

Metrics syntax:

```
from sklearn.metrics import (metric)

mse_adv = mean_squared_error(y_test_adv, y_pred_adv)

metrics=accuracy_score(Y_test,y_pred)

precision = precision_score(Y_test, Y_pred)

metrics = classification_report(y_test, y_pred)

metrics= confusion_matrix(y_test, y_pred)

metrics= mean_absolute_error(y_test, y_pred)
```

Question 1: Implement a classification algorithm on diabetes.csv and print accuracy

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

data=pd.read_csv('diabetes.csv')

data.drop(columns=['patientID'])

x=data.drop(columns=['class'])
y=data[class]

X_train,X_test,Y_train,Y_test=train_test_split(x,y test_size=0.2,random_state=42)

model=RandomForestClassifier
model.fit=(X_train,Y_train)

y_pred=model.predict(X_test)

metrics=accuracy_score(Y_test,y_pred)
print("Accuracy:",metrics)
```

Question 9: Implement support vector classifier on diabetes.csv dataset and print precision score

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import precision_score

data=pd.read_csv('diabetes.csv')
data.drop(columns=['patientID'])

x=data.drop(columns=['class'])
y=data['class']

X_train,X_test,Y_train,Y_test=train_test_split(x,y,test_size=0.2,random_state=42)

model = SVC()
model.fit(X_train, Y_train)

Y_pred=model.predict(X_test)
precision = precision_score(Y_test, Y_pred)
print("Precision Score:", precision)
```

#Question 12: Implement linear regression on salary.csv dataset and print mean absolute error. Plot a graph year of experience vs salary

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
```

```
# Load the dataset
data = pd.read_csv('salary.csv')
```

```
# Remove the null values
data.dropna(inplace=True)
```

```
# Extract features and target
X = data['YearsExperience'].values.reshape(-1, 1)
y = data['Salary'].values
```

```
# Initialize and train linear regression model
model = LinearRegression()
model.fit(X, y)
```

```
# Predict the salaries
y_pred = model.predict(X)
```

```
# Calculate mean absolute error
mae = mean_absolute_error(y, y_pred)
print("Mean Absolute Error for Linear Regression:", mae)
```

```
# Plot the graph
plt.scatter(X, y, color='blue')
plt.plot(X, y_pred, color='red')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.title('Linear Regression: Years of Experience vs Salary')
plt.show()
```

Question 16: Implement a KNN classifier on HeartDisease1.csv dataset and print accuracy score

```
# Initialize and train KNN classifier for HeartDisease1.csv dataset
knn_model_heart = KNeighborsClassifier()
knn_model_heart.fit(X_train_heart, y_train_heart)

# Predict the target values for the test set
y_pred_knn_heart = knn_model_heart.predict(X_test_heart)

# Calculate and print the accuracy score for HeartDisease1.csv dataset
accuracy_knn_heart = accuracy_score(y_test_heart, y_pred_knn_heart)
print("Accuracy Score for KNN Classifier on HeartDisease1.csv dataset:", accuracy_knn_heart)
```