Q1)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report


df=pd.read_csv("BankNote_Authentication.csv")

df.isnull().sum()

df.dropna()

X = df['rm'].values
y = df['medv'].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

scaler= StandardScaler()
X_train_scaled = scaler.fit_transform(X_train.reshape(-1, 1))
X_test_scaled = scaler.transform(X_test.reshape(-1, 1))
```

```
ridge = Ridge(alpha=1.0)
lasso = Lasso(alpha=0.1)
elasticnet = ElasticNet(alpha=0.1, l1_ratio=0.5)

ridge.fit(X_train_scaled, y_train)
lasso.fit(X_train_scaled, y_train)
elasticnet.fit(X_train_scaled, y_train)

y_pred_ridge = ridge.predict(X_test_scaled)
y_pred_lasso = lasso.predict(X_test_scaled)
y_pred_elasticnet = elasticnet.predict(X_test_scaled)

mse_ridge = mean_squared_error(y_test, y_pred_ridge)
mse_lasso = mean_squared_error(y_test, y_pred_lasso)
mse_elastic = mean_squared_error(y_test, y_pred_elasticnet)

print("Ridge Regression MSE:", mse_ridge)
print("Lasso Regression MSE:", mse_lasso)
print("ElasticNet Regression MSE:", mse_elastic)

print("Prediction for 5 rooms is:",ridge.predict([[5]])[0])
print("Prediction for 5 rooms is:",lasso.predict([[5]])[0])
print("Prediction for 5 rooms is:",elasticnet.predict([[5]])[0])
plt.scatter(X, y, color="black", label="Original Data")
plt.plot(X_test, y_pred_ridge, color='yellow', label="Ridge Regression")

plt.plot(X_test, y_pred_lasso, color='red', label="Lasso Regression")

plt.plot(X_test, y_pred_elasticnet, color='green', label="elaticnet Regression")

plt.xlabel("Number of Rooms (RM)")

plt.ylabel(" Price")

plt.title("Ridge vs Lasso Regression (Boston Housing)")

plt.legend()

plt.show()
```

Q2)

```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix


data=pd.read_csv("BankNote_Authentication.csv")

print(data)


df=df.dropna()


df.isnull().sum()
```

```
X = data.drop('class', axis=1)
y = data['class']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = DecisionTreeClassifier(random_state=50)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

print("\nAccuracy Score:")
print(accuracy_score(y_test, y_pred))

print("\nClassification Report:")
print(classification_report(y_test, y_pred))

print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```