MULYA FAJAR NINGSIH ALWI

001202000101

IT 2020 - CLASS 4

LAB ASSIGNMENT 14B

(CODING AND BIG DATA COURSE)

PRACTICE

LabBPracticeWk14.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

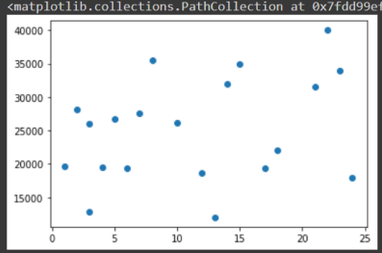
RAM Disk Editing

```
[276] import pandas as pd

      df = pd.read_csv('./gold_karat.csv')
      df.head()
```

	karat	Age(yrs)	Sell Price($)
0	24	6	18000
1	23	3	34000
2	10	5	26100
3	22	2	40000
4	21	4	31500

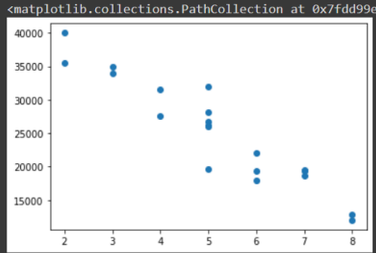
```
[277] import matplotlib.pyplot as plt
      %matplotlib inline

      plt.scatter(df['karat'], df['Sell Price($)'])
```

```
<matplotlib.collections.PathCollection at 0x7fdd99ef4250>
```




```
[278] import matplotlib.pyplot as plt
      %matplotlib inline

      plt.scatter(df['Age(yrs)'], df['Sell Price($)'])
```

```
<matplotlib.collections.PathCollection at 0x7fdd99e67750>
```




```
[279] x = df[['karat', 'Age(yrs)']]
      y = df['Sell Price($)']
```

```
[280] from sklearn.model_selection import train_test_split

      x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3)
```

```
[281] x_train
```

	karat	Age(yrs)
8	13	8
7	17	6
17	1	5
13	7	4
9	18	6
2	10	5
6	14	5
4	21	4
15	15	3
19	2	5
5	5	5
16	8	2
10	12	7
0	24	6

```
[282] x_test
```

	karat	Age(yrs)
14	6	7
3	22	2
12	3	5
18	3	8
1	23	3
11	4	7

```
[283] y_train
```

```
8     12000
7     19300
17    19700
13    27500
9     22000
2     26100
6     32000
4     31500
15    35000
19    28200
5     26750
16    35500
10    18700
0     18000
Name: Sell Price($), dtype: int64
```

```
[284] y_test
```

```
14    19400
3     40000
12    26000
18    12800
1     34000
11    19500
Name: Sell Price($), dtype: int64
```

```
[285] from sklearn.linear_model import LinearRegression

      clf = LinearRegression()
      clf.fit(x_train, y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
[286] x_test
```

	karat	Age(yrs)
14	6	7
3	22	2
12	3	5
18	3	8
1	23	3
11	4	7

```
[287] clf.predict(x_test)
```

```
array([16756.28878824, 38580.41624875, 24965.25905843, 12402.5416092 ,
       34448.23533099, 16645.5056576 ])
```

```
[288] y_test
```

```
14    19400
3     40000
12    26000
18    12800
1     34000
11    19500
Name: Sell Price($), dtype: int64
```

```
[289] clf.score(x_test, y_test)
```

```
0.9640555510827157
```

```
[290] x_train, x_test, y_train, y_test = train_test_split(
          x, y, test_size = 0.3, random_state = 10)

      x_test
```

	karat	Age(yrs)
7	17	6
10	12	7
5	5	5
6	14	5
3	22	2
18	3	8

CHALLENGE

LabBChallengeWk14.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
[21] import pandas as pd

     df = pd.read_csv('./titanic.csv')
     df = df.dropna(subset = ['Age'])

     df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

```
[22] inputs = df.drop('Survived', axis = 'columns')
     target = df['Survived']
```

```
[23] from sklearn.preprocessing import LabelEncoder

     le_Sex = LabelEncoder()
     inputs['Sex_n'] = le_Sex.fit_transform(inputs['Sex'])
```

```
[24] inputs
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Sex_n
0	1	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S	1
1	2	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C	0
2	3	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	0
3	4	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S	0
4	5	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S	1
...	...	...	...	...	...	...	...	...	...	...	...	...
885	886	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652	29.1250	NaN	Q	0
886	887	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S	1
887	888	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S	0
889	890	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C	1
890	891	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q	1

714 rows × 12 columns

```
[25] inputs_n = inputs.drop(['PassengerId', 'Name', 'Sex',
                            'SibSp', 'Parch', 'Ticket', 'Cabin',
                            'Embarked'], axis = 'columns')

     inputs_n
```

	Pclass	Age	Fare	Sex_n
0	3	22.0	7.2500	1
1	1	38.0	71.2833	0
2	3	26.0	7.9250	0
3	1	35.0	53.1000	0
4	3	35.0	8.0500	1
...	...	...	...	...
885	3	39.0	29.1250	0
886	2	27.0	13.0000	1
887	1	19.0	30.0000	0
889	1	26.0	30.0000	1
890	3	32.0	7.7500	1

714 rows × 4 columns

```
[26] target
```

```
0      0
1      1
2      1
3      1
4      0
      ..
885    0
886    0
887    1
889    1
890    0
Name: Survived, Length: 714, dtype: int64
```

```
[27] from sklearn import tree

     clf = tree.DecisionTreeClassifier()
```

```
[28] clf.fit(inputs_n, target)
```

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=None, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')
```

```
[29] clf.score(inputs_n, target)
```

```
0.9845938375350141
```

```
[30] print('are the passengers in the first-class and have a female gender can survive?')
     clf.predict([[1, 30, 70, 0]])
```

```
are the passengers in the first-class and have a female gender can survive?
array([1])
```

```
[31] print('is the passengers in the first-class and have a male gender can survive?')
     clf.predict([[1, 30, 70, 1]])
```

```
is the passengers in the first-class and have a male gender can survive?
array([0])
```

```
[32] print('are the passengers in the second-class and have a female gender can survive?')
     clf.predict([[2, 30, 50, 0]])
```

```
are the passengers in the second-class and have a female gender can survive?
array([1])
```

```
[33] print('are the passengers in the second-class and have a male gender can survive?')
     clf.predict([[2, 30, 50, 1]])
```

```
are the passengers in the second-class and have a male gender can survive?
array([0])
```

```
[34] print('are the passengers in the third-class and have a female gender can survive?')
     clf.predict([[3, 30, 30, 0]])
```

```
are the passengers in the third-class and have a female gender can survive?
array([0])
```

```
[35] print('are the passengers in the third-class and have a male gender can survive?')
     clf.predict([[3, 30, 30, 1]])
```

```
are the passengers in the third-class and have a male gender can survive?
array([0])
```

```
[36] print('are the passengers in the first-class and have a female gender with the oldest age can survive?')
     clf.predict([[1, 70, 100, 0]])
```

```
are the passengers in the first-class and have a female gender with the oldest age can survive?
array([1])
```

```
[37] print('are the passengers in the first-class and have a male gender with the oldest age can survive?')
     clf.predict([[1, 70, 100, 1]])
```

```
are the passengers in the first-class and have a male gender with the oldest age can survive?
array([0])
```

```
[38] print('are the passengers in the first-class with the highest fare can survive?')
     clf.predict([[1, 30, 500, 1]])
```

```
are the passengers in the first-class with the highest fare can survive?
array([1])
```

```
[39] print('are the passengers in the third-class with the lowest fare can survive?')
     clf.predict([[3, 30, 10, 1]])
```

```
are the passengers in the third-class with the lowest fare can survive?
array([1])
```

```
[40] print('are the passengers in the first-class with the youngest age can survive?')
     clf.predict([[1, 3, 100, 1]])
```

```
are the passengers in the first-class with the youngest age can survive?
array([1])
```