0310120 陳家煒

Introduction to Machine Learning HW3

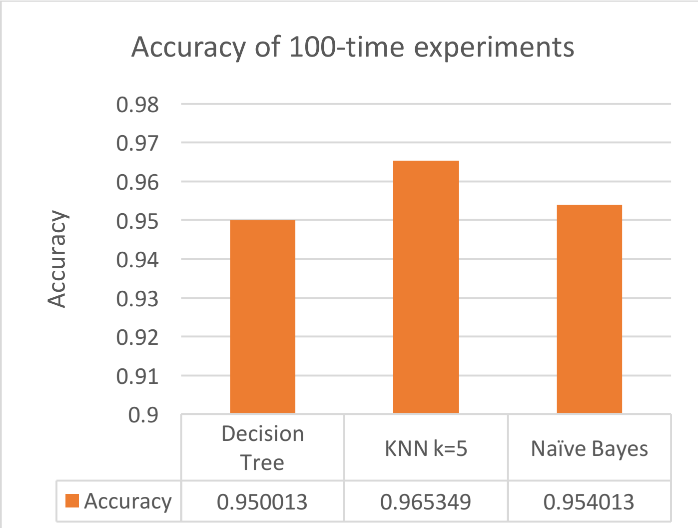
- Task：According to the class, we know what Decision Tree, K- nearest neighbor and naïve Bayes do. This time we use different classifiers/regressors to analyze two data sets (Iris.csv / forestfires.csv)
- Environment：OSX MAC、Ubuntu 16.04.3 LTS
- Language：Python 2.7.12, jupyter notebook
- Library：numpy, pandas, sklearn, seaborn, matplotlib

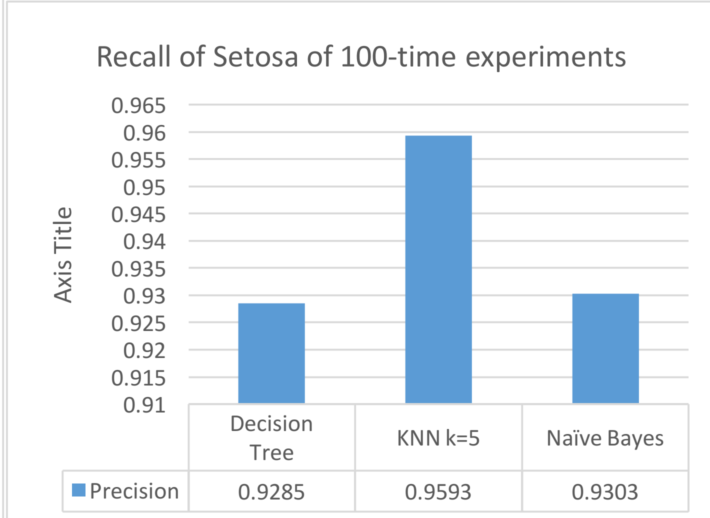
Dataset 1 : Iris.csv

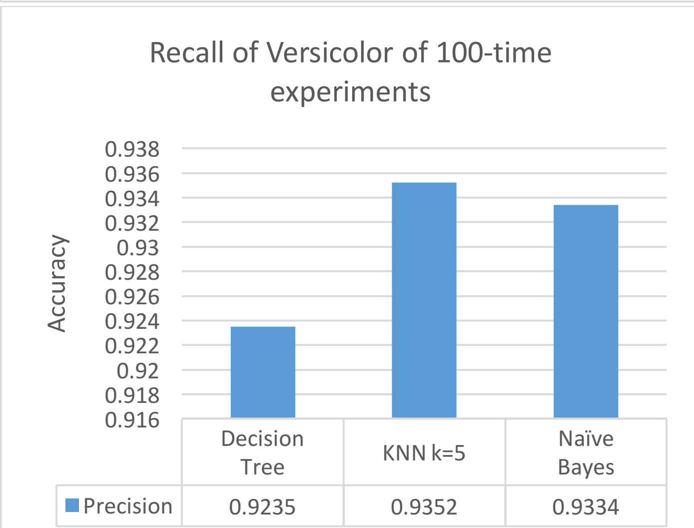
	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

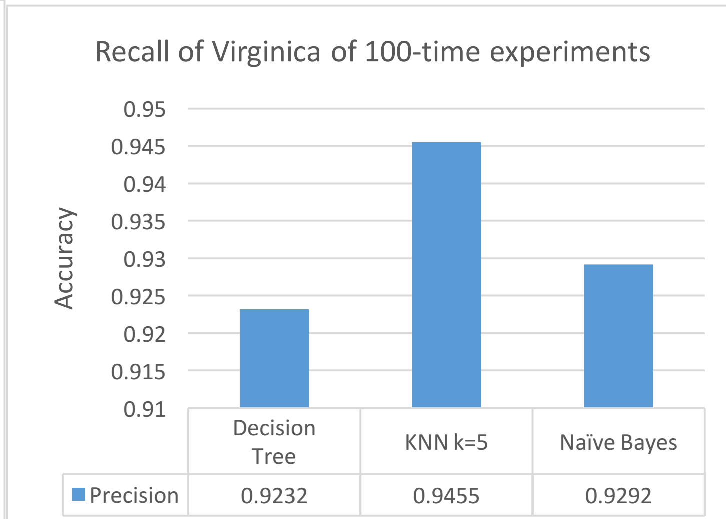
Simply imply sklearn Decision Tree, K-NN k=5, Gaussian Naïve Bayes …

```
DiawChen  /Users/DiawChen/NCTU/Senior/Intro to Machine Learning/HW3  python hw3-1.py
After 100-time testing result:

Decision Tree Accuracy = 0.941122
setosa  precision: 1.0 recall 0.906
versicolor      precision: 1.0 recall 0.9155
virginica       precision: 1.0 recall 0.9069

K-Nearest Neighbor Accuracy = 0.962457
setosa  precision: 1.0 recall 0.9494
versicolor      precision: 1.0 recall 0.9351
virginica       precision: 1.0 recall 0.9402

Naive Bayes Accuracy = 0.955121
setosa  precision: 1.0 recall 0.9222
versicolor      precision: 1.0 recall 0.9423
virginica       precision: 1.0 recall 0.93
```









According to the performance of accuracy, precision and recall (note that precision are both 100%), we can conclude that KNN > Naïve Bayes > Decision Tree in this case.

Dataset 2 : forestfires.csv

1. First convert the categorical type back to intuitive value:
   - month - month of the year: jan to dec -> 1 to 12
   - day - day of the week: mon to sun -> 1 to 7

	X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
0	7	5	3	5	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	0.0
1	7	4	10	2	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.0
2	7	4	10	6	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.0
3	8	6	3	5	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.0
4	8	6	3	7	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.0

2. Explosion data analysis
   - look into statistic info of dataframe

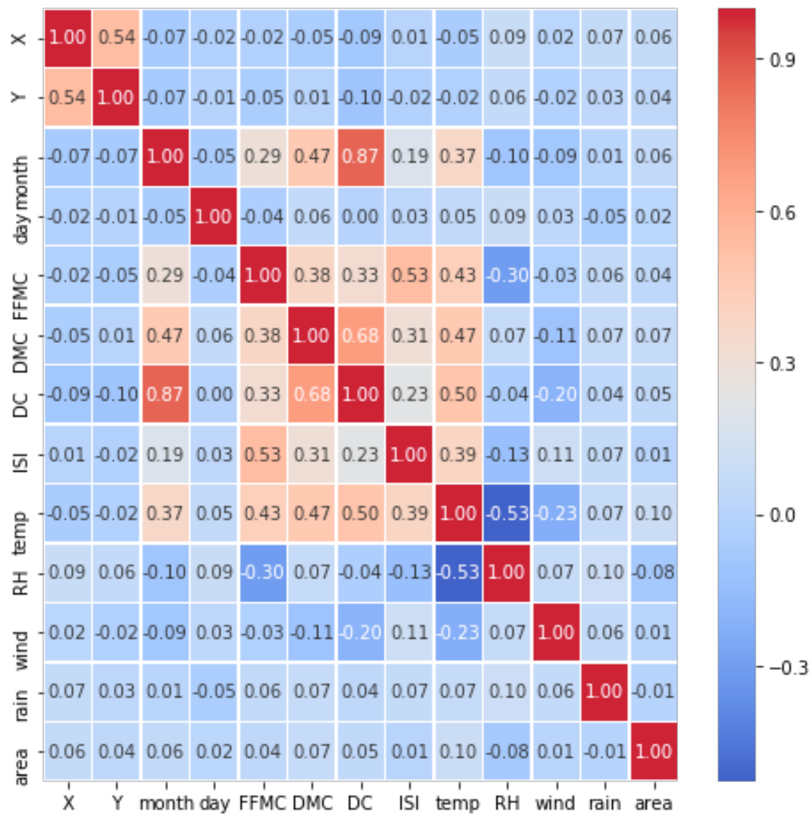
	X	Y	month	day	FFMC	DMC	DC	ISI	temp
count	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000
mean	4.669246	4.299807	7.475822	4.259188	90.644681	110.872340	547.940039	9.021663	18.889168
std	2.313778	1.229900	2.275990	2.072929	5.520111	64.046482	248.066192	4.559477	5.806625
min	1.000000	2.000000	1.000000	1.000000	18.700000	1.100000	7.900000	0.000000	2.200000
25%	3.000000	4.000000	7.000000	2.000000	90.200000	68.600000	437.700000	6.500000	15.500000
50%	4.000000	4.000000	8.000000	5.000000	91.600000	108.300000	664.200000	8.400000	19.300000
75%	7.000000	5.000000	9.000000	6.000000	92.900000	142.400000	713.900000	10.800000	22.800000
max	9.000000	9.000000	12.000000	7.000000	96.200000	291.300000	860.600000	56.100000	33.300000

RH	wind	rain	area
517.000000	517.000000	517.000000	517.000000
44.288201	4.017602	0.021663	12.847292
16.317469	1.791653	0.295959	63.655818
15.000000	0.400000	0.000000	0.000000
33.000000	2.700000	0.000000	0.000000
42.000000	4.000000	0.000000	0.520000
53.000000	4.900000	0.000000	6.570000
100.000000	9.400000	6.400000	1090.840000

   - Show the correlation with heat map

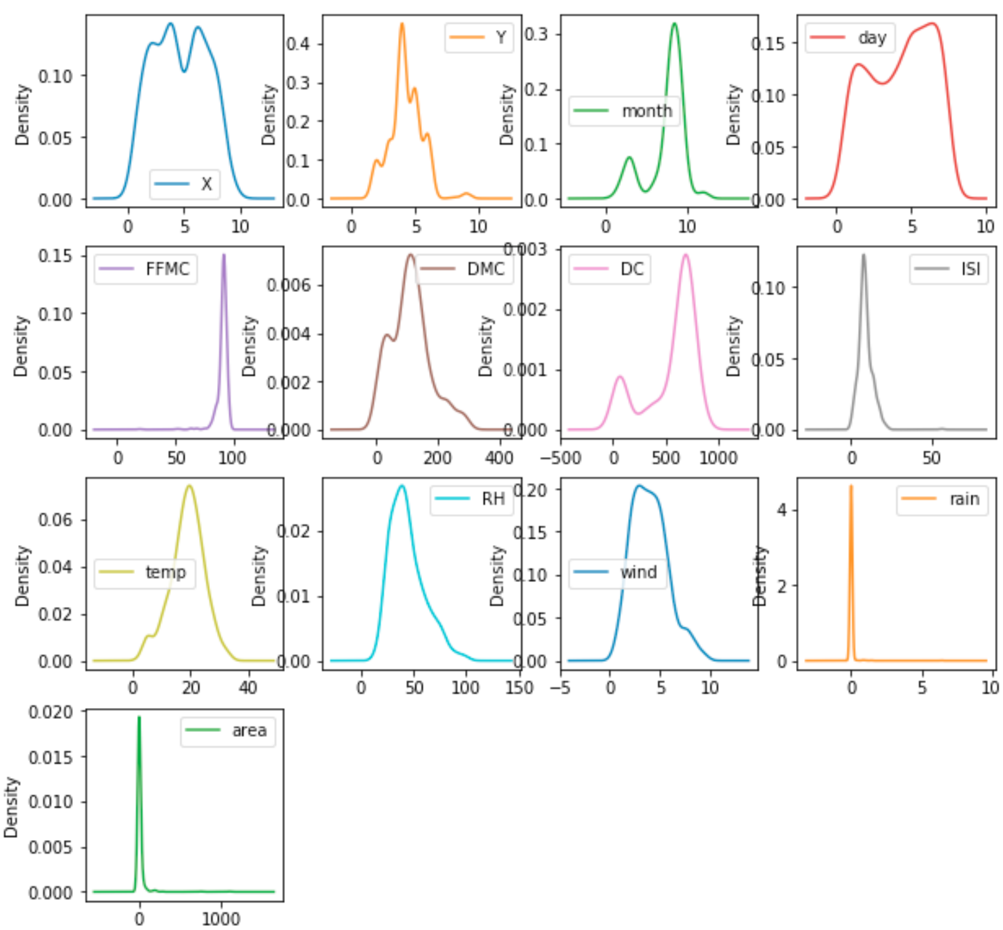
Notice that 'temp','DMC','X','month','DC','Y','FFMC','day','ISI','wind' are positive relative to the target "area"

In contrast, 'RH','rain' are more irrelative to the target "area"

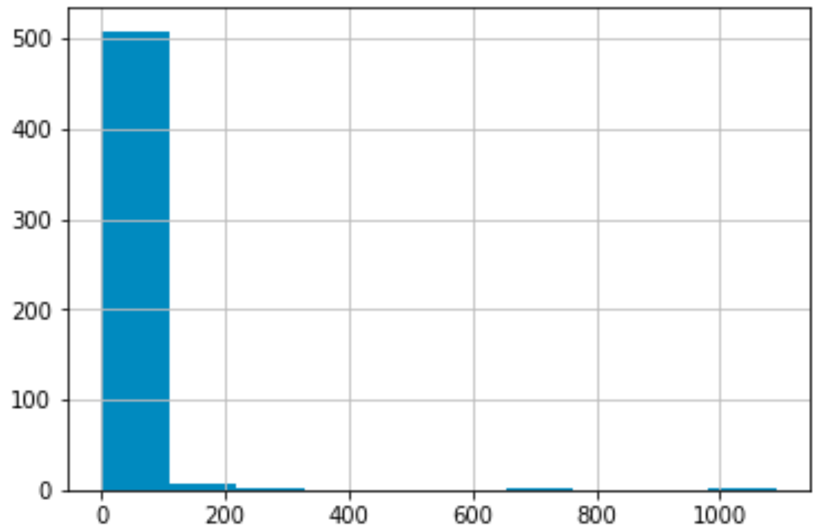


- Visualize distribution with density function

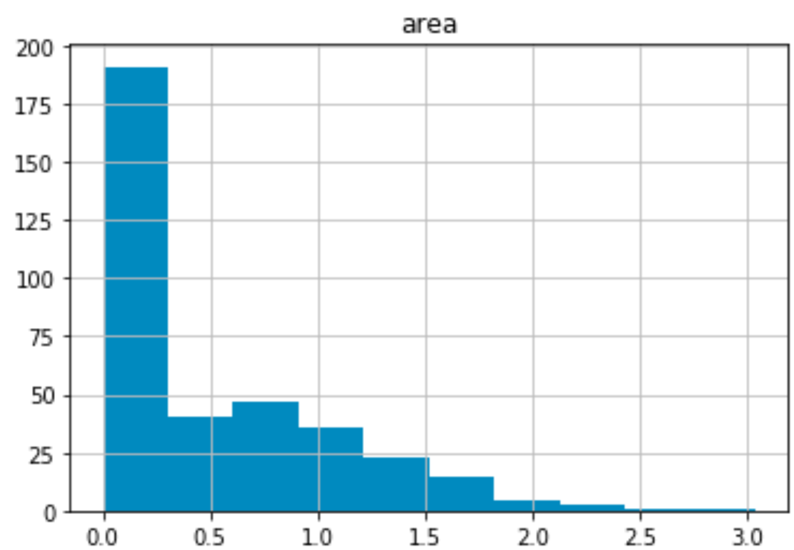
Notice that feature temp acts like normal distribution in the dataset



Take a look at target ‘area’:
Notice that near 95% data gathering around 0



Apply log transformation y = log(1+x) → much more non-skew distribution



Since the original target ‘area’ distribution is too skew, so I decide to transfer area into log scale so that the regressor model can fit more concentrative data. i.e. area target from 0 – 1000 -> 0 – 3。

3. Data normalization:
Use sklearn MinMaxScaler

	X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain
0	0.750	0.428571	0.181818	0.666667	0.870968	0.086492	0.101325	0.090909	0.192926	0.423529	0.700000	0.00000
1	0.750	0.285714	0.818182	0.166667	0.927742	0.118194	0.775419	0.119430	0.508039	0.211765	0.055556	0.00000
2	0.750	0.285714	0.818182	0.833333	0.927742	0.146795	0.796294	0.119430	0.398714	0.211765	0.100000	0.00000
3	0.875	0.571429	0.181818	0.666667	0.941935	0.110958	0.081623	0.160428	0.196141	0.964706	0.400000	0.03125
4	0.875	0.571429	0.181818	1.000000	0.910968	0.172984	0.110590	0.171123	0.295820	0.988235	0.155556	0.00000

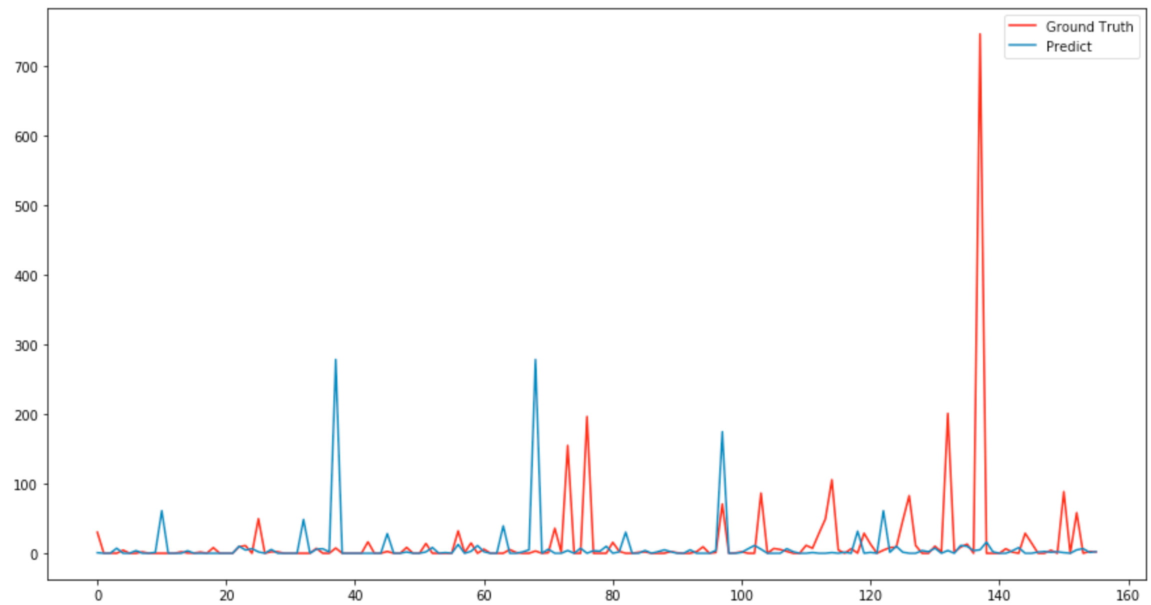
4. Model evulation:
   - Mean Square Error

```
((y_true - y_pred) ** 2).sum()
```

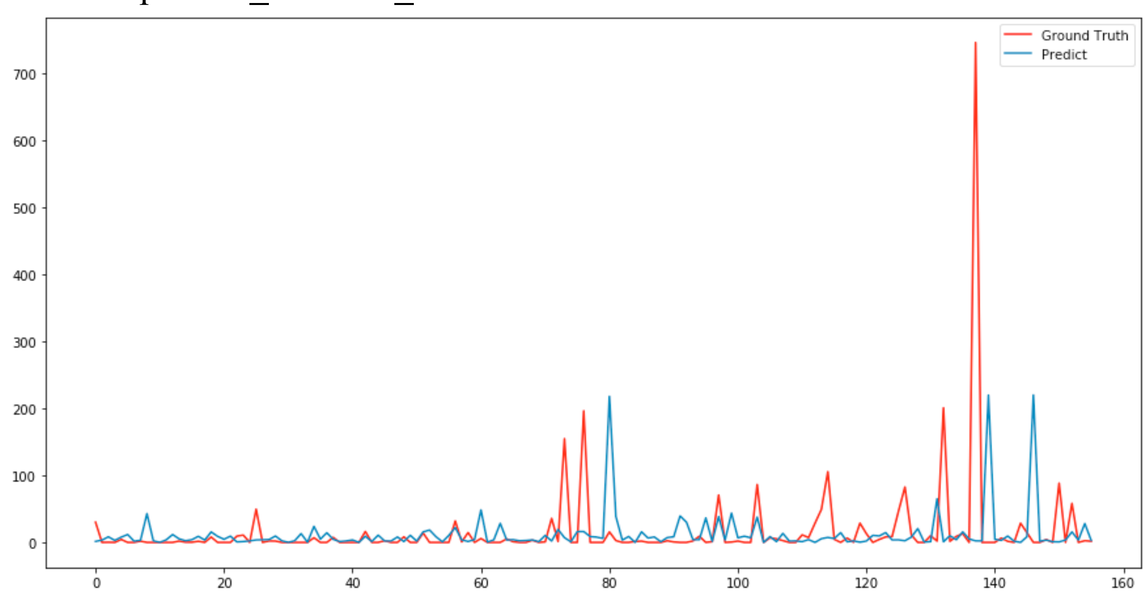
   - Variance score

The variance_score is defined as (1 - u/v), where u is the residual sum of squares ((y_true - y_pred) ** 2).sum() and v is the total sum of squares ((y_true - y_true.mean()) ** 2).sum(). The best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse)

   - <u>Decision Tree Regressor</u>
     - Decision Tree mean squared error = 5596.96581573
     - Explained_variance_score: -0.274530239573

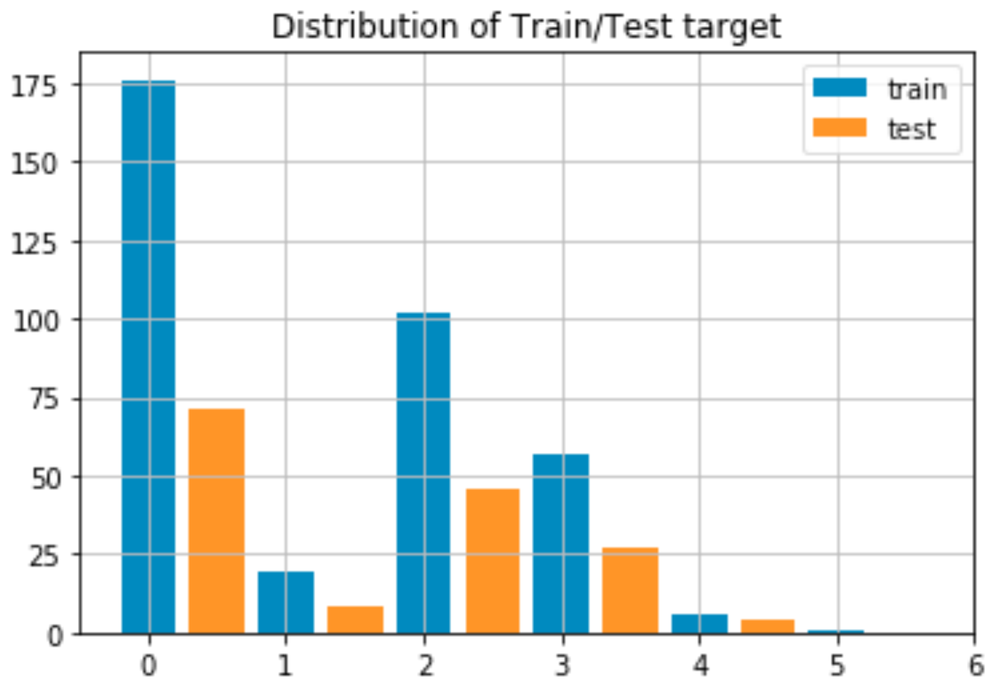


   - <u>KNN Regressor</u>
     - KNeighborsRegressor mean squared error = 5431.07468267
     - Explained_variance_score: -0.245506833378



- <u>Naïve Bayes approach: We define two different model</u>

We define new class for target area: [0, 0-10, 10-100, 100-1000, 1000+] 6 different type



- Categorical features such as ['X','Y','month','day'] apply categorical model - MultinomialNB (with Laplace smooth)

$$P(X_i | Y) = \frac{N(X_i | Y)+k}{N(Y)+km}$$

- Continuous features such as ['FFMC','DMC', 'DC', 'ISI', 'temp','RH', 'wind', 'rain'] apply categorical model - GaussianNB (with Gaussian smooth)

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

- Mix categorical and continuous model
  Concept of Naïve Bayes

$$M(q) = argmaxP(Y)\prod_{i=0}^{m} P(X_i|Y)$$

  Implementation of Mixture Model

$$M(q) = argmax\,(log\ prob\ of\ categorical\ model + log\ prob\ of\ continuous\ model - log\ prob\ of\ prior)$$

- Performance
  Categorical: 45.51%
  Continuous: 22.44%
  Mix model: 23.72%
- Detail prediction of each model

Categorical	Naive_cat_pre array([0, 0])
Continuous	Naive_con_pre array([2, 4, 2, 2, 1, 2, 4, 1, 4, 2, 2, 1, 2, 2, 2, 0, 2, 3, 1, 1, 2, 2, 2, 4, 2, 2, 2, 2, 4, 1, 2, 2, 0, 2, 2, 2, 1, 2, 2, 2, 2, 4, 2, 2, 1, 4, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 4, 1, 4, 2, 2, 2, 2, 2, 0, 1, 4, 2, 2, 2, 2, 2, 2, 2, 4, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 4, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 0, 1, 2, 2, 0, 1, 0, 2, 2, 3, 2, 2, 2, 2, 4, 2, 1, 1, 2, 2, 2, 2, 2, 1, 0, 3, 2, 2, 2, 2, 4, 1, 2, 0, 2, 1, 2, 2, 2, 4, 2, 2, 4, 2, 2, 2, 2, 2, 2, 2, 2, 2])
Mix model	mix_prediction array([2, 4, 2, 2, 1, 2, 4, 1, 4, 2, 2, 1, 2, 2, 2, 0, 2, 3, 1, 1, 2, 2, 2, 4, 2, 2, 2, 2, 4, 1, 2, 2, 0, 2, 2, 2, 1, 2, 2, 2, 2, 4, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 4, 1, 4, 1, 4, 2, 2, 2, 0, 2, 4, 2, 2, 2, 2, 2, 2, 2, 4, 2, 2, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 4, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 0, 1, 2, 2, 0, 2, 0, 2, 4, 3, 2, 2, 4, 2, 4, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 3, 2, 1, 2, 2, 4, 1, 2, 0, 2, 1, 2, 2, 2, 4, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])

It is clear that the result of Categorical model almost dominated by the most frequent tar -get 0, since probability base approach will lead to this kind of result especially when the data distribution is quite skew.

In the contrary, predictions of Gaussian model are much more distributed and strongly influence the prediction of mix model.