Machine Learning

Multiple Linear Regression

Step-1 Import Libraries

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Step-2 Import dataset

```
import pandas as pd
df = pd.read_csv("mldata1.csv")
df.head()
```

	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-3 Making input and Output Variable

```
df["gender"] = df["gender"].replace("Male",1)
df["gender"] = df["gender"].replace("Female",0)
```

```
X = df[["weight","gender"]]
y = df["likeness"]
```

Step-4 Making Machine Learning Model

```
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 DecisionTreeClassifie
  warnings.warn(
array(['Samosa'], dtype=object)
```

Step-5 Checking machine learning model performance

```
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
```

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

checking the score

```
score = accuracy_score(y_test, predicted_values)
score
```

```
0.6530612244897959
```

Step-6 Making Visualization

```
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)
```

STEP-1: Import libraries

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

STEP-2: Import dataset

```
from sklearn.datasets import load_digits
digits=load_digits()
```

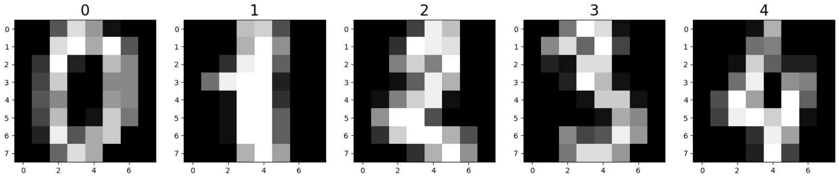
```
X= digits.data
X.shape
```

```
     (1797, 64)
```

```
y=digits.target
y.shape
```

```
     (1797,)
```

```
plt.figure(figsize=(20,4))
for index, (image,label) in enumerate(zip(digits.data[0:5],digits.target[0:5])):
  plt.subplot(1,5,index+1)
  plt.imshow(np.reshape(image,(8,8)),cmap=plt.cm.gray)
  plt.title(label,fontsize=20)
```



```
 # split the data
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)
```

```
# model train
from sklearn.linear_model import LogisticRegression
model = LogisticRegression().fit(X_train,y_train)
model
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: Con
```

```
# prediction
predictions = model.predict(X_test)
predictions
```

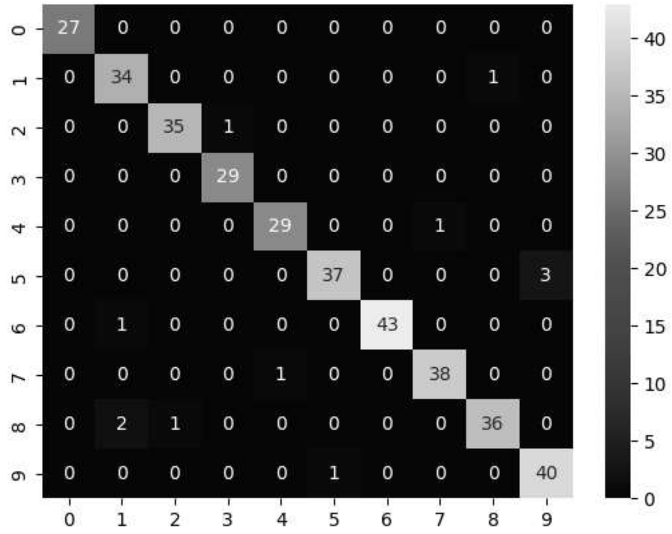
```
array([2, 8, 2, 6, 6, 7, 1, 9, 8, 5, 2, 8, 6, 6, 6, 6, 1, 0, 5, 8, 8, 7,
       8, 4, 7, 5, 4, 9, 2, 9, 4, 7, 6, 8, 9, 4, 3, 1, 0, 1, 8, 6, 7, 7,
       1, 0, 7, 6, 2, 1, 9, 6, 7, 9, 0, 0, 9, 1, 6, 3, 0, 2, 3, 4, 1, 9,
       2, 6, 9, 1, 8, 3, 5, 1, 2, 8, 2, 2, 9, 7, 2, 3, 6, 0, 9, 3, 7, 5,
       1, 2, 9, 9, 3, 1, 4, 7, 4, 8, 5, 8, 5, 5, 2, 5, 9, 0, 7, 1, 4, 7,
       3, 4, 8, 9, 7, 9, 8, 2, 1, 5, 2, 5, 8, 4, 1, 7, 0, 6, 1, 5, 5, 9,
       9, 5, 9, 9, 5, 7, 5, 6, 2, 8, 6, 9, 6, 1, 5, 1, 5, 9, 9, 1, 5, 3,
       6, 1, 8, 9, 8, 7, 6, 7, 6, 5, 6, 0, 8, 8, 9, 8, 6, 1, 0, 4, 1, 6,
       3, 8, 6, 7, 4, 9, 6, 3, 0, 3, 3, 3, 0, 7, 7, 5, 7, 8, 0, 7, 1, 9,
       6, 4, 5, 0, 1, 4, 6, 4, 3, 3, 0, 9, 5, 9, 2, 1, 4, 2, 1, 6, 8, 9,
       2, 4, 9, 3, 7, 6, 2, 3, 3, 1, 6, 9, 3, 6, 3, 3, 2, 0, 7, 6, 1, 1,
       9, 7, 2, 7, 8, 5, 5, 7, 5, 2, 3, 7, 2, 7, 5, 5, 7, 0, 9, 1, 6, 5,
       9, 7, 4, 3, 8, 0, 3, 6, 4, 6, 3, 2, 6, 8, 8, 8, 4, 6, 7, 5, 2, 4,
       5, 3, 2, 4, 6, 9, 4, 5, 4, 3, 4, 6, 2, 9, 0, 1, 7, 2, 0, 9, 6, 0,
       4, 2, 0, 7, 9, 8, 5, 7, 8, 2, 8, 4, 3, 7, 2, 6, 9, 1, 5, 1, 0, 8,
       2, 8, 9, 5, 6, 2, 2, 7, 2, 1, 5, 1, 6, 4, 5, 0, 9, 4, 1, 1, 7, 0,
       8, 9, 0, 5, 4, 3, 8, 8])
```

```
 # confusion matrix
from sklearn import metrics
cm = metrics.confusion_matrix(y_test, predictions)
cm
```

```
array([[27,  0,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0, 34,  0,  0,  0,  0,  0,  0,  1,  0],
       [ 0,  0, 35,  1,  0,  0,  0,  0,  0,  0],
       [ 0,  0,  0, 29,  0,  0,  0,  0,  0,  0],
       [ 0,  0,  0,  0, 29,  0,  0,  1,  0,  0],
       [ 0,  0,  0,  0,  0, 37,  0,  0,  0,  3],
       [ 0,  1,  0,  0,  0,  0, 43,  0,  0,  0],
       [ 0,  0,  0,  0,  1,  0,  0, 38,  0,  0],
       [ 0,  2,  1,  0,  0,  0,  0,  0, 36,  0],
       [ 0,  0,  0,  0,  0,  1,  0,  0,  0, 40]])
```

```
import seaborn as sns
sns.heatmap(cm, annot=True)
```

```
<Axes: >
```



✓ 1s completed at 11:48 PM ● ✕

Machine Learning

Multiple Linear Regression

Step-1 Import Libraries

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Step-2 Import dataset

```
import pandas as pd
df = pd.read_csv("ml_data_salary.csv")
df.head()
```

	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-3 Define dependent and independent variables

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

Step-4 3 Fit Linear Regression Model

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model = model.fit(X, y)
model
```

```
▾ LinearRegression
LinearRegression()
```

```
model.coef_
```

```
array([-3.00216193e+15,  1.18788781e+15,  3.24424072e+13])
```

Step-5 Evaluating Model Fitness

```
print(model.score(X, y))
```

```
0.9569960750337954
```

Step-6 Prediction of unknown values

```
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
  warnings.warn(
array([36209.375])
```

Step-7 Prediction accuracy score

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
data = pd.read_csv('ml_data_salary.csv')  # Replace 'your_dataset.csv' with your actual dataset filename
X = data.drop('distance', axis=1)  # Replace 'target_variable' with the column name of your target variable
y = data['distance']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = r2_score(y_test, y_pred)
print("Accuracy score:", accuracy)
```

```
Accuracy score: 1.0
```

✓ 0s completed at 8:36 PM

STEP-1: Import libraries

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

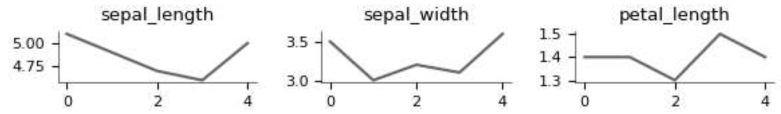
STEP-2: Import dataset

First 5 enteries

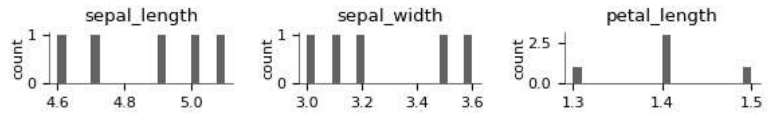
```
df=sns.load_dataset("iris")
df.head()
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

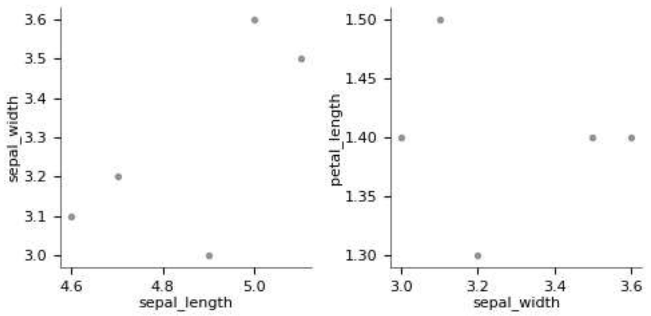
Values



Distributions



2-d distributions



STEP-3: Selecting input and output

```
X=df.iloc[:,:-1]
y=df.iloc[:,-1:]
```

STEP-4: Model creation

```
from sklearn.naive_bayes import GaussianNB
model=GaussianNB().fit(X,y)
```

```
model
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: DataConv
  y = column_or_1d(y, warn=True)
▾ GaussianNB
GaussianNB()
```

STEP-5: Train test split and checking accuracy

```
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-6: Training the model on training data

```
from sklearn.naive_bayes import GaussianNB
model= GaussianNB().fit(X_train,y_train)
model
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: DataConv
  y = column_or_1d(y, warn=True)
▾ GaussianNB
GaussianNB()
```

STEP-7 Making prediction on testing data

```
y_pred=model.predict(X_test)
y_pred
```

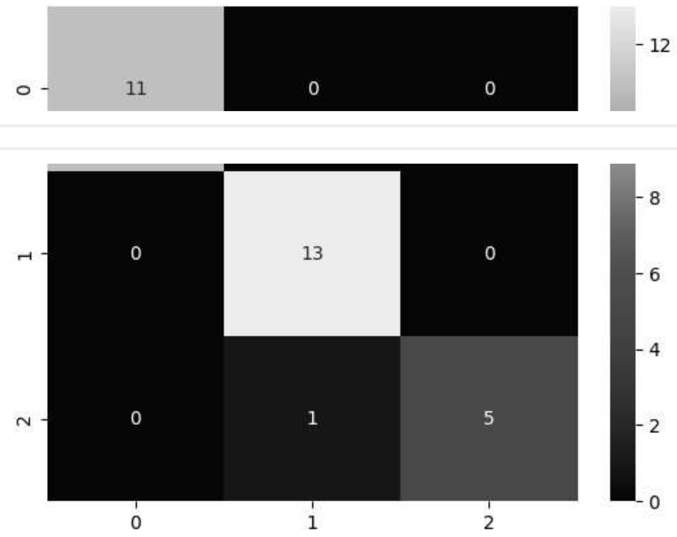
```
array(['virginica', 'versicolor', 'setosa', 'virginica', 'setosa',
       'virginica', 'setosa', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'setosa', 'versicolor', 'versicolor', 'setosa',
       'setosa', 'virginica', 'versicolor', 'setosa', 'setosa',
       'virginica', 'setosa', 'setosa', 'versicolor', 'versicolor',
       'setosa'], dtype='<U10')
```

```
from sklearn.metrics import accuracy_score
score= accuracy_score (y_test,y_pred)
print("Naive bayes model accuracy is", score*100)
```

```
Naive bayes model accuracy is 96.66666666666667
```

```
from sklearn.metrics import confusion_matrix
cm=confusion_matrix (y_test,y_pred)
sns.heatmap(cm,annot=True)
```

```
<Axes: >
```



✓ 0s completed at 10:41 PM

Machine Learning

Simple Linear Regression

Step-1 Import Libraries

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Step-2 Import dataset

```
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 Spliting dataset into training and testing data

```
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 Fit Linear Regression Model

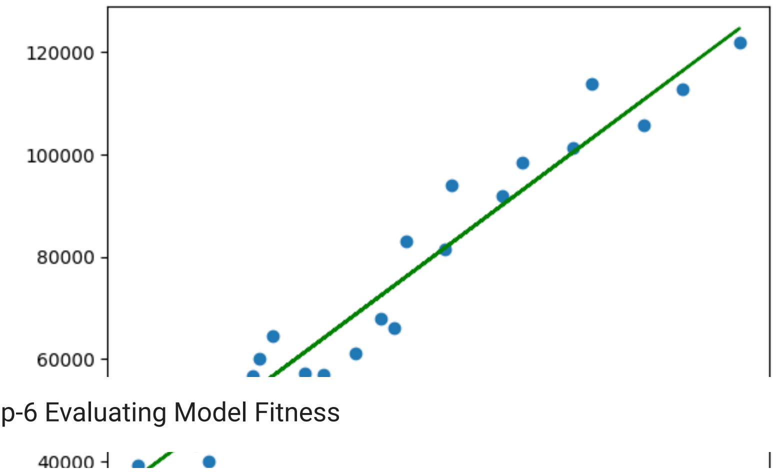
```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model = model.fit(X_train, y_train)
model
```

```
▾ LinearRegression
LinearRegression()
```

Step-5 Plotting

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

```
[<matplotlib.lines.Line2D at 0x7f95f405ebf0>]
```



Step-6 Evaluating Model Fitness

```
print(model.score(X_train, y_train))
print(model.score(X_test, y_test))

     0.9411949620562126
     0.988169515729126
```

Step-7 Prediction of unknown values

```
model.predict([[10],[15],[20]])

     /usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but LinearRegression was
       warnings.warn(
     array([119905.85041792, 166468.72605157, 213031.60168521])
```

step-8 mean squared error

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# Assuming you have your features and target variable in X and y, respectively

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a linear regression model
model = LinearRegression()

# Fit the model on the training data
model.fit(X_train, y_train)

# Predict the target variable for the testing set
y_pred = model.predict(X_test)

# Compute the mean squared error (MSE)
mse = mean_squared_error(y_test, y_pred)

print("Mean Squared Error (MSE):", mse)

     Mean Squared Error (MSE): 49830096.85590839
```

✓ 0s completed at 8:27 PM ● ×