Machine Learning, Fall 2019: Project 1

Liam Li G48502460

Head:

Python: pandas and matplotlib modules for visualization

Sublime text 3

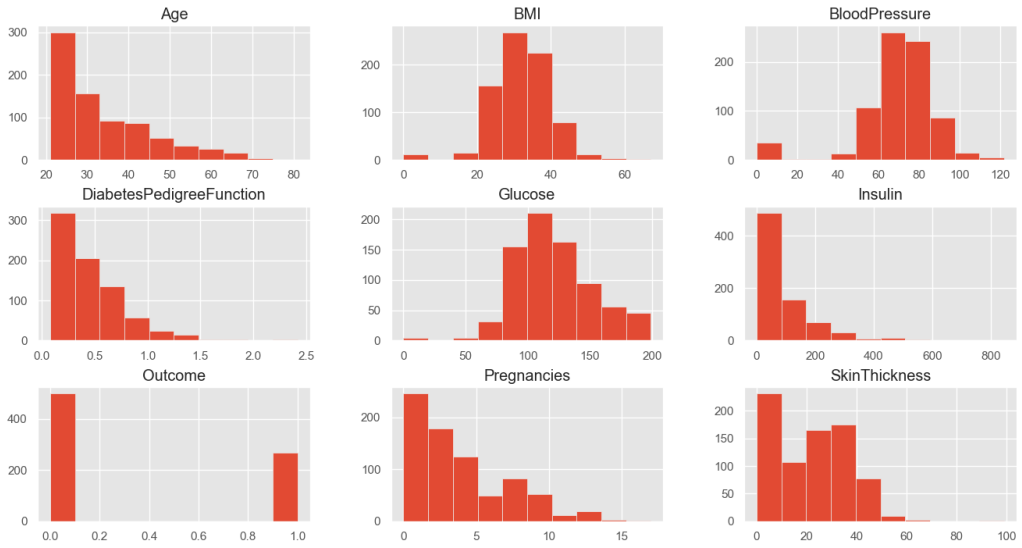
Pima Dataset

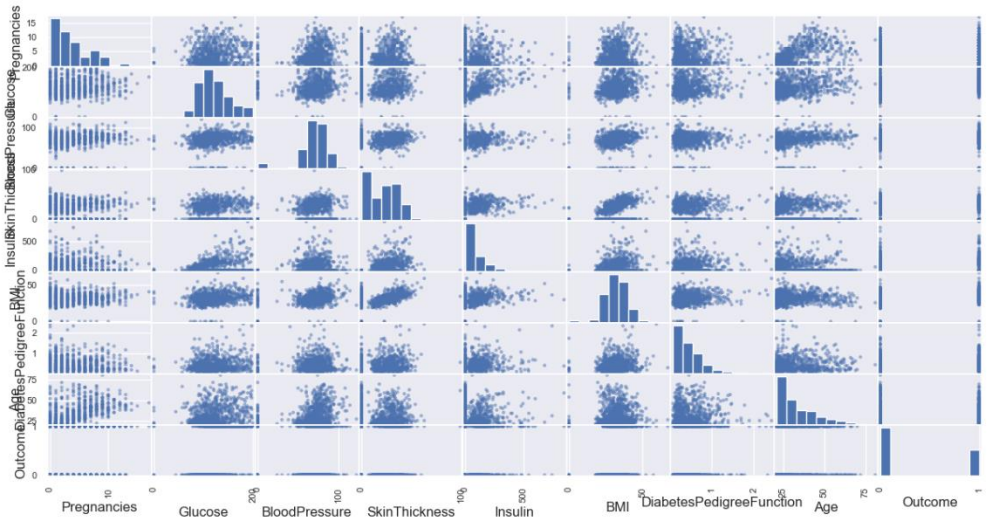
Dataset details:

This dataset contains eight medical factors that may affect diabetes, and one outcome, which indicates that whether the unit has diabetes. More details show in the image below.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
Pregnancies                 768 non-null int64
Glucose                     768 non-null int64
BloodPressure               768 non-null int64
SkinThickness               768 non-null int64
Insulin                     768 non-null int64
BMI                         768 non-null float64
DiabetesPedigreeFunction    768 non-null float64
Age                         768 non-null int64
Outcome                     768 non-null int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
None
```

The histogram represents the distribution of each column.





For this dataset, I split the data into training set, validation set and test set as 80%, 10% and 10%.

Algorithm description:

This project is a classifer based on K-NN algorithm.

First, It transfers csv data file into a 2D array by readfile() function.

Then it splits data into into training set, validation set and test set as 80%, 10% and 10% by split() and split_scale(), split_scale() function is used to guarantee the proportion of labels in each splitted data is same as the proportion in the whole dataset.

It doesn't include feature scaling considering this dataset is relatively small.

Three distance metrics are used to compare the result, which are Manhattan distance(p=1), Euclidean distance(p=2), and p=4.

Knn(target,d,k) returns a prediction given by the majority of k neighbors.

Loss(data,k) returns the error rate on data.

Evaluation(data,ks) shows different error rate when k in range(1,ks=15), by observation of the result we can obtain the optimal k for the classfier.

Algorithm results:

Manhattan distance(p=1)

```
k   loss
1 0.6493506493506493
2 0.35064935064935066
3 0.37662337662337664
4 0.33766233766233766
5 0.35064935064935066
6 0.3116883116883117
7 0.33766233766233766
8 0.3116883116883117
9 0.2727272727272727
10 0.2727272727272727
11 0.2597402597402597
12 0.23376623376623376
13 0.2597402597402597
14 0.2857142857142857
15 0.2857142857142857
optimal k= 12
loss of test set is 0.22077922077922077
('false positive', 9)
('false negative', 8)
('true positive', 18)
('true negative', 42)
[Finished in 5.0s]
```

	False	True	
Positive	9	18	27
Negitive	8	42	50
	17	60	

Euclidean distance(p=2)

```
k   loss
1 0.6493506493506493
2 0.38961038961038963
3 0.44155844155844154
4 0.3246753246753247
5 0.3246753246753247
6 0.2857142857142857
7 0.24675324675324675
8 0.24675324675324675
9 0.2857142857142857
10 0.2727272727272727
11 0.2727272727272727
12 0.2597402597402597
13 0.2727272727272727
14 0.2727272727272727
15 0.2727272727272727
optimal k= 7
loss of test set is 0.2857142857142857
('false positive', 14)
('false negative', 8)
('true positive', 13)
('true negative', 42)
[Finished in 5.6s]
```

	False	True	
Positive	14	13	27
Negitive	8	42	50
	22	55	

p=4.

```
k   loss
1 0.6493506493506493
2 0.37662337662337664
3 0.4155844155844156
4 0.33766233766233766
5 0.35064935064935066
6 0.2597402597402597
7 0.24675324675324675
8 0.23376623376623376
9 0.2597402597402597
10 0.2597402597402597
11 0.23376623376623376
12 0.24675324675324675
13 0.23376623376623376
14 0.2727272727272727
15 0.2597402597402597
optimal k= 8
loss of test set is 0.35064935064935066
('false positive', 10)
('false negative', 17)
('true positive', 17)
('true negative', 33)
[Finished in 6.3s]
```

	False	True	
Positive	10	17	27
Negitive	17	33	50
	27	50	

Runtime

Time complexity is O(nd), n is the size of training data, d is the number of features.

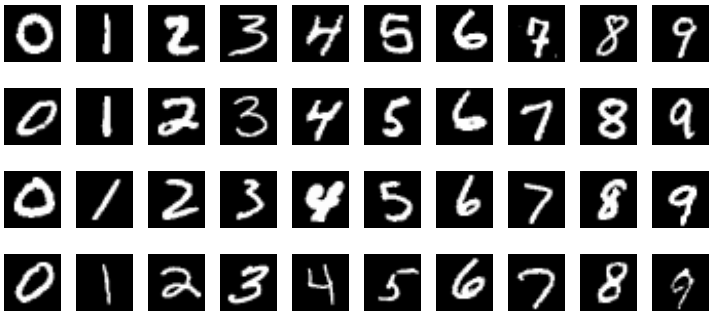
Wall-clock time is presented in the images of the result part (including loading data file and splitting data).

MNIST Dataset

Dataset details:

"The data files train.csv and test.csv contain gray-scale images of hand-drawn digits, from zero through nine."

"Each image is 28 pixels in height and 28 pixels in width, for a total of 784 pixels in total. Each pixel has a single pixel-value associated with it, indicating the lightness or darkness of that pixel, with higher numbers meaning darker. This pixel-value is an integer between 0 and 255, inclusive."



Algorithm description:

This project is a classifer based on K-NN algorithm.

First, It transfers train.csv data file into a 2D array by readfile() function. Due to limited time and computing performance, I shrink the dataset 15 times smaller. The size of training data is 2240.

Then it splits data into into training set and validation set as 80%, 20% by split(), and test set is originally separated as test.csv.

Distance metrics is Euclidean distance(p=2).

Knn(target,d,k) returns a prediction given by the majority of k neighbors.

Loss(data,k) returns the error rate on data.

Evaluation(data,ks) shows different error rate when k in (3,6,9,12,15), by observation of the result we can obtain the optimal k for the classfier.

Algorithm results:

When k=3, loss is 15%. Matrix[i][j] is the number of the scenario that actual digit is I and predicted digit is j. The size of validation set is 560.

```
loss of validation set is 0.14642857142857144
[64, 1, 0, 0, 0, 0, 3, 0, 0, 0]
[0, 66, 2, 0, 0, 0, 0, 0, 0, 1]
[3, 1, 41, 0, 1, 0, 2, 0, 0, 0]
[1, 2, 1, 52, 1, 7, 0, 1, 2, 1]
[0, 1, 0, 0, 48, 0, 0, 0, 0, 12]
[0, 3, 0, 2, 1, 36, 2, 0, 1, 0]
[1, 2, 0, 0, 0, 0, 44, 0, 0, 0]
[0, 0, 0, 0, 1, 0, 0, 45, 0, 3]
[0, 4, 0, 1, 0, 3, 1, 0, 38, 2]
[3, 0, 0, 1, 5, 1, 0, 2, 0, 44]
```

When k=6, loss is 11%. Matrix[i][j] is the number of the scenario that actual digit is I and predicted digit is j. The size of validation set is 560.

```
loss of validation set is 0.11071428571428571
[63, 1, 0, 0, 0, 0, 3, 0, 1, 0]
[0, 67, 2, 0, 0, 0, 0, 0, 0, 0]
[1, 1, 40, 0, 1, 1, 3, 0, 1, 0]
[0, 2, 1, 57, 1, 4, 0, 0, 2, 1]
[0, 1, 0, 0, 59, 0, 0, 0, 0, 1]
[0, 3, 0, 1, 1, 37, 3, 0, 0, 0]
[0, 2, 0, 0, 0, 0, 45, 0, 0, 0]
[0, 0, 0, 0, 1, 0, 0, 47, 0, 1]
[0, 5, 0, 0, 1, 3, 2, 0, 36, 2]
[1, 1, 0, 2, 2, 0, 0, 3, 0, 47]
```

When k=9, loss is 12%. Matrix[i][j] is the number of the scenario that actual digit is I and predicted digit is j. The size of validation set is 560.

```
loss of validation set is 0.12142857142857143
[62, 1, 0, 0, 0, 0, 5, 0, 0, 0]
[0, 67, 2, 0, 0, 0, 0, 0, 0, 0]
[0, 1, 42, 0, 1, 1, 2, 0, 0, 1]
[0, 2, 1, 58, 1, 3, 0, 1, 2, 0]
[0, 1, 0, 0, 51, 0, 0, 0, 0, 9]
[0, 3, 0, 1, 1, 38, 2, 0, 0, 0]
[0, 2, 0, 0, 0, 1, 44, 0, 0, 0]
[0, 0, 0, 0, 1, 1, 0, 46, 0, 1]
[0, 6, 0, 0, 1, 2, 1, 0, 36, 3]
[1, 1, 0, 2, 2, 0, 0, 2, 0, 48]
```

When k=12, loss is 12%. Matrix[i][j] is the number of the scenario that actual digit is I and predicted digit is j. The size of validation set is 560.

```
loss of validation set is 0.12321428571428572
[63, 1, 0, 0, 0, 0, 4, 0, 0, 0]
[0, 67, 2, 0, 0, 0, 0, 0, 0, 0]
[0, 1, 41, 0, 1, 1, 2, 0, 1, 1]
[0, 2, 0, 59, 0, 3, 0, 1, 2, 1]
[0, 1, 0, 0, 50, 0, 0, 0, 0, 10]
[0, 4, 0, 1, 1, 37, 1, 0, 1, 0]
[0, 2, 0, 0, 0, 1, 44, 0, 0, 0]
[0, 0, 0, 0, 1, 1, 0, 46, 0, 1]
[0, 7, 0, 0, 0, 1, 2, 0, 35, 4]
[1, 1, 0, 2, 2, 0, 0, 1, 0, 49]
```

Runtime

Time complexity is O(nd), n is the size of training data, d is the number of features.

Wall-clock time is presented in the images of the result part(including loading data file and splitting data).

Reference

1. https://www.kaggle.com/uciml/pima-indians-diabetes-database
2. https://www.kaggle.com/c/digit-recognizer/data
3. https://www.kaggle.com/shrutimechlearn/step-by-step-diabetes-classification-knn-detailed

1. $H(Y) = -P_+ \log_2 P_+ - P_- \log_2 P_-$

$= -\frac{4}{7}\log_2\frac{4}{7} - \frac{3}{7}\log_2\frac{3}{7} = 0.98$

2. $IG(X_1) \equiv H(*Y) - H(Y|X_1)$

$H(Y) - \frac{8}{21}H(Y|X_1=T) - \frac{13}{21}H(Y|X_1=F)$

$H(Y|X_1=T) = -\frac{7}{8}\log_2\frac{7}{8} - \frac{1}{8}\log_2\frac{1}{8} = 0.54$

$H(Y|X_1=F) = -\frac{5}{13}\log_2\frac{5}{13} - \frac{8}{13}\log_2\frac{8}{13} = 0.96$

$IG(X_1) = 0.98 - 0.54 \times \frac{8}{21} - 0.96 \times \frac{13}{21}$

$= 0.18$

$IG(X_2) \equiv H(Y) - \frac{10}{21}H(Y|X_2=T) - \frac{11}{21}H(Y|X_2=F)$

$H(Y|X_2=T) = -\frac{7}{10}\log_2\frac{7}{10} - \frac{3}{10}\log_2\frac{3}{10} = 0.88$

$H(Y|X_2=F) = -\frac{5}{11}\log_2\frac{5}{11} - \frac{6}{11}\log_2\frac{6}{11} = 0.99$

$IG(X_2) = 0.98 - 0.88 \times \frac{10}{21} - 0.99 \times \frac{11}{21}$

$= 0.04$

3.