Q1)

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import r2_score

df = pd.read_csv('Position_Salaries .csv')
df

X = df[['Level']].values
y = df['Salary'].values

lin_reg = LinearRegression()
lin_reg.fit(X, y)

poly_reg = PolynomialFeatures(degree=4)
X_poly = poly_reg.fit_transform(X)
lin_reg_poly = LinearRegression()
lin_reg_poly.fit(X_poly, y)

level_11_salary_linear = lin_reg.predict([[11]])[0]
level_12_salary_linear = lin_reg.predict([[12]])[0]
level_11_salary_poly = lin_reg_poly.predict(poly_reg.transform([[11]]))[0]
level_12_salary_poly = lin_reg_poly.predict(poly_reg.transform([[12]]))[0]
```

```
print("\nPredicted Salaries:")
print("level_11_salary_linear:",level_11_salary_linear)
print("level_12_salary_linear:",level_12_salary_linear)
print("level_11_salary_poly:",level_11_salary_poly)
print("level_12_salary_poly:",level_12_salary_poly)

y_pred_linear = lin_reg.predict(X)
y_pred_poly = lin_reg_poly.predict(X_poly)

print("Linear Regression R² Score:", r2_score(y, y_pred_linear))
print("Polynomial Regression R² Score:", r2_score(y, y_pred_poly))

plt.scatter(X, y, color='red', label='Original Data')
plt.plot(X, y_pred_linear, color='blue', label='Linear Regression')
plt.plot(X,y_pred_poly, color='green', label='Polynomial Regression')
plt.title('Position Level vs Salary')
plt.xlabel('Position Level')
plt.ylabel('Salary')
plt.show()
```

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')
iris

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])
```