untitled6-1

June 13, 2023

0.1 1-MACHINE LEARNING

0.2 1.1 SIMPLE LINEAR REGRESSION

STEP-1 IMPORT DATASET

[ ]:
```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
```

STEP-2 SPLITTING DATASET AND TRAINING AND TESTING DATA

[ ]:
```
df=pd.read_csv("salary_data.csv")
df.head()
```

[ ]:

	YearsExperience	Salary
0	1.1	39343
1	1.3	46205
2	1.5	37731
3	2.0	43525
4	2.2	39891

STEP-3 SELECTING INPUT AND OUTPUT VARIABLES

[ ]:
```
X=df[["YearsExperience"]]
y=df["Salary"]
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.
↪2,random_state=0)
```

STEP-4 MAKING LINEAR REGRESSION MODEL

[ ]:
```
from sklearn.linear_model import LinearRegression
model= LinearRegression()
```

STEP-5 FITTING THE MODEL

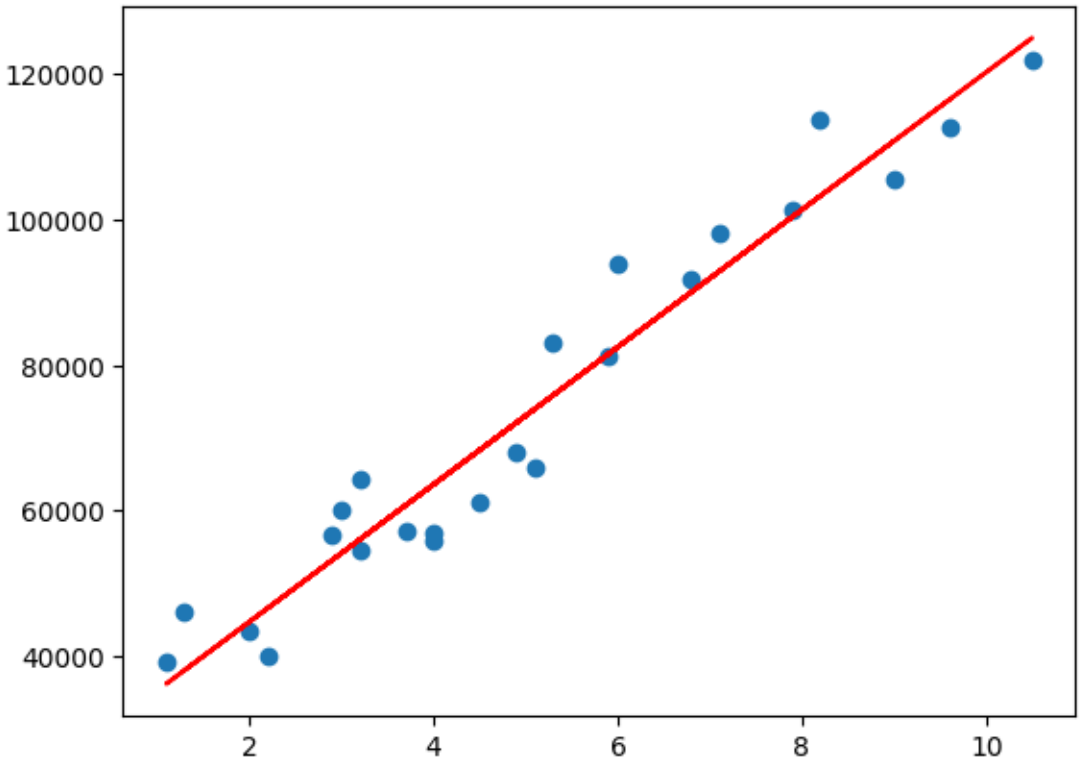
[ ]:
```
model = model.fit(X,y)
model
```

[ ]:
```
LinearRegression()
```

GRAPH

[ ]:
```
import matplotlib.pyplot as plt
plt.scatter(X_train,y_train)
plt.plot(X_train.values, model.predict(X_train), color="red")
```

[ ]:
```
[<matplotlib.lines.Line2D at 0x7f4d91734cd0>]
```



STEP-7 PRICDICTION THE MODEL

[ ]:
```
model.predict([[10]])
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does
not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

[ ]:
```
array([120291.82341322])
```

STEP-8 EVALUATING THE MODEL

[ ]:
```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=0)
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = r2_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.988169515729126
```

0.3 MULTIPLE LINEAR REGRESSION

STEP-1 IMPORT DATASET

[42]:
```
import pandas as pd
df = pd.read_csv("ml_data_salary.csv")
df.head()
```

[42]:

	age	distance	YearsExperience	Salary
0	31.1	77.75	1.1	39343
1	31.3	78.25	1.3	46205
2	31.5	78.75	1.5	37731
3	32.0	80.00	2.0	43525
4	32.2	80.50	2.2	39891

STEP-2 DEFINE DEPENDENT AND INDEPENDENT VARIABLES

[43]:
```
X = df[["age","distance", "YearsExperience"]]
y = df["Salary"]
```

STEP-3 FIT LINEAR REGRESSION MODEL

[44]:
```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model = model.fit(X, y)
model
```

[44]:
```
LinearRegression()
```

STEP-4 EVALUATING MODEL FITNESS

[45]:
```
print("Score for data =" ,model.score(X, y))
```

```
Score for data = 0.9569960750337954
```

STEP-5 PREDICTION OF UNKNOWN VALUES

[46]:
```
model.predict([[31.1,77.75,1.1]])
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does
not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

[46]:
```
array([36209.375])
```

0.4 DECISION FREE CLASSIFIER

STEP-1 IMPORT DATA

[48]:
```
import pandas as pd
df = pd.read_csv("mldata1.csv")
df.head()
```

[48]:

	age	height	weight	gender	likeness
0	27	170.688	76.0	Male	Biryani
1	41	165	70.0	Male	Biryani
2	29	171	80.0	Male	Biryani
3	27	173	102.0	Male	Biryani
4	29	164	67.0	Male	Biryani

STEP-2 MAKING INPUT AND OUTPUT VARIABLES

[49]:
```
df["gender"] = df["gender"].replace("Male",1)
df["gender"] = df["gender"].replace("Female",0)
X = df[["weight","gender"]]
y = df["likeness"]
```

STEP-3 MAKING MACHINE LEARNING MODEL

[50]:
```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier().fit(X,y)
model.predict([[43,0]])
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does
not have valid feature names, but DecisionTreeClassifier was fitted with feature
names
  warnings.warn(
```

[50]:
```
array(['Samosa'], dtype=object)
```

STEP-4 CHECKING MACHINE LEARNING MODEL AND PERFORMANCE

[51]:
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2)
model = DecisionTreeClassifier().fit(X_train,y_train)
predicted_values = model.predict(X_test)
predicted_values
```

[51]:
```
array(['Biryani', 'Biryani', 'Biryani', 'Pakora', 'Biryani', 'Biryani',
       'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani',
       'Biryani', 'Samosa', 'Biryani', 'Pakora', 'Samosa', 'Biryani',
       'Biryani', 'Biryani', 'Pakora', 'Biryani', 'Biryani', 'Biryani',
       'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani',
       'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani',
       'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani', 'Biryani',
       'Samosa', 'Biryani', 'Biryani', 'Pakora', 'Biryani', 'Biryani',
       'Biryani'], dtype=object)
```

STEP-5 MAKING VISUALIZATION

[52]:
```
from sklearn import tree
model = DecisionTreeClassifier().fit(X,y)
tree.export_graphviz(model,out_file= "foodie.dot",
feature_names=["age","gender"],
class_names=sorted(y.unique()),
label="all",rounded=True,filled=True)
```