

Лабораторная работа №8 по дисциплине "Искусственный интеллект и машинное обучение"

Выполнил: студент 2-го курса Звездин Алексей Сергеевич

Группа: ПИЖ-6-о-22-1

Руководитель практики: Березина Виктория Андреевна, ассистент кафедры информационных систем и технологий института цифрового развития

Тема работы: Полиномиальная регрессия

Цель работы: Научиться применять разработанный пайплайн для тиражирования кода с целью решения задачи полиномиальной регрессии.

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


```
dataset = pd.read_csv('https://raw.githubusercontent.com/InternetHacker1123/bd_ai/main/laba8/data.csv')
dataset.head()
```



	Position	Level	Salary	
0	Data Analyst	1	47000	
1	Project Coordinator	2	52000	
2	Operations Manager	3	63000	
3	Strategy Director	4	85000	
4	Regional Director	5	115000	


Next steps:  [View recommended plots](#)

```
X = dataset.iloc[:, 1:2].values
y = dataset.iloc[:, 2].values
print ("Матрица признаков"); print(X[:5])
print ("Зависимая переменная"); print(y[:5])
```




```
Матрица признаков
[[1]
 [2]
 [3]
 [4]
 [5]]
Зависимая переменная
[ 47000  52000  63000  85000 115000]
```

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




```
▼ LinearRegression
LinearRegression()
```

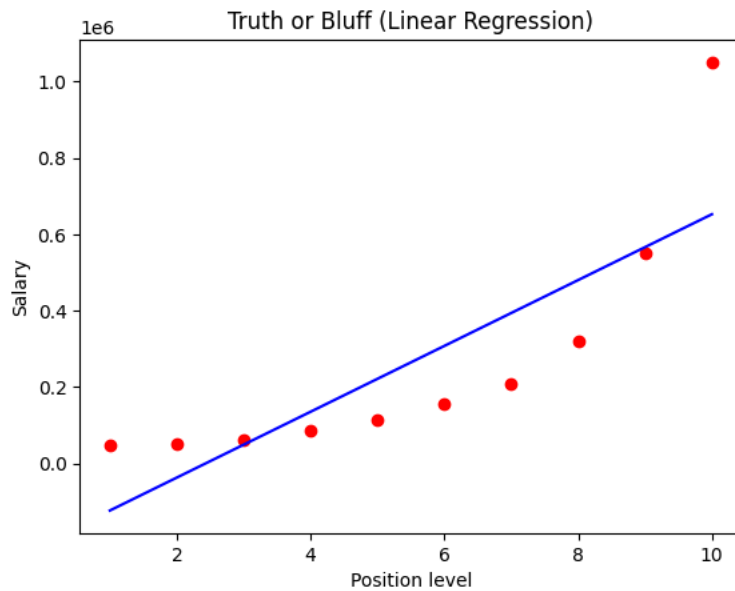
```
from sklearn.preprocessing import PolynomialFeatures
poly_reg = PolynomialFeatures(degree = 10)
X_poly = poly_reg.fit_transform(X)
poly_reg.fit(X_poly, y)
lin_reg_2 = LinearRegression()
lin_reg_2.fit(X_poly, y)
```



```
▼ LinearRegression
LinearRegression()
```

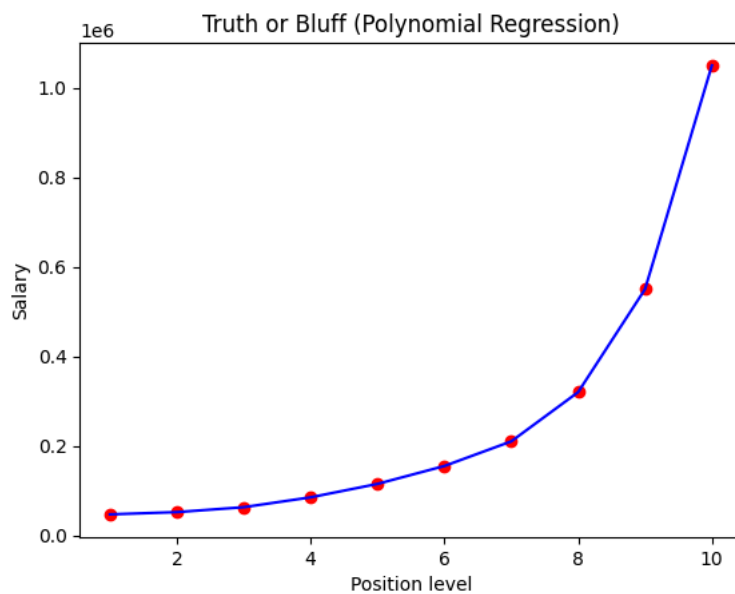
```
y_pred_lin = lin_reg.predict([[6.5]])
y_pred_poly = lin_reg_2.predict(poly_reg.fit_transform([[6.5]]))
print(y_pred_lin, y_pred_poly)
plt.scatter(X, y, color = 'red')
plt.plot(X, lin_reg.predict(X), color = 'blue')
plt.title('Truth or Bluff (Linear Regression)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```

 [350839.39393939] [179259.63240206]



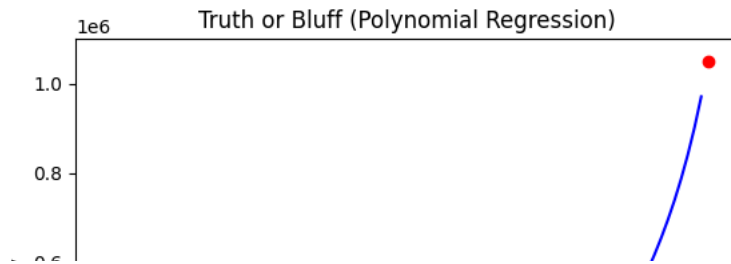
```
plt.scatter(X, y, color = 'red')
plt.plot(X, lin_reg_2.predict(poly_reg.fit_transform(X)), color = 'blue')
plt.title('Truth or Bluff (Polynomial Regression)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```





```
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()
```

```
<ipython-input-9-72d1221b8e2e>:1: DeprecationWarning: Conversion of an array with ndim > 0 to a scalar is deprecated, and will error
X_grid = np.arange(min(X), max(X), 0.1)
```



✓ Контрольные вопросы:

1. Почему при реализации многомерной линейной регрессии необходимо добавить фиктивный признак с единственным значением 1.0?

Добавление фиктивного признака с единственным значением 1.0 в многомерную линейную регрессию является необходимым для корректного обучения модели. Этот признак представляет собой свободный член (intercept) и позволяет модели учитывать смещение (bias) в данных. Без этого фиктивного признака модель может неправильно интерпретировать данные и давать неверные прогнозы.

2. Что такое фиктивная переменная? Поясните причину удаления одной фиктивной переменной, возникающей при перекодировке категориального признака.

Фиктивная переменная (также известная как dummy переменная или индикаторная переменная) - это бинарный признак, который используется для представления категориальных данных в числовой форме в статистических моделях, таких как линейная регрессия. Фиктивные переменные принимают значение 0 или 1 в зависимости от принадлежности к определенной категории.

3. С использованием какого класса создается модель полиномиальной регрессии?

В Python модель полиномиальной регрессии обычно создается с использованием класса PolynomialFeatures из библиотеки sklearn.preprocessing. Этот класс позволяет создавать новые признаки, которые являются полиномиальными комбинациями исходных признаков.

4. Поясните принцип преобразования признаков при построении полиномиальной регрессии.

Преобразование признаков при построении полиномиальной регрессии заключается в создании новых признаков путем комбинирования исходных признаков с использованием полиномиальных функций. Это позволяет модели аппроксимировать нелинейные зависимости между признаками и целевой переменной.

5. Возможно ли применение технологий масштабирования признаков при реализации полиномиальной регрессии?

Да, применение технологий масштабирования признаков также может быть полезным при реализации полиномиальной регрессии. Масштабирование признаков помогает улучшить процесс обучения модели и повысить её эффективность.