshev	83	28	24	19	72%
2.	Manhattan	88	31	19	16	77%
3.	Euclidean	95	34	12	13	83%

Confusion Matrix (Using Euclidean Distance)



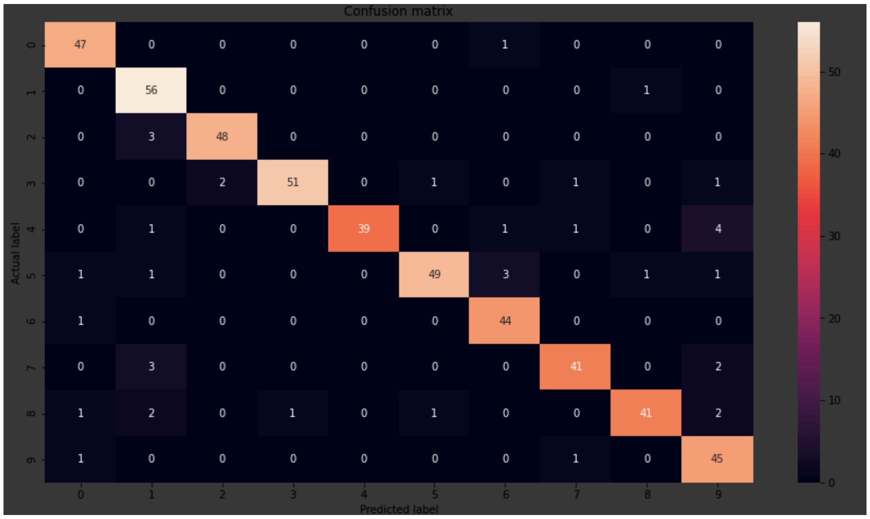
Runtime of the program is 0.88 sec

Digit Recognizer

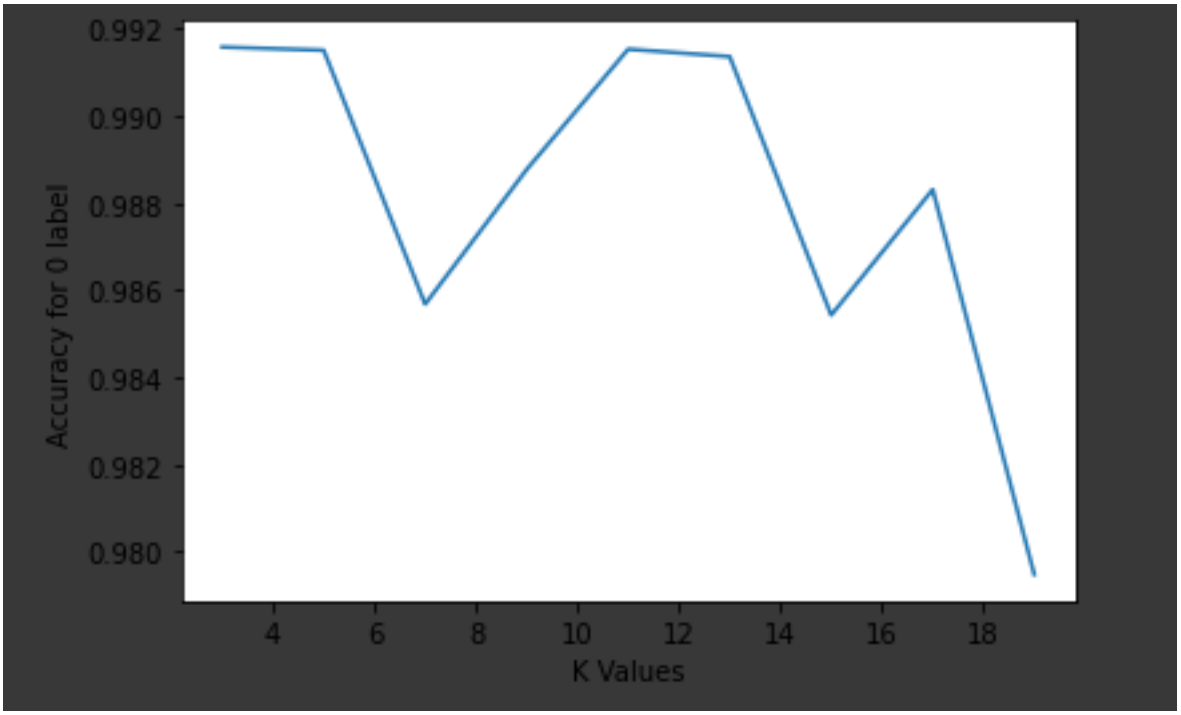
Overall Accuracy : 93%

Runtime of the program is 167.81 sec for taking 10000 rows from training data.

Confusion Matrix



K Values VS Accuracy for label “0”


Accuracy for 0 label
K Values