Q1)

```
import numpy as np
from sklearn.datasets import fetch_california_housing
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score

housing = fetch_california_housing()
X = housing.data
y = housing.target

print(X)
print(y)

average_price = np.mean(y)
y_binary = (y > average_price).astype(int)

X_train, X_test, y_train, y_test = train_test_split(X, y_binary, test_size=0.3)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
mlp = MLPClassifier(hidden_layer_sizes=(35,5),activation='tanh', max_iter=500)
mlp.fit(X_train, y_train)

y_pred = mlp.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Test Accuracy:",accuracy)
```

Q2)

```
import pandas as pd
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report,accuracy_score

iris=pd.read_csv('IRIS.csv')

x=iris.drop('species',axis=1)
y=iris['species']

X_train,X_test, y_train, y_test=train_test_split(x,y,test_size=0.3)
```

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

kernels = ['linear', 'poly', 'rbf', 'sigmoid']
for k in kernels:
 model = SVC(kernel=k)
 model.fit(X_train, y_train)
 y_pred = model.predict(X_test)
 print(k, "accuracy =", accuracy_score(y_test, y_pred))

new_flower=[[1.3,7.5,5.5,7.7]]
new_flower=scaler.transform(new_flower)
print("The newly predicted flower is:",model.predict(new_flower)[0])
```