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

dataset=pd.read_csv(r"C:\Users\admin\Downloads\emp_sal.csv")

x=dataset.iloc[:,1:2].values
y=dataset.iloc[:,2].values

 # Linear Regression
from sklearn.linear_model import LinearRegression
lin_reg=LinearRegression()
lin_reg.fit(x,y)
plt.scatter(x,y,color='red')
plt.plot(x,lin_reg.predict(x),color='blue')
plt.title('Linear regression graph')
plt.xlabel('Position lavel')
plt.ylabel('Salary')
plt.show()

from sklearn.preprocessing import PolynomialFeatures
poly_reg=PolynomialFeatures(degree=3)
x_poly=poly_reg.fit_transform(x)
poly_reg.fit(x_poly,y)
 #Linear Model built with 2nd degree
lin_reg2=LinearRegression()
lin_reg2.fit(x_poly,y)
 #Polymodel
plt.scatter(x,y,color='red')
plt.plot(x,lin_reg2.predict(poly_reg.fit_transform(x)),color='blue')
plt.title('Truth or bluff (Polynomial Regression)')
plt.xlabel('Position lavel')
plt.ylabel('Salary')
plt.show()

from sklearn.preprocessing import PolynomialFeatures
poly_reg=PolynomialFeatures(degree=3)
x_poly=poly_reg.fit_transform(x)
poly_reg.fit(x_poly,y)
lin_reg2=LinearRegression()
lin_reg2.fit(x_poly,y)
plt.scatter(x,y,color='red')
plt.plot(x,lin_reg2.predict(poly_reg.fit_transform(x)),color='blue')
plt.title('Truth or bluff (Polynomial Regression)')
plt.xlabel('Position lavel')
plt.ylabel('Salary')
plt.show()
lin_model_pred=lin_reg.predict([[6.5]])
print(lin_model_pred)
poly_reg_pred=lin_reg2.predict(poly_reg.fit_transform([[6.5]]))
print(poly_reg_pred)

from sklearn.preprocessing import PolynomialFeatures
poly_reg=PolynomialFeatures(degree=4)
x_poly=poly_reg.fit_transform(x)
poly_reg.fit(x_poly,y)
lin_reg2=LinearRegression()
```

```
lin_reg2.fit(x_poly,y)
plt.scatter(x,y,color='red')
plt.plot(x,lin_reg2.predict(poly_reg.fit_transform(x)),color='blue')
plt.title('Truth or bluff (Polynomial Regression)')
plt.xlabel('Position lavel')
plt.ylabel('Salary')
plt.show()
lin_model_pred=lin_reg.predict([[6.5]])
print(lin_model_pred)

from sklearn.preprocessing import PolynomialFeatures
poly_reg=PolynomialFeatures(degree=5)
x_poly=poly_reg.fit_transform(x)
poly_reg.fit(x_poly,y)
lin_reg2=LinearRegression()
lin_reg2.fit(x_poly,y)
plt.scatter(x,y,color='red')
plt.plot(x,lin_reg2.predict(poly_reg.fit_transform(x)),color='blue')
plt.title('Truth or bluff (Polynomial Regression)')
plt.xlabel('Position lavel')
plt.ylabel('Salary')
plt.show()
lin_model_pred=lin_reg.predict([[6.5]])
print(lin_model_pred)

 ## svr model
from sklearn.svm import SVR
svr_model = SVR(kernel='poly',degree=4,gamma='auto',C = 10.0)
svr_model.fit(x,y)

svr_model_pred = svr_model.predict([[6.5]])
print(svr_model_pred)

 ## knn regression model
from sklearn.neighbors import KNeighborsRegressor

# Create the model
knn_model = KNeighborsRegressor(
    n_neighbors=5,
    weights='distance',
    algorithm='brute',
    p=1
)

# Fit the model
knn_model.fit(x, y)   # use X and Y if those are your dataset variables

# Make a prediction
knn_model_pred = knn_model.predict([[6.5]])
print(knn_model_pred)

## Decission tree model
from sklearn.tree import DecisionTreeRegressor
dt_model = DecisionTreeRegressor()
dt_model.fit(x,y)
```

```
dt_model_pred = dt_model.predict([[6.5]])
print(dt_model_pred)

##Random Forest

from sklearn.ensemble import RandomForestRegressor
rf_model = RandomForestRegressor(random_state=0)
rf_model.fit(x,y)

rf_model_pred = rf_model.predict([[6.5]])
print(rf_model_pred)

from sklearn.ensemble import RandomForestRegressor
rf_model = RandomForestRegressor( n_estimators=27,random_state=0)
rf_model.fit(x,y)

rf_model_pred = rf_model.predict([[6.5]])
print(rf_model_pred)

from sklearn.ensemble import RandomForestRegressor
rf_model = RandomForestRegressor( n_estimators=30,random_state=0)
rf_model.fit(x,y)

rf_model_pred = rf_model.predict([[6.5]])
print(rf_model_pred)

from sklearn.ensemble import RandomForestRegressor
rf_model = RandomForestRegressor( n_estimators=23,random_state=0)
rf_model.fit(x,y)

rf_model_pred = rf_model.predict([[6.5]])
print(rf_model_pred)
```