Chương 4: Hồi quy tuyến tính (Linear Regression)

4.1 Biến

- Ví dụ: hồ sơ tiền lương, điểm thi, tuổi hoặc chiều cao của một người và giá cổ phiếu → đều thuộc danh mục Biến số
- Ví dụ: màu sắc, kết quả (Có/Không), Xếp hạng (Tốt/Kém/Trung bình).
- Ví dụ (sự tương quan):
    - công suất và quãng đường đi được của một phương tiện có thể có mối tương quan nghịch bởi vì khi chúng ta tăng công suất thì quãng đường đi được của phương tiện sẽ giảm xuống.
    - Tiền lương và số năm kinh nghiệm làm việc là một ví dụ về các biến tương quan thuận.
- Ví dụ: Hồ sơ của bốn người với dữ liệu như sau:
    - tuổi, mức lương

Sử dụng thư viện pandas

import pandas as pd data = {'Age': [20, 30, 40, 50], 'Salary':[5, 10, 15, 20]}

Đưa dữ liệu vào DataFrame

df = pd.DataFrame(data = data)

in ra để xem

df

Lưu dữ liệu vào tệp csv

df.to_csv('data.csv')

Xem thông tin các biến

df.columns

+ sử dụng thư viện spark
```python
from pyspark.sql.types
import StringType, DoubleType, IntegerType

# Tải dữ liệu
df=spark.read.csv('data.csv',inferSchema=True,header=True)
# Hiển thị dữ liệu
df.show(2)

# Chuyển đổi kiểu dữ liệu
df.withColumn('Age variance',(df['Age'].var()).cast(DoubleType())).show
(2,False)
```

Vẽ dữ liệu trên biểu đồ

- Sử dụng phương thức scatter

```
df.plot.scatter(x = 'Age', y = 'Salary', s = 100, title='Scatter plot of
Salary')
```

Bài tập hồi quy tuyến tính

Bước 0: Tạo một đối tượng SparkSession mới để sử dụng Spark

```
import findspark
findspark.init()

from pyspark.sql import SparkSession
spark=SparkSession.builder.appName('Linear Regression').getOrCreate()
```

Bước 1: Tải thư viện

```
from pyspark.ml.regression import LinearRegression
```

Bước 2: Tải dữ liệu

```
df=spark.read.csv
('Linear_regression_dataset.csv',inferSchema=True,header=True)
```

Bước 3: Khai thác và phân tích dữ liệu

Kiểm tra kích thước dữ liệu

```
print((df.count(), len(df.columns)))
```

Xem kiểu dữ liệu

```
df.printSchema()
```

Xem thống kê dữ liệu

```
df.describe().show(5,False)
```

```
df.head(5)
```

Xem mối tương quan giữa các biến đầu vào và đầu ra

```
from pyspark.sql.functions import corr
```

```
df.select(corr('var_1','output')).show()
```

Bước 4: Kỹ thuật đặc trưng

Tải thư viện

```
from pyspark.ml.linalg import Vector
from pyspark.ml.feature import VectorAssembler
```

Xem các biến đầu vào và đầu ra

```
df.columns
```

> ['var_1', 'var_2', 'var_3', 'var_4', 'var_5', 'output']

Tạo véc tơ assembler kết hợp 5 biến đầu vào gọi là feature

```
vec_assmebler = VectorAssembler(inputCols=['var_1', 'var_2', 'var_3',
'var_4',
'var_5'], outputCol='feature')

# biến đổi giá trị trong vector
features_df = vec_assmebler.transform(df)
```

Xem lại véc tơ đặc trưng

```
features_df.printSchema()
```

Bổ sung thêm cột đặc trưng

```
# xem véc tơ đặc trưng
features_df.select('features).show(5,False)
```

Tạo dữ liệu huấn luyện: gồm cột đặc trưng (features) và đầu ra (ouput)

```
data_model = features_df.select('features', 'output')
```

Xem dữ liệu huấn luyện

```
data_model.show(5,False)
```

Xem kích cỡ dữ liệu huấn luyện

```
print((data_model.count(), len(data_model.columns)))
```

Bước 5: Chia tập dữ liệu huấn luyện và kiểm tra

- Thông thường chia theo tỉ lệ 70-30 hoặc 20-80

```
train_df, test_df = data_model.randomSplit([0.7,0.3])
```

- xem kích cỡ của tập huấn luyện

```
print((train_df.count(), len(train_df.columns)))
```

- xem kích cỡ của tập kiểm tra

```
print((test_df.count(), len(test_df.columns)))
```

Bước 6: Xây dựng mô hình hồi quy tuyến tính

6.1 Tải thư viện học máy

```
from pyspark.ml.regression import LinearRegression
```

6.2 Khởi tạo mô hình

```
model = LinearRegression(labelCol='output')
```

6.3 Huấn luyện mô hình

```
model_lr =  model.fit(train_df)
```

6.4 lỖI bình phương tối thiểu trên dữ liệu huấn luyện

- Đánh giá mô hình huấn luyện

```
train_predict = model_lr.evaluate(train_df)
```

- Lỗi bình phương tối thiểu

```
train_predict.meanSquareError
```

- Lỗi R2 trên dữ liệu huấn luyện

```
print(train_predict.r2)
```

Bước 7: Đánh giá mô hình trên dữ liệu kiểm tra

7.1 Đánh giá dữ liệu kiểm tra

```
test_results = model_lr.evaluate(train_df)
```

7.2 In kết quả đánh giá

```
print(test_results.r2)
# lỗi bình phương tối thiểu
test_results.meanSquareError
```