

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

```
dataset=pd.read_csv(r"D:\Data Science with AI\Data Science With AI\28-august-25th, 26th - Svr, L
```

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

```
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 model(Linear Regression)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```

```
lin_model_pred=lin_reg.predict([[6.5]])
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_reg_2=LinearRegression()
lin_reg_2.fit(x_poly,y)
```

```
poly_model_pred=lin_reg_2.predict(poly_reg.fit_transform([[6.5]]))
poly_model_pred
```

```
plt.scatter(x,y,color='red')
plt.plot(x,lin_reg_2.predict(poly_reg.fit_transform(x)),color='blue')
plt.title('polymodel (Polynomial Regression)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```

```
from sklearn.svm import SVR
svr_reg=SVR(kernel='poly',gamma='auto',degree=4)
svr_reg.fit(x,y)
```

```
svr_reg_pred=svr_reg.predict([[6.5]])
svr_reg_pred
```

```
from sklearn.neighbors import KNeighborsRegressor
knn_reg=KNeighborsRegressor(n_neighbors=2)
knn_reg.fit(x,y)
```

```
knn_reg_pred=knn_reg.predict([[6.5]])
knn_reg_pred
```

```
from sklearn.tree import DecisionTreeRegressor
dt_reg=DecisionTreeRegressor()
dt_reg.fit(x,y)
```

```
dt_reg_pred=dt_reg.predict([[6.5]])
dt_reg_pred
```

```
from sklearn.ensemble import RandomForestRegressor
rf_reg=RandomForestRegressor()
rf_reg.fit(x,y)
```

```
rf_reg_pred=rf_reg.predict([[6.5]])
rf_reg_pred
```

```
rf_reg=RandomForestRegressor(n_estimators=27,random_state=0)
rf_reg.fit(x,y)
```

```
rf_reg_pred=rf_reg.predict([[6.5]])
rf_reg_pred
```

```
import xgboost as xg
xgb_r=xg.XGBRegressor()
xgb_r.fit(x,y)
```

```
xgb_reg_pred=xgb_r.predict([[6.5]])
xgb_reg_pred
```

```
xgb_r=xg.XGBRegressor(objective='reg:linear',n_estimators=4)
xgb_r.fit(x,y)
```

```
xgb_reg_pred=xgb_r.predict([[6.5]])
xgb_reg_pred
```

