Bike Sharing Demand

7107018013 統研碩一 郭又嘉

變數名稱	變數解釋
Datetime	hourly date + timestamp
season	1 = spring, 2 = summer, 3 = fall, 4 = winter
holiday	whether the day is considered a holiday
workingday	whether the day is neither a weekend nor holiday
weather	1: Clear, Few clouds, Partly cloudy, Partly cloudy 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
temp	temperature in Celsius
atemp	"feels like" temperature in Celsius
humidity	relative humidity
windspeed	wind speed
casual	number of non-registered user rentals initiated
registered	number of registered user rentals initiated
count	number of total rentals

資料前處理

查看是否有缺失值

```
1 df_train.toPandas().isnull().sum()
```

▸ (1) Spark Jobs

```
Out[61]:
datetime      0
season        0
holiday       0
workingday    0
weather       0
temp          0
atemp         0
humidity      0
windspeed     0
casual        0
registered    0
count         0
dtype: int64
```

查看資料類型

```
1 df_train.printSchema()
```

```
root
 |-- datetime: string (nullable = true)
 |-- season: string (nullable = true)
 |-- holiday: string (nullable = true)
 |-- workingday: string (nullable = true)
 |-- weather: string (nullable = true)
 |-- temp: string (nullable = true)
 |-- atemp: string (nullable = true)
 |-- humidity: string (nullable = true)
 |-- windspeed: string (nullable = true)
 |-- casual: string (nullable = true)
 |-- registered: string (nullable = true)
 |-- count: string (nullable = true)
```

資料前處理

- 除了「datatime」其他轉成DoubleType

把String轉成數值型資料

```
1 from pyspark.sql.types import DoubleType, IntegerType, StringType
2 from pyspark.sql.functions import col
3
4 columns = df_train.columns[1:]
5 for c in columns:
6   df_train = df_train.withColumn(c, col(c).cast(DoubleType()))
7 display(df_train)
```

- 把「datatime」切割出year、month、hour
- year：2011轉成0、2012轉成1

datetime	year_label	month	hour
2011-01-01 00:00:00	0	1	0
2011-01-01 01:00:00	0	1	1
2011-01-01 02:00:00	0	1	2
2011-01-01 03:00:00	0	1	3
2011-01-01 04:00:00	0	1	4
2011-01-01 05:00:00	0	1	5
2011-01-01 06:00:00	0	1	6
2011-01-01 07:00:00	0	1	7

處理類別型變數

```
from pyspark.ml.feature import OneHotEncoderEstimator

encoder = OneHotEncoderEstimator(inputCols=["year_label", "month", "hour", "season", "holiday", "workingday", "weather"],
                                 outputCols=["year_label_onehot", "month_onehot", "hour_onehot", "season_onehot", "holiday_onehot",
                                             "workingday_onehot", "weather_onehot"])
model = encoder.fit(df)
df1 = model.transform(df)
display(df1)
```

選取類別型變數轉Dummy Variable

```
root
 |-- temp: double (nullable = true)
 |-- atemp: double (nullable = true)
 |-- humidity: double (nullable = true)
 |-- windspeed: double (nullable = true)
 |-- season_onehot: vector (nullable = true)
 |-- holiday_onehot: vector (nullable = true)
 |-- year_label_onehot: vector (nullable = true)
 |-- month_onehot: vector (nullable = true)
 |-- weather_onehot: vector (nullable = true)
 |-- hour_onehot: vector (nullable = true)
 |-- workingday_onehot: vector (nullable = true)
 |-- label: double (nullable = true)
```

查看變數之間的相關性

- 「count」與「casual」、「registered」之間相關性高
  ➔ 可發現「count」= 「casual」+「registered」

- 「atemp」與「temp」相關係數為 0.9849，擇一放入模型

```
Pearson correlation matrix:
DenseMatrix([[ 1.        ,  0.69041357,  0.97094811,  0.39445364,  0.38978444,
              -0.31737148,  0.10136947],
             [ 0.69041357,  1.        ,  0.49724969,  0.46709706,  0.46206654,
              -0.3481869 ,  0.09227619],
             [ 0.97094811,  0.49724969,  1.        ,  0.31857128,  0.31463539,
              -0.26545787,  0.09105166],
             [ 0.39445364,  0.46709706,  0.31857128,  1.        ,  0.98494811,
              -0.06494877, -0.01785201],
             [ 0.38978444,  0.46206654,  0.31463539,  0.98494811,  1.        ,
              -0.04353571, -0.057473  ],
             [-0.31737148, -0.3481869 , -0.26545787, -0.06494877, -0.04353571,
               1.        , -0.31860699],
             [ 0.10136947,  0.09227619,  0.09105166, -0.01785201, -0.057473  ,
              -0.31860699,  1.        ]])
```

- 刪除無用數據

temp	humidity	windspeed	season_onehot	holiday_onehot	year_label_onehot	month_onehot	weather_onehot	hour_onehot	workingday_onehot	label
9.84	81	0	[0,4,[1],[1]]	[0,1,[0],[1]]	[0,1,[0],[1]]	[0,12,[1],[1]]	[0,4,[1],[1]]	[0,23,[0],[1]]	[0,1,[0],[1]]	16
9.02	80	0	[0,4,[1],[1]]	[0,1,[0],[1]]	[0,1,[0],[1]]	[0,12,[1],[1]]	[0,4,[1],[1]]	[0,23,[1],[1]]	[0,1,[0],[1]]	40
9.02	80	0	[0,4,[1],[1]]	[0,1,[0],[1]]	[0,1,[0],[1]]	[0,12,[1],[1]]	[0,4,[1],[1]]	[0,23,[2],[1]]	[0,1,[0],[1]]	32
9.84	75	0	[0,4,[1],[1]]	[0,1,[0],[1]]	[0,1,[0],[1]]	[0,12,[1],[1]]	[0,4,[1],[1]]	[0,23,[3],[1]]	[0,1,[0],[1]]	13
9.84	75	0	[0,4,[1],[1]]	[0,1,[0],[1]]	[0,1,[0],[1]]	[0,12,[1],[1]]	[0,4,[1],[1]]	[0,23,[4],[1]]	[0,1,[0],[1]]	1
9.84	75	6.0032	[0,4,[1],[1]]	[0,1,[0],[1]]	[0,1,[0],[1]]	[0,12,[1],[1]]	[0,4,[2],[1]]	[0,23,[5],[1]]	[0,1,[0],[1]]	1
9.02	80	0	[0,4,[1],[1]]	[0,1,[0],[1]]	[0,1,[0],[1]]	[0,12,[1],[1]]	[0,4,[1],[1]]	[0,23,[6],[1]]	[0,1,[0],[1]]	2
8.2	86	0	[0,4,[1],[1]]	[0,1,[0],[1]]	[0,1,[0],[1]]	[0,12,[1],[1]]	[0,4,[1],[1]]	[0,23,[7],[1]]	[0,1,[0],[1]]	3
9.84	75	0	[0,4,[1],[1]]	[0,1,[0],[1]]	[0,1,[0],[1]]	[0,12,[1],[1]]	[0,4,[1],[1]]	[0,23,[8],[1]]	[0,1,[0],[1]]	8

- 對數值型資料做標準化

temp	humidity	windspeed	season_onehot	holiday_onehot	year_label_onehot	month_onehot	weather_onehot	hour_onehot	workingday_one
-1.3335994358032517	0.9931674345025104	-1.5664629008057203	[0,4,[1],[1]]	[0,1,[0],[1]]	[0,1,[0],[1]]	[0,12,[1],[1]]	[0,4,[1],[1]]	[0,23,[0],[1]]	[0,1,[0],[1]]
-1.4388411151548357	0.9412059758972607	-1.5664629008057203	[0,4,[1],[1]]	[0,1,[0],[1]]	[0,1,[0],[1]]	[0,12,[1],[1]]	[0,4,[1],[1]]	[0,23,[1],[1]]	[0,1,[0],[1]]
-1.4388411151548357	0.9412059758972607	-1.5664629008057203	[0,4,[1],[1]]	[0,1,[0],[1]]	[0,1,[0],[1]]	[0,12,[1],[1]]	[0,4,[1],[1]]	[0,23,[2],[1]]	[0,1,[0],[1]]
-1.3335994358032517	0.6813986828710117	-1.5664629008057203	[0,4,[1],[1]]	[0,1,[0],[1]]	[0,1,[0],[1]]	[0,12,[1],[1]]	[0,4,[1],[1]]	[0,23,[3],[1]]	[0,1,[0],[1]]
-1.3335994358032517	0.6813986828710117	-1.5664629008057203	[0,4,[1],[1]]	[0,1,[0],[1]]	[0,1,[0],[1]]	[0,12,[1],[1]]	[0,4,[1],[1]]	[0,23,[4],[1]]	[0,1,[0],[1]]
-1.3335994358032517	0.6813986828710117	-0.8312049616085506	[0,4,[1],[1]]	[0,1,[0],[1]]	[0,1,[0],[1]]	[0,12,[1],[1]]	[0,4,[2],[1]]	[0,23,[5],[1]]	[0,1,[0],[1]]
-1.4388411151548357	0.9412059758972607	-1.5664629008057203	[0,4,[1],[1]]	[0,1,[0],[1]]	[0,1,[0],[1]]	[0,12,[1],[1]]	[0,4,[1],[1]]	[0,23,[6],[1]]	[0,1,[0],[1]]
-1.54408279450642	1.2529747275287595	-1.5664629008057203	[0,4,[1],[1]]	[0,1,[0],[1]]	[0,1,[0],[1]]	[0,12,[1],[1]]	[0,4,[1],[1]]	[0,23,[7],[1]]	[0,1,[0],[1]]

Forward selection

Linear Regression

```
Add  hour_onehot     with R2 0.52150294863182788951
Add  temp            with R2 0.59801414551935705433
Add  year_label_onehot with R2 0.65672636154639185 69
Add  month_onehot    with R2 0.67367829077289598239
Add  weather_onehot  with R2 0.68990366993962681796
Add  humidity        with R2 0.69310056744913661753
Add  holiday_onehot  with R2 0.69321743978346928028
Add  workingday_onehot with R2 0.69322680189390151195
```

Generalized Linear Regression (family=Poisson)

```
Add  hour_onehot     with AIC 135036.94036944629624
Add  temp            with AIC 133156.64791820058599
Add  year_label_onehot with AIC 131429.86909403282334
Add  month_onehot    with AIC 130899.71248453833687
Add  weather_onehot  with AIC 130362.15025211582542
Add  humidity        with AIC 130233.97136275662342
Add  windspeed       with AIC 130222.51453624089481
Add  holiday_onehot  with AIC 130222.49716141901445
```

Response Type： Count

訓練模型

切割70/30 的 training, testing data
選取特徵項，將特徵項合併成「feature」

Linear Regression R^2=0.692389

```
1  from pyspark.ml.regression import LinearRegression
2
3  lr = LinearRegression(maxIter=10, regParam=0.3, elasticNetParam=0.8)
4  lrModel = lr.fit(training_lr)
5
6  #print("Coefficients: %s" % str(lrModel.coefficients))
7  #print("Intercept: %s" % str(lrModel.intercept))
8
9  trainingSummary = lrModel.summary
10 print("RMSE: %f" % trainingSummary.rootMeanSquaredError)
11 print("r2: %f" % trainingSummary.r2)
```

▸ (5) Spark Jobs

```
RMSE: 100.824586
r2: 0.692389
```

label	prediction
580	512.9306860195825
583	502.83994460346145
237	480.2213130093696
271	475.27483028845046
446	472.84859781736213
297	466.7591980235349
399	450.1471097927048
297	437.4526882324427

訓練模型

Generalized Linear Regression (Poisson)

```
from pyspark.ml.regression import GeneralizedLinearRegression
glr = GeneralizedLinearRegression(featuresCol = 'features',labelCol = 'label',\
                                  family="Poisson", link="identity", maxIter=10, regParam=0.3)

model_Poisson = glr.fit(training_glr)

#print("Coefficients: " + str(model_Poisson.coefficients))
#print("Intercept: " + str(model_Poisson.intercept))

aic = model_Poisson.summary.aic
print("AIC: " + str(aic))
```

▸ (2) Spark Jobs

```
RMSE on test data = 164.527
R2 on test data = 0.185836
```

label	prediction
711	186.65239862198723
693	189.05310700350688
680	197.88474252312145
679	157.06814786680013
659	173.936819246623
648	200.95808365862786
638	226.15789021566758
638	213.7904651906482

訓練模型

Decision Tree Regressor

▸ (2) Spark Jobs

```
RMSE on test data = 13.6999
R2 on test data = 0.99427
```

label	prediction
638	785.2258064516129
553	574.1908396946565
569	574.1908396946565
564	574.1908396946565
586	574.1908396946565
587	574.1908396946565
533	558.6567164179105
550	558.6567164179105

Random Forest Regressor

▸ (2) Spark Jobs

```
RMSE on test data = 18.2812
R2 on test data = 0.989798
```

label	prediction
638	691.4157180897259
564	552.6776829799161
586	552.6776829799161
587	552.6776829799161
569	548.5554035342997
579	545.133026584994
550	544.5010085734559
538	543.1725833575014