

# *Project Title:*

**Prediction of Salaries According to Years of  
Experience and Job Position with  
Polynomial Linear Regression**

**Name:K.Pranavi**

**Roll No:20X01A0584**

**Year: 4th Year**

**Project Guide Name:**

**Mr. Lokesh B Sir**

# Problem Statement

This project aims to employ Polynomial Linear Regression to predict salary levels based on job positions and experience. The dataset includes various job titles, experience levels, and corresponding salaries. The primary objective is to build an accurate model that estimates salaries based on these two key factors. By doing so, we can provide valuable insights to both job seekers and employers. Job seekers can better understand the salary expectations associated with different positions and experience levels, enabling them to negotiate competitive compensation packages. Employers can use this model to make data-driven decisions when determining salary structures, contributing to fairer and more transparent hiring practices. Ultimately, this project seeks to bridge the information gap in the job market, promoting equity and informed decision-making.

## ▼ Polynomial Regression

### ▼ Importing the libraries

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

### ▼ Importing the dataset

```
dataset = pd.read_csv('Position_Salaries.csv')
X = dataset.iloc[:,1:-1].values
y = dataset.iloc[:, -1].values
```

### ▼ Training the Linear Regression model on the whole dataset

```
from sklearn.linear_model import LinearRegression
lin_reg = LinearRegression()
lin_reg.fit(X, y)
```

```
▼ LinearRegression
LinearRegression()
```

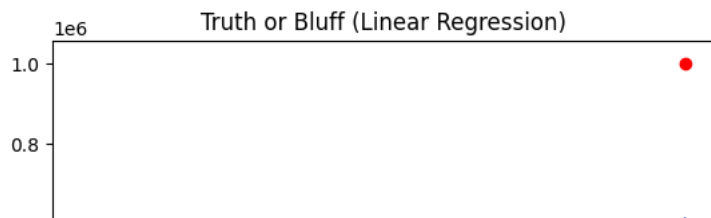
### ▼ Training the Polynomial Regression model on the whole dataset

```
from sklearn.preprocessing import PolynomialFeatures
poly_reg = PolynomialFeatures(degree = 4)
X_poly = poly_reg.fit_transform(X)
lin_reg_2 = LinearRegression()
lin_reg_2.fit(X_poly, y)
```

```
▼ LinearRegression
LinearRegression()
```

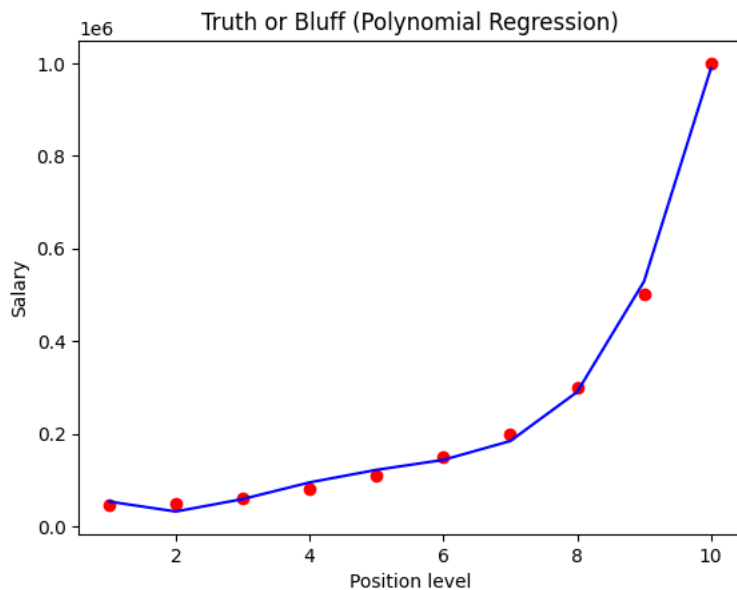
### ▼ Visualising the Linear Regression results

```
plt.scatter(X, y, color = 'red')
plt.plot(X, lin_reg.predict(X), color = 'blue')
plt.title('Truth or Bluff (Linear Regression)')
plt.xlabel('Position Level')
plt.ylabel('Salary')
plt.show()
```



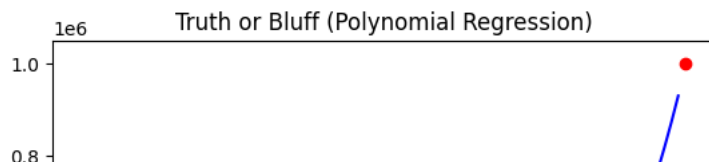
### Visualising the Polynomial Regression results

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



### Visualising the Polynomial Regression results (for higher resolution and smoother curve)

```
X_grid = np.arange(min(X), max(X), 0.1)
X_grid = X_grid.reshape((len(X_grid), 1))
plt.scatter(X, y, color = 'red')
plt.plot(X_grid, lin_reg_2.predict(poly_reg.fit_transform(X_grid)), color = 'blue')
plt.title('Truth or Bluff (Polynomial Regression)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```



### ▼ Predicting a new result with Linear Regression

```
lin_reg.predict([[6.5]])
array([330378.78787879])
```

### ▼ Predicting a new result with Polynomial Regression

```
lin_reg_2.predict(poly_reg.fit_transform([[6.5]]))
array([158862.45265155])
```

Position level

[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 9:18 PM

