Лабораторная работа 8. Построение пайплайна одномерной регрессии

Подключение библиотек

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

Загрузка данных и разделение на матрицу признаков и зависимую переменную

```
dataset = pd.read_csv('data.csv')
dataset.head()
```


```
{"summary":"{\n  \"name\": \"dataset\",\n  \"rows\": 29,\n
\"fields\": [\n    {\n      \"column\": \"Age\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
2.7907394362400124,\n        \"min\": 1.3,\n        \"max\": 10.5,\n
\"num_unique_values\": 27,\n        \"samples\": [\n          3.9,\n
5.1,\n          4.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Height\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 18.5133717883958,\n        \"min\": 79.45,\n        \"max\":
141.26,\n        \"num_unique_values\": 29,\n        \"samples\": [\n
140.33,\n          111.11,\n          103.11\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"dataset"}
```


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

```
Матрица признаков
[[1.3]
 [1.6]
 [1.9]
 [2. ]
 [2.6]]
Зависимая переменная
[79.45 81.53 84.51 88.27 91.85]
```

Обработка пропущенных значений (если требуется)

```
# from sklearn.preprocessing import Imputer
# imputer = Imputer(missing_values = 'NaN', strategy = 'mean', axis =
0)
# imputer = imputer.fit(X[:, 1:3])
# X[:, 1:3] = imputer.transform(X[:, 1:3])
# print(X)
```

Обработка категориальных данных (если требуется)

Замена категории кодом (LabelEncoder)

```
# from sklearn.preprocessing import LabelEncoder
# labelencoder_y = LabelEncoder()
# print("Зависимая переменная до обработки")
# print(y)
# y = labelencoder_y.fit_transform(y)
# print("Зависимая переменная после обработки")
# print(y)
```

Применение OneHotEncoder

```
# from sklearn.preprocessing import OneHotEncoder
# labelencoder_X = LabelEncoder()
# X[:, 0] = labelencoder_X.fit_transform(X[:, 0])
# onehotencoder = OneHotEncoder(categorical_features = [0])
# X = onehotencoder.fit_transform(X).toarray()
# print("Перекодировка категориального признака")
# print(X)
```

Разделение выборки на тестовую и тренировочную

```
# from sklearn.cross_validation import train_test_split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
1/4, random_state = 0)
```

Обучение линейной модели регрессии

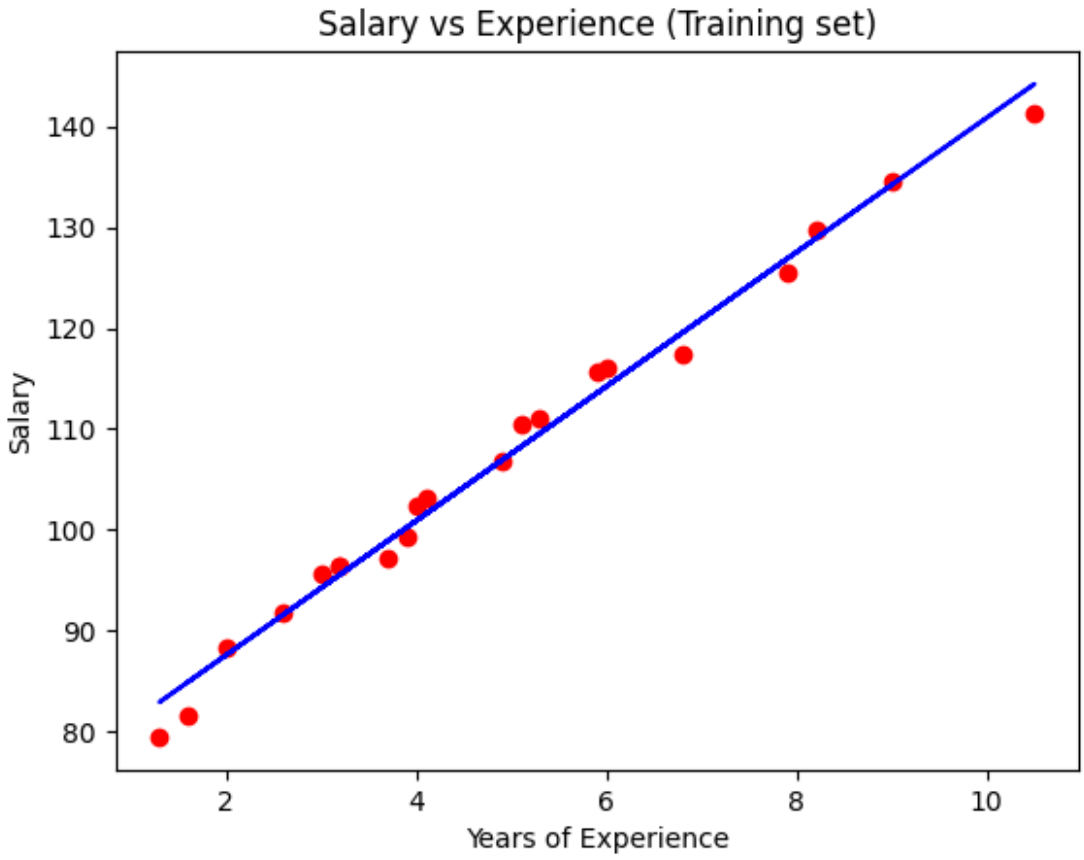
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

LinearRegression()
```

Предсказание, обработка и визуализация результатов

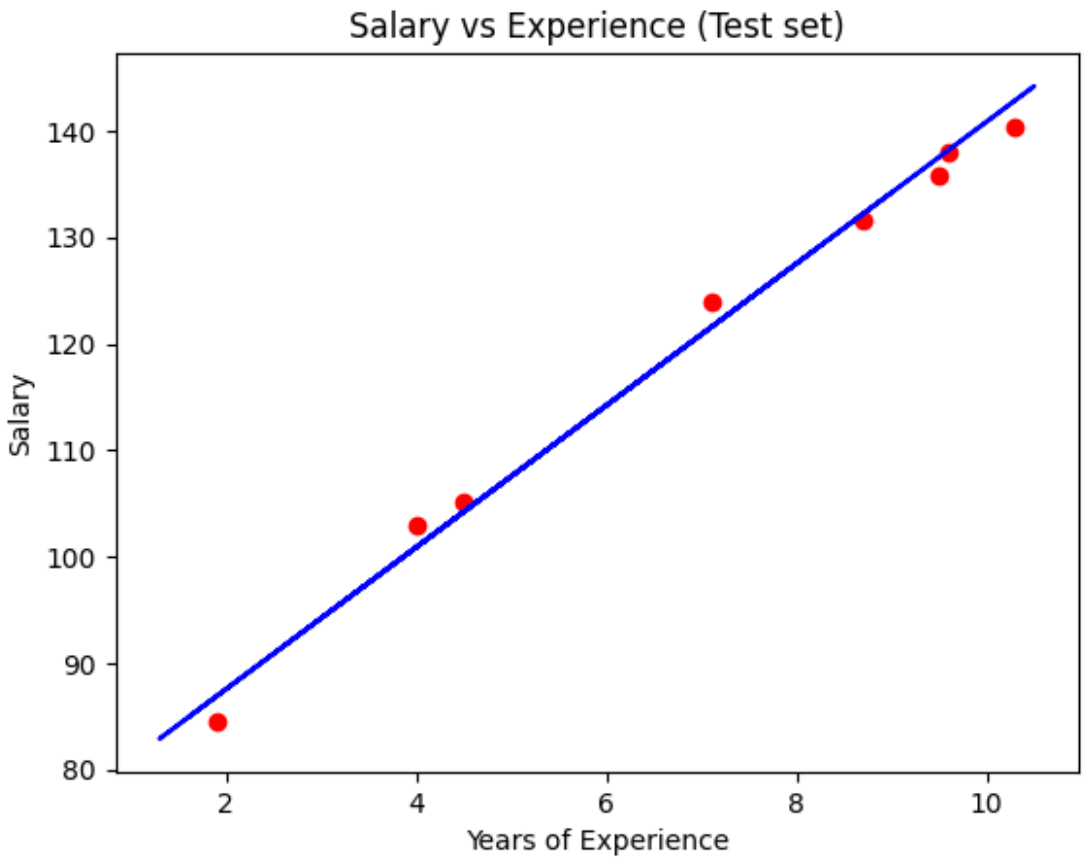
```
y_pred = regressor.predict(X_test)
print(y_pred)
```

```
[ 86.8872325  121.54762112 132.21235608 104.21742681 100.88469713
 137.54472356 142.87709104 138.21126949]
```

```
plt.scatter(X_train, y_train, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Salary vs Experience (Training set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```




```
plt.scatter(X_test, y_test, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Salary vs Experience (Test set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```


Salary vs Experience (Test set)
Salary
Years of Experience
140
130
120
110
100
90
80
2
4
6
8
10