

```
In [26]: ### Create a Machine Learning Project

# 1. Import Data: csv files
# 2. Clean the Data: clean the duplicated data, incompleted data
# 3. Split the Data Training/Test: 80% for Training, 20% for test
# 4. Create a Model: Algorithm for Training
# 5. Train the Model
# 6. Make Prediction: Not accurate
# 7. Evaluate and Imporve

# Press Shift+Tab: Show function info
# Press Esc+d+d: Delete column
```

```
In [27]: # Read CSV File

import pandas as pd

data_frame = pd.read_csv('computer_sale.csv')

# Show the matrix size (row, column)
data_frame.shape

# Mean: average
# std: standard measure value
# min: minimum value

# Show complete dataframe specification
data_frame.describe()
```

```
Out[27]:
```

	price	speed	hd	ram	screen
count	6259.000000	6259.000000	6259.000000	6259.000000	6259.000000
mean	2219.576610	52.011024	416.601694	8.286947	14.608723
std	580.803956	21.157735	258.548445	5.631099	0.905115
min	949.000000	25.000000	80.000000	2.000000	14.000000
25%	1794.000000	33.000000	214.000000	4.000000	14.000000
50%	2144.000000	50.000000	340.000000	8.000000	14.000000
75%	2595.000000	66.000000	528.000000	8.000000	15.000000
max	5399.000000	100.000000	2100.000000	32.000000	17.000000

```
In [28]: # Load CSV File

import pandas as pd

data_frame = pd.read_csv('computer_sale.csv')

data_frame
```

```
Out[28]:
```

	price	speed	hd	ram	screen
0	1499	25	80	4	14
1	1795	33	85	2	14
2	1595	25	170	4	15

	price	speed	hd	ram	screen
3	1849	25	170	8	14
4	3295	33	340	16	14
...
6254	1690	100	528	8	15
6255	2223	66	850	16	15
6256	2654	100	1200	24	15
6257	2195	100	850	16	15
6258	2490	100	850	16	17

```
In [40]: # Splitting Data

import pandas as pd

data_frame = pd.read_csv('computer_sale.csv')

# Create new Dataset without columns, price
# X: Declared as Input
X = data_frame.drop(columns=['price'])

# y: declared as Output
y = data_frame['price']

y
```

```
Out[40]: 0      1499
1      1795
2      1595
3      1849
4      3295
...
6254    1690
6255    2223
6256    2654
6257    2195
6258    2490
Name: price, Length: 6259, dtype: int64
```

```
In [41]: # Build the Model

# This method use: Decision Tree

import pandas as pd
from sklearn.tree import DecisionTreeClassifier

data_frame = pd.read_csv('computer_sale.csv')

# Create new Dataset with extracted (price) as output
# X: Declared as Input (any other data other than price)
# X: Is required data in order to get Output
X = data_frame.drop(columns=['price'])

# y: declared as Output (price)
y = data_frame['price']

# Create Model
model = DecisionTreeClassifier()

# Train Model
model.fit(X, y) # (Input, Output)

speed = 25
hd = 250
ram = 16
screen = 14

# Give these Input, to get Output (price)
model1 = [speed, hd, ram, screen]
mode2 = [50, 50, 50, 50]

# Predict Data Input
predictions = model.predict([model1, mode2]) # predict price 100$, 35$ comp

predictions
```

```
Out[41]: array([2595, 2799])
```

```

In [31]: # Measure Model Accuracy

# This method use: Decision Tree

import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

data_frame = pd.read_csv('computer_sale.csv')

# Create new Dataset without columns, price
# X: Declared as Input
X = data_frame.drop(columns=['price'])

# y: declared as Output
y = data_frame['price']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2) # 20% for testing

# Create Model
model = DecisionTreeClassifier()

# Train Model
model.fit(X_train, y_train) # (Input, Ouput)

# Predict Data Input
predictions = model.predict(X_test) # predict price 100$, 35$ computer

# range from 0-1: 0-100%
score = accuracy_score(y_test, predictions)

score

```

Out[31]: 0.12300319488817892

```

In [32]: # Save Trained Model to File

import pandas as pd
from sklearn.tree import DecisionTreeClassifier
import joblib

data_frame = pd.read_csv('computer_sale.csv')

X = data_frame.drop(columns=['price'])
y = data_frame['price']

model = DecisionTreeClassifier()
model.fit(X, y)

joblib.dump(model, 'computer_sale_trained_model.joblib')

```

Out[32]: ['computer_sale_trained_model.joblib']

```

In [42]: # Load Model

# Save Trained Model to File

import pandas as pd
from sklearn.tree import DecisionTreeClassifier
import joblib

model = joblib.load('computer_sale_trained_model.joblib')

speed = 25
hd = 250
ram = 16
screen = 14
model = [speed, hd, ram, screen]
mode2 = [50, 50, 50, 50]

# Predict Data Input
predictions = model.predict([model, mode2]) # predict price 100$, 35$ comp

predictions

```

Out[42]: array([2595, 2799])

```

In [49]: # Visualize Decision Tree from Trained Model

import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
import joblib

data_frame = pd.read_csv('computer_sale.csv')

X = data_frame.drop(columns=['price'])
y = data_frame['speed']

model = DecisionTreeClassifier()
model.fit(X, y)

tree.export_graphviz(model, out_file='graph.dot',
                      feature_names=['speed', 'hd', 'ram', 'screen'], # Input
                      class_names=str(sorted(y.unique())),
                      label='all',
                      rounded=True,
                      filled=True)

```

In []: