Introduction to Spark ML with Python

韓奐宇

Big concept

- Why Spark ML?

Faster calculation for “big data” machine learning.

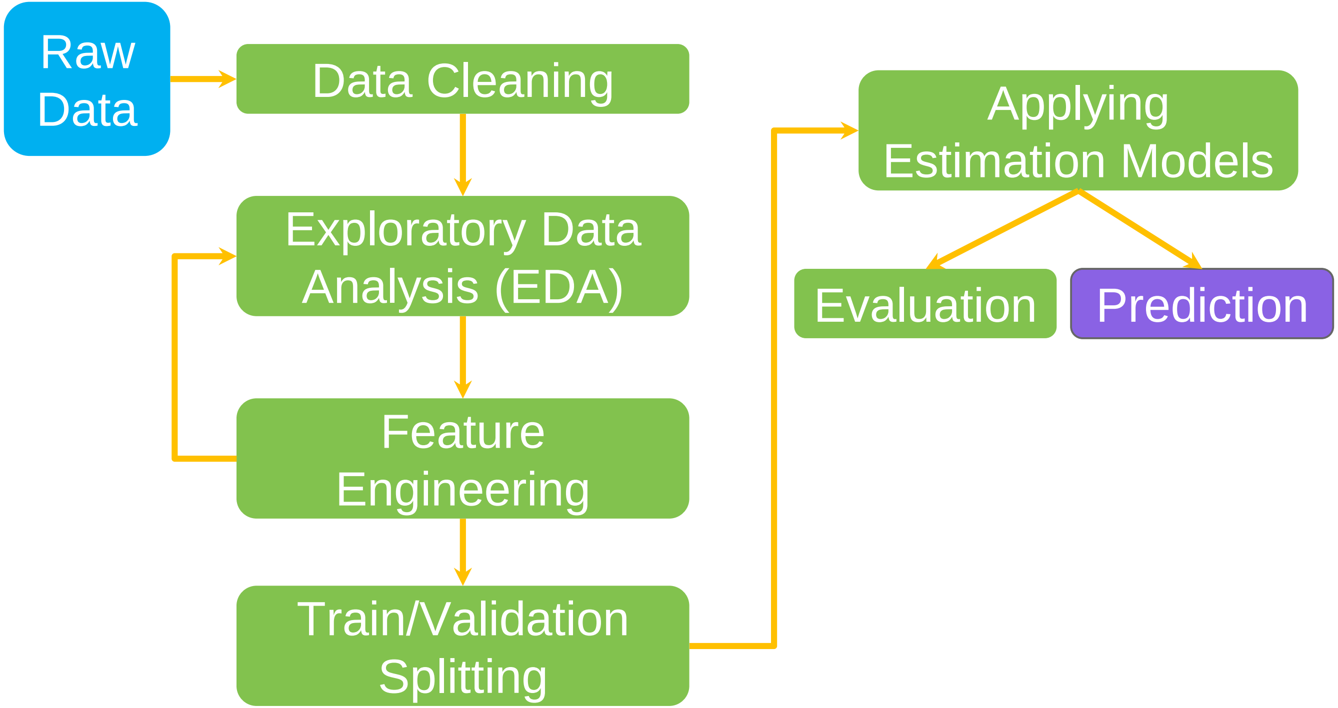
- Why Python?

Easier to program and high integration.

1

An Overview of Making a Prediction

An Overview of Making a Prediction



An Overview of Making a Prediction

➢ **Preprocessing Task**

- Data Cleaning
- Data Transformation
- Data Reduction

Start a Spark Session

Read from file

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import lit

spark = SparkSession \
        .builder.appName('sql') \
        .getOrCreate()

training = spark.read.csv('train.csv',inferSchema=True,header=True)
testing = spark.read.csv('test.csv',inferSchema=True,header=True)
testing = testing.withColumn('Survived',lit(None)).select([
                          'PassengerId',
                          'Survived',
                          'Pclass',
                          'Name',
                          'Sex',
                          'Age',
                          'SibSp',
                          'Parch',
                          'Ticket',
                          'Fare',
                          'Cabin',
                          'Embarked'
                          ])

data = training.union(testing)
```

Start a Spark Session

Read from Hive

```
import pyspark

spark = pyspark.sql.SparkSession \
        .builder.appName('sql') \
        .enableHiveSupport() \
        .getOrCreate()

data = spark.sql("select * from mydatabase.mytable")
```

# train.csv PassengerId	# train.csv Survived	# train.csv Pclass	Abc train.csv Name	Abc train.csv Sex	# train.csv Age	# train.csv SibSp	# train.csv Parch	Abc train.csv Ticket	# train.csv Fare	Abc train.csv Cabin	Abc train.csv Embarked
1	0	3	Braund, Mr. Owe...	male	22.0000	1	0	A/5 21171	7.250	*null*	S
2	1	1	Cumings, Mrs. Jo...	female	38.0000	1	0	PC 17599	71.283	C85	C
3	1	3	Heikkinen, Miss. ...	female	26.0000	0	0	STON/O2. 31012...	7.925	*null*	S
4	1	1	Futrelle, Mrs. Jac...	female	35.0000	1	0	113803	53.100	C123	S
5	0	3	Allen, Mr. Willia...	male	35.0000	0	0	373450	8.050	*null*	S
6	0	3	Moran, Mr. James	male	*null*	0	0	330877	8.458	*null*	Q
7	0	1	McCarthy, Mr. Ti...	male	54.0000	0	0	17463	51.863	E46	S
8	0	3	Palsson, Master. ...	male	2.0000	3	1	349909	21.075	*null*	S
9	1	3	Johnson, Mrs. Os...	female	27.0000	0	2	347742	11.133	*null*	S
10	1	2	Nasser, Mrs. Nic...	female	14.0000	1	0	237736	30.071	*null*	C
11	1	3	Sandstrom, Miss...	female	4.0000	1	1	PP 9549	16.700	G6	S
12	1	1	Bonnell, Miss. Eli...	female	58.0000	0	0	113783	26.550	C103	S
13	0	3	Saundercock, Mr....	male	20.0000	0	0	A/5. 2151	8.050	*null*	S

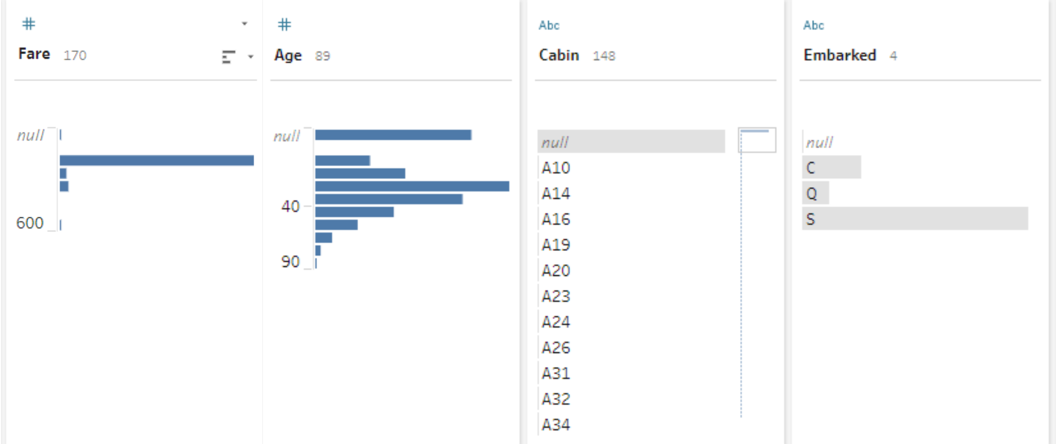
Raw data

Data cleaning recap

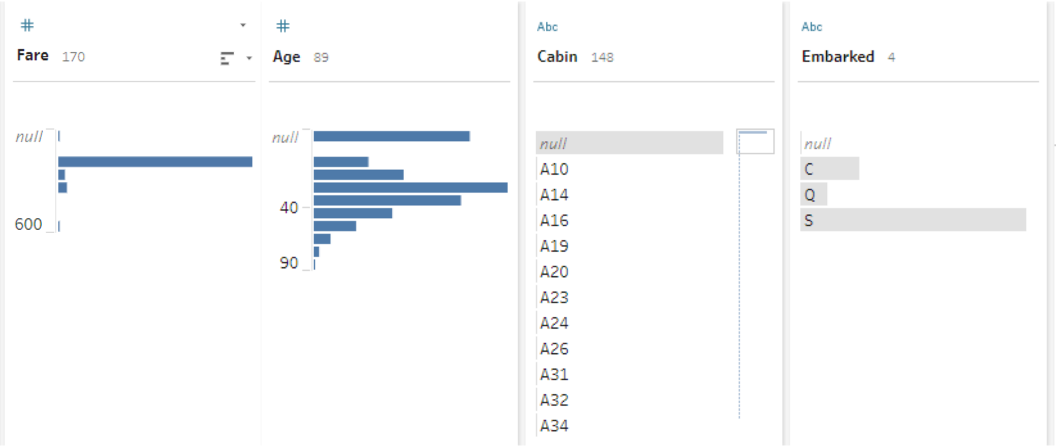
- Missing Values
  - Drop the missing data
  - Replace them by certain statistical values
  - Label them as the missing value

mean /
median /
mode /
clustering /
modeling methods

- Outlier Detection
- Redundant Features
  - We usually remove them

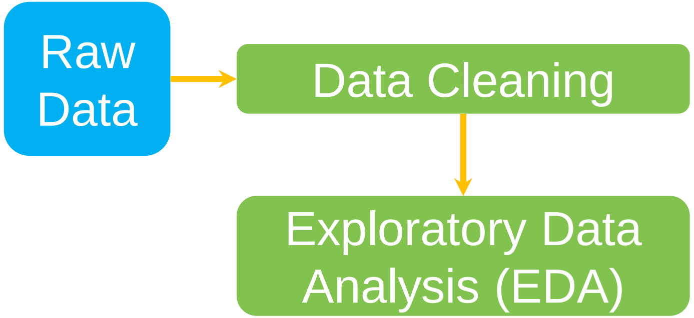


Missing Values



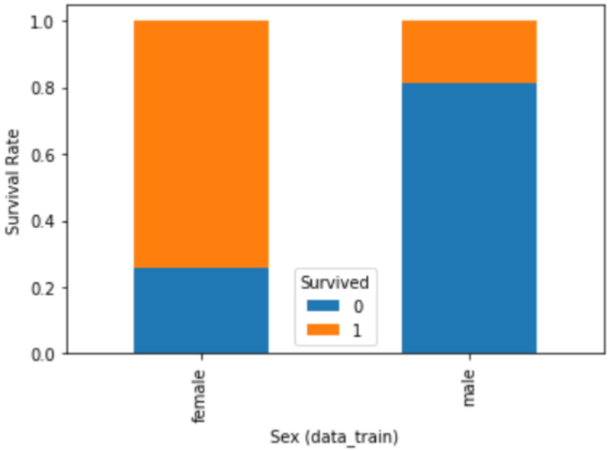

```
data = data.drop('Name')

data = data.drop('Cabin')

data_age = data.select('Age').dropna()
age_avg = data_age.agg({"Age":"avg"}).collect()[0][0]
data = data.fillna(age_avg, subset=['Age'])

data_age = data.select('Fare').dropna()
age_avg = data_age.agg({"Fare":"avg"}).collect()[0][0]
data = data.fillna(age_avg, subset=['Fare'])

data = data.fillna('NULL', subset=['Embarked'])
```

An Overview of Making a Prediction

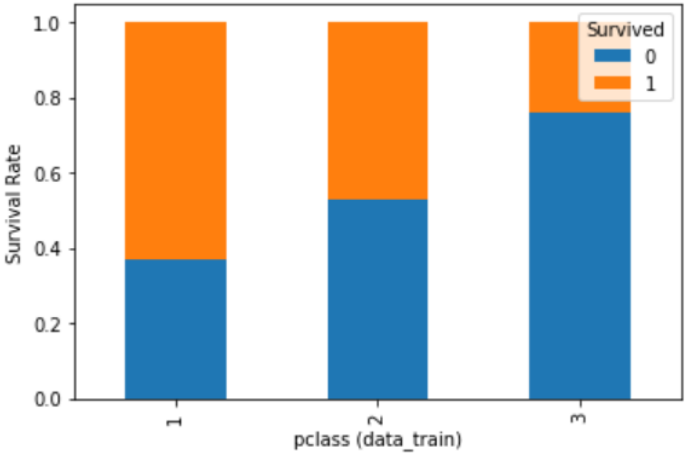


Exploratory Data Analysis(EDA)

- Helps to gain familiarity with dataset
  - Identify features distribution
  - Identify features with null or erroneous values
  - Identify features that are important or not

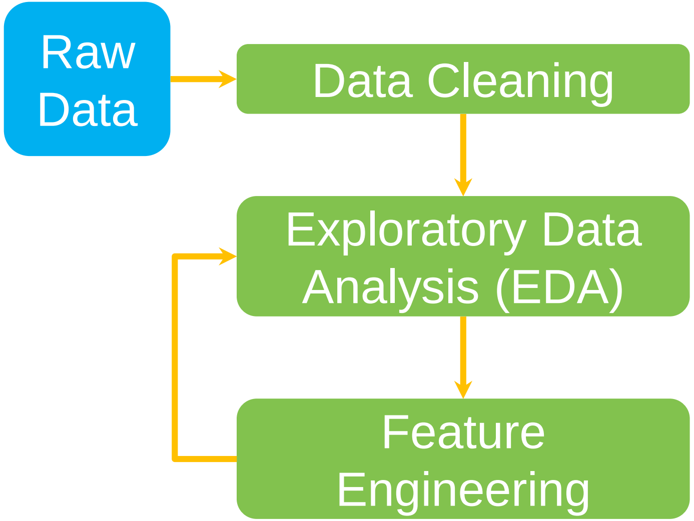


Survival Rate
1.0
0.8
0.6
0.4
0.2
0.0
Survived
0
1
female
male
Sex (data_train)



Survival Rate
1.0
0.8
0.6
0.4
0.2
0.0
Survived
0
1
1
2
3
pclass (data_train)

An Overview of Making a Prediction

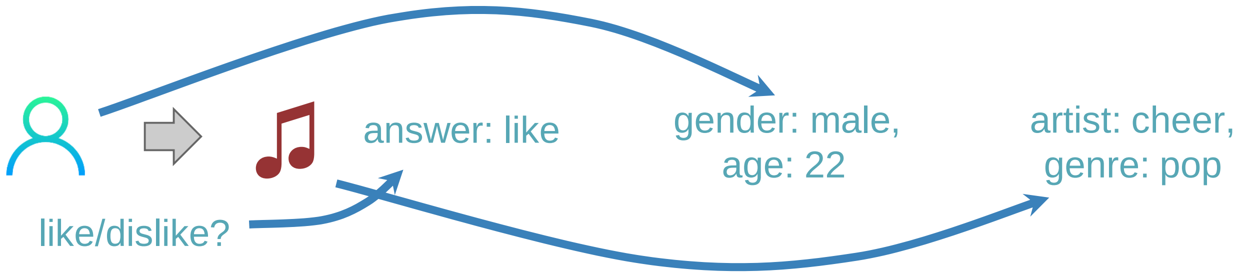


RoadMap

- Feature Engineering
  - Feature Encoding
    - Binary Features
    - Numeric Features
    - Categorical Features

Feature Engineering

- Feature engineering is the process of using domain knowledge of the data **to create features** that make machine learning algorithms work.
(https://en.wikipedia.org/wiki/Feature_engineering)



like/dislike = gender*w1 + age*w2 + artist*w3 + genre*w4

Feature Engineering

- Convert the extracted features to be readable by applied machine learning model.

like/dislike = gender*w1 + age*w2 + artist*w3 + genre*w4

$$\Downarrow$$

1 / 0 = (1 , 0)*w1 + (0~100)*w2 + (0~30)*w3 + (0~50)*w4

Binarization

	Gender
User A	male
User B	male
User C	female

0 for male
1 for female

	Gender
User A	0
User B	0
User C	1

	Age
User A	16
User B	25
User C	31

0 for <= 18
1 for > 18

	Adult
User A	0
User B	1
User C	1

Binarization

```
from pyspark.ml.feature import Binarizer

binarizer = Binarizer(inputCol='Age',outputCol='AgeBin',threshold=15)
data = binarizer.transform(data)
```

```
+-----------------+------+
|              Age|AgeBin|
+-----------------+------+
|             22.0|   1.0|
|             38.0|   1.0|
|             26.0|   1.0|
|             35.0|   1.0|
|             35.0|   1.0|
|29.69911764705882|   1.0|
|             54.0|   1.0|
|              2.0|   0.0|
+-----------------+------+
```

Categotical Features

	Artist
User A	Jack
User B	Peter
User C	Lee

Label Encoding

	Artist
User A	0
User B	1
User C	2

Categotical Features

```
from pyspark.ml.feature import StringIndexer

tk_indxer = StringIndexer(inputCol='Ticket',outputCol='TicketIndex')
sex_indxer = StringIndexer(inputCol='Sex',outputCol='SexIndex')
data = tk_indxer.fit(data).transform(data)
data = sex_indxer.fit(data).transform(data)
```

```
+----------------+-----------+------+--------+
|          Ticket|TicketIndex|   Sex|SexIndex|
+----------------+-----------+------+--------+
|       A/5 21171|      257.0|  male|     0.0|
|        PC 17599|      608.0|female|     1.0|
|STON/O2. 3101282|      292.0|female|     1.0|
|          113803|       46.0|female|     1.0|
|          373450|      425.0|  male|     0.0|
+----------------+-----------+------+--------+
```

Categotical Features

	Artist
User A	Jack
User B	Peter
User C	Lee

One-hot Encoding

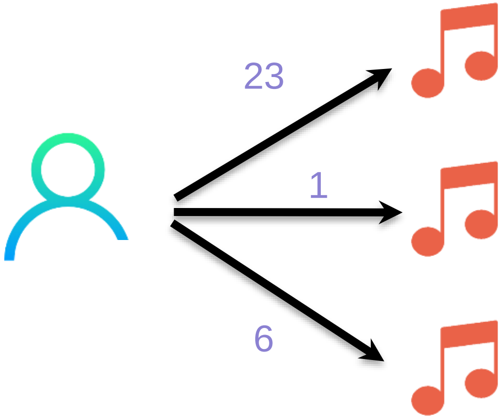
	Jack	Peter	Lee
User A	1	0	0
User B	0	1	0
User C	0	0	1

Categotical Features

```
from pyspark.ml.feature import StringIndexer,OneHotEncoderEstimator

em_indexer = StringIndexer(inputCol='Embarked',outputCol='EmbarkedIndex')
encoder = OneHotEncoderEstimator(inputCols=['EmbarkedIndex'],
                                  outputCols=['EmbarkedOneHot'])
data = em_indexer.fit(data).transform(data)
data = encoder.fit(data).transform(data)
```

```
+--------+--------------+
|Embarked|EmbarkedOneHot|
+--------+--------------+
|       S| (3,[0],[1.0])|
|       C| (3,[1],[1.0])|
|       S| (3,[0],[1.0])|
|       S| (3,[0],[1.0])|
|       S| (3,[0],[1.0])|
+--------+--------------+
```

Numerical Features



	R1	R2	R3
count	23	1	6

binary	1	0	1

probability	23/30	1/30	6/30

Numerical Features

- Standardization
- Normalization
- Rescaling

Numerical Features

```
fare_mean = data.agg({"Fare":"mean"}).collect()[0][0]
fare_std = data.agg({"Fare":"stddev"}).collect()[0][0]
data = data.withColumn("FareStd",(data['Fare'] - fare_mean) / fare_std)
```

```
+-------+-------------------+
|   Fare|            FareStd|
+-------+-------------------+
|   7.25|-0.5021631365156046|
|71.2833| 0.7864036178345 39|
|  7.925|-0.4885798515812604|
|   53.1|  0.420494069765 41|
|   8.05|-0.4860644284452707|
+-------+-------------------+
```

Advanced Feature Engineering

- Feature Extraction
  - Feature interactions
  - Data Mining
  - Dimensional Reduction
  - Domain-specific Process

Feature Interactions

like/dislike = gender*w1 + age*w2 + artist*w3 + genre*w4 + (gender AND genre)*w5

Meaning Behind the Observed Features

- 2018/12/25

  Holiday? Weekday?

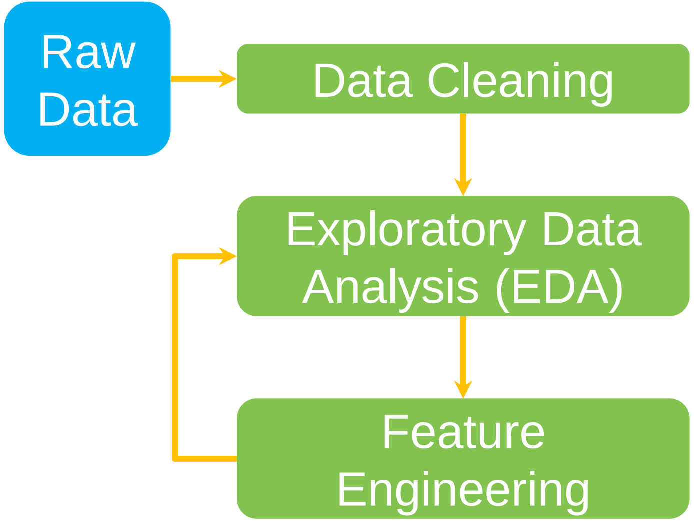
  Day? Night?

- Taipei

  Asia

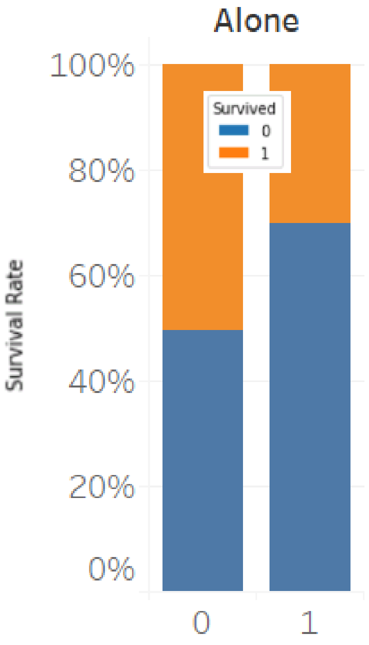
  Mandarin

An Overview of Making a Prediction



EDA After Feature Engineering

(SibSp AND Parch)



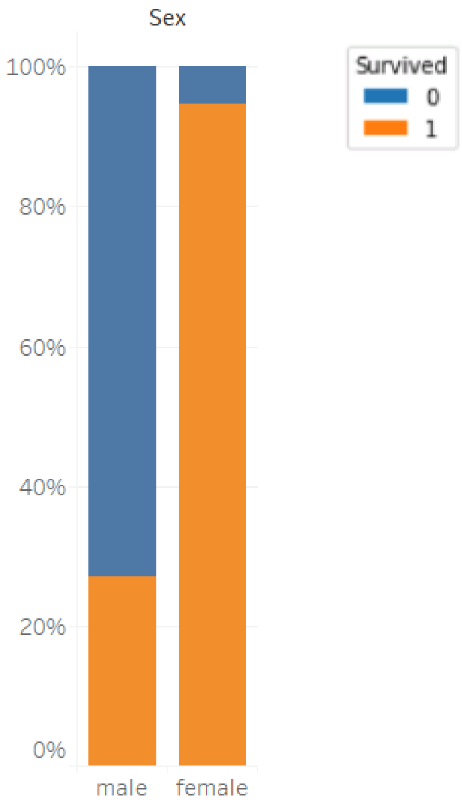
EDA After Feature Engineering

```
from pyspark.sql.functions import when
data = data.withColumn("SibSpParch",
                       when((data['SibSp'] == 0) & (data['Parch'] == 0) ,1 )
                       .otherwise(0))
```

```
+-----+-----+----------+
|SibSp|Parch|SibSpParch|
+-----+-----+----------+
|    1|    0|         0|
|    1|    0|         0|
|    0|    0|         1|
|    1|    0|         0|
|    0|    0|         1|
|    0|    0|         1|
|    0|    0|         1|
|    3|    1|         0|
|    0|    2|         0|
|    1|    0|         0|
+-----+-----+----------+
```

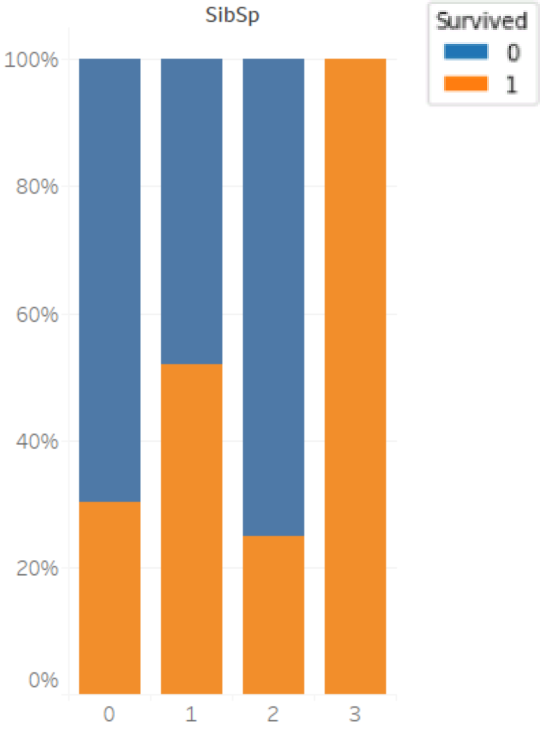
EDA After Feature Engineering

(Pclass = 1 OR 2) AND Sex = female



EDA After Feature Engineering

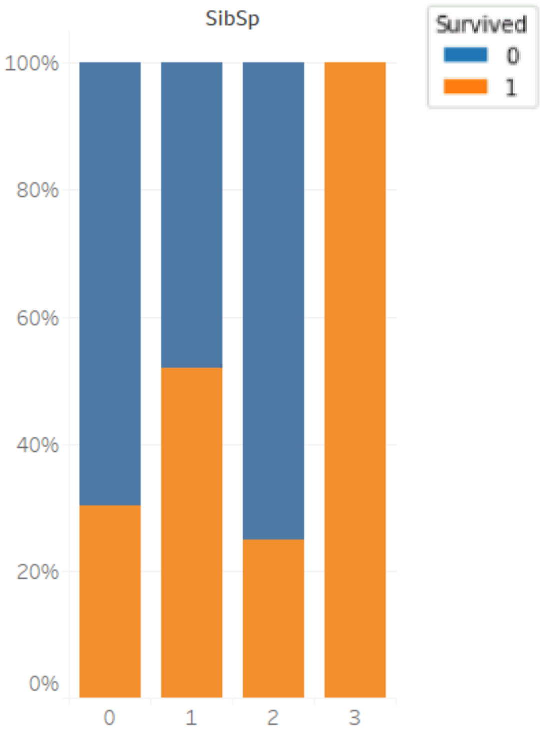
Parch = 0 AND SibSp = 3

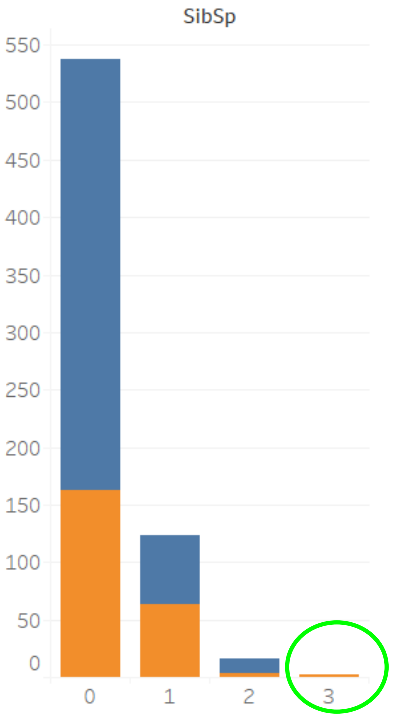


EDA After Feature Engineering

~~Parch = 0 AND SibSp = 3~~

Too small



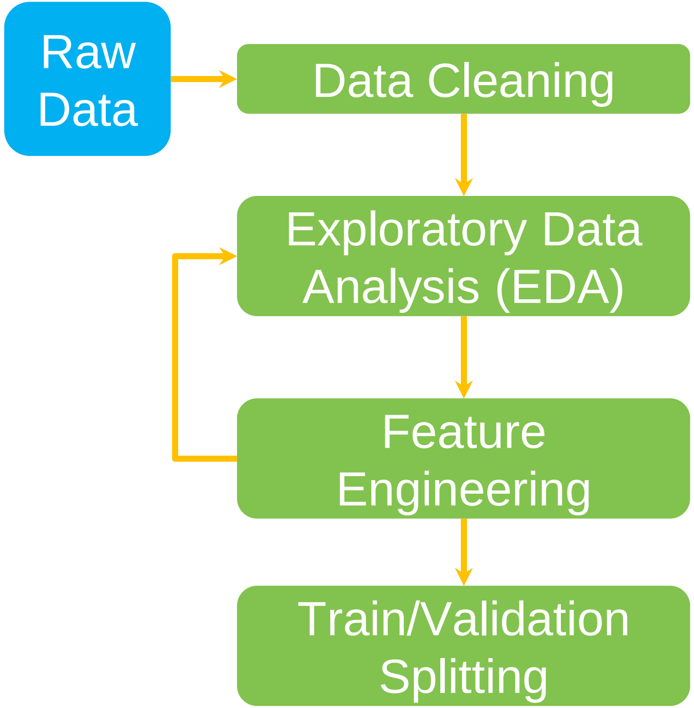


Prepare for modeling

```
from pyspark.ml.feature import VectorAssembler

assembler = VectorAssembler(inputCols = [
 'Pclass',
 'Age',
 'AgeBin',
 'SibSp',
 'Parch',
 'Fare',
 'TicketIndex',
 'SexIndex',
 'EmbarkedOneHot',
 'FareStd',
 'SibSpParch'], outputCol='features')
data = assembler.transform(data)
```

```
+-------------------------------------------------------------------------+
|features                                                                 |
+-------------------------------------------------------------------------+
|[3.0,22.0,1.0,1.0,0.0,7.25,257.0,0.0,1.0,0.0,0.0,-0.502163136515604 6,0.0] |
|[1.0,38.0,1.0,1.0,0.0,71.2833,608.0,1.0,0.0,1.0,0.0,0.786403617834539,0.0]|
|[3.0,26.0,1.0,0.0,0.0,7.925,292.0,1.0,1.0,0.0,0.0,-0.4885798515812604,1.0]|
|[1.0,35.0,1.0,1.0,0.0,53.1,46.0,1.0,1.0,0.0,0.0,0.42049406976541,0.0]    |
|[3.0,35.0,1.0,0.0,0.0,8.05,425.0,0.0,1.0,0.0,0.0,-0.4860644284452707,1.0] |
+-------------------------------------------------------------------------+
```

An Overview of Making a Prediction



- **Cross-Validation**
  - Random Splitting
  - Split by Time
  - Split by ID

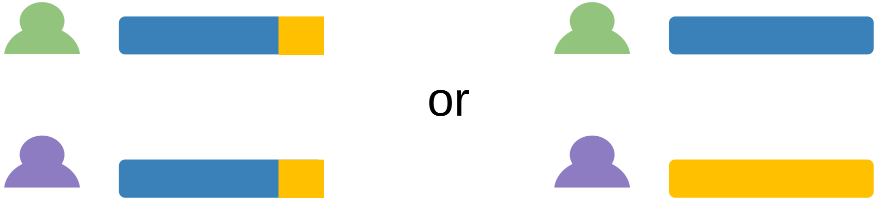
Hold A Proper Validation

- Train
- Validation
- Test

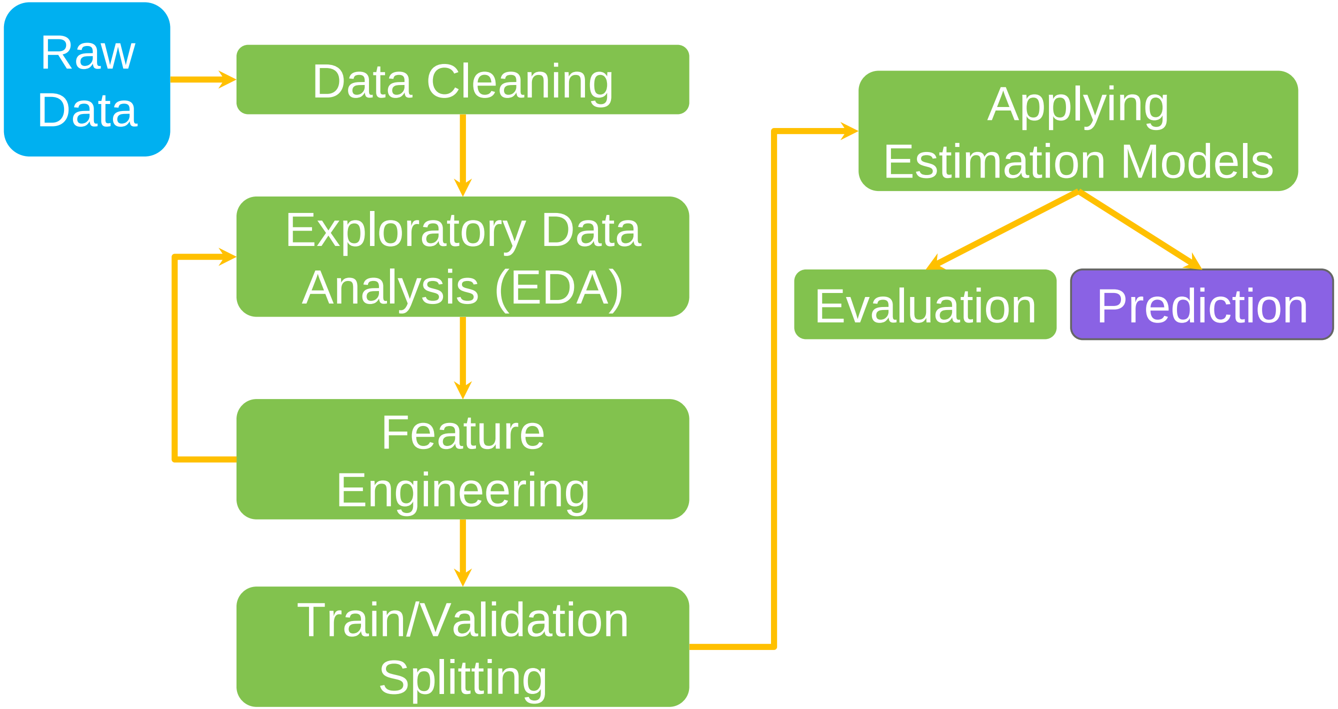
Random Spitting

Split by Time

Split by ID



An Overview of Making a Prediction



Applying Estimation Models

```
from pyspark.ml.classification import LogisticRegression
from pyspark.ml.evaluation import BinaryClassificationEvaluator
from pyspark.ml.tuning import ParamGridBuilder, CrossValidator

lgr = LogisticRegression(labelCol="Survived", featuresCol="features"
                         ,maxIter=3000)
paramGrid = ParamGridBuilder().build()

evaluator = BinaryClassificationEvaluator(labelCol="Survived")
cv = CrossValidator (estimator=lgr,
                           estimatorParamMaps=paramGrid,
                           evaluator=evaluator,
                           numFolds=5)
train = data.filter(data['Survived'].isNotNull())
test = data.filter(data['Survived'].isNull())
model = cv.fit(train)
results = model.transform(test).select("PassengerId", "prediction")

results.coalesce(1).write.format('csv').save('results',header=True)
```

4378	new	**hanbarry**	0.78468	10	2d

Your Best Entry ↑

Lab 1

Data: food.csv

Features:

- A
- B
- C
- D
- Spoiled

請分析不同成分比例的食品，哪個成分影響腐壞與否最大?

✓ 注意題目，我們不預測

Lab 2

Data: Beijing PM2.5 Data

Features:

…

- PM25 (Target)
- DEWP
- TEMP
- PRES
- cbwd

…

請使用今天的氣象資料，
預測明天的**PM2.5**

✓ 將資料平移一天

2 特徵工程常見方法

Binarization

	Age
User A	16
User B	25
User C	31

	Adult
User A	0
User B	1
User C	1

0 for <= 18

1 for > 18

Binarization

	Color
User A	Red
User B	Blue
User C	N/A

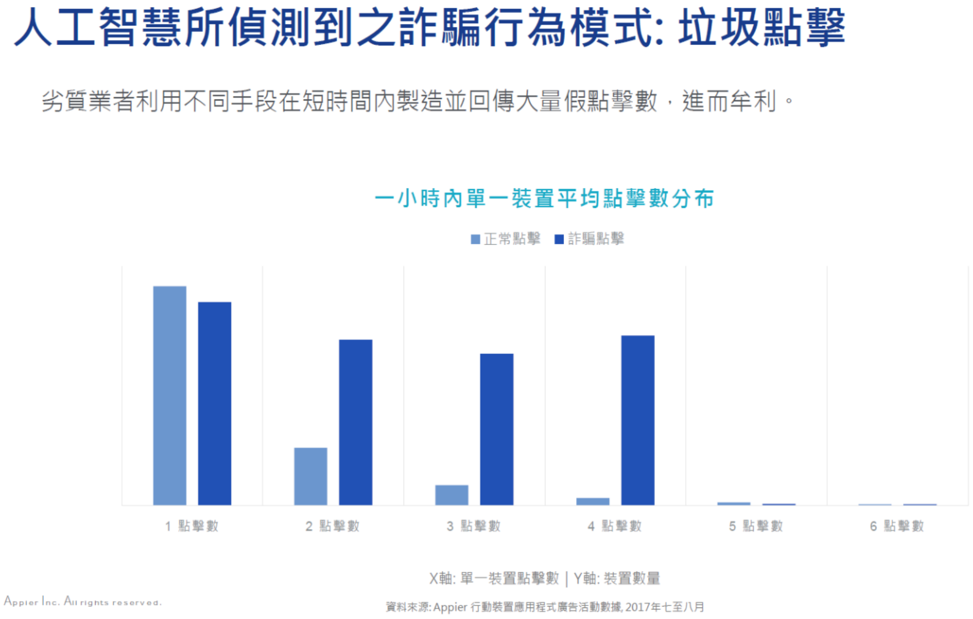
	has_color
User A	1
User B	1
User C	0

Bin-Counting

	Ans	Views	Clicks	CTR
AD_1	1	100	5	0.0500
AD_2	1	220	7	0.0318
AD_3	0	413	1	0.0024

Feature Construction

- 通常用來對**log**做加工(重複行為的整理)。
- 表示「趨勢」的特徵。



Features Interaction

- 針對numerical 特徵

	Ans	SibSp	Parch	Family_Size
User A	1	0	1	2
User B	1	1	2	4
User C	0	0	0	1

包含自己

Features Combination

- 針對categorical 特徵

	Ans	gender	Pclass	gender_Pclass
User A	1	male	1	1
User B	0	male	3	2
User C	1	femal	1	3
User D	1	femal	2	4

2(性別) * 3(船艙) = 6種組合

Features Combination

- 同時針對 categorical 與 numerical 加工

	Ans	product	price	prod_median	price_median_diff
User A	1	P1	110	110	0
User B	0	P2	250	250	0
User C	0	P1	130	110	20
User D	1	P1	70	110	-40

Lab 3

Data: Telco-Customer-Churn.csv

Features:

...

- OnlineSecurity
- OnlineBackup
- tenure
- Churn (Target)

...

請預測客戶是否流失

3 Introduction to NLP

Token

Can I convert montra helicon D to a mountain bike by just changing the tyres?	
How did Otto von Guericke used the Magdeburg hemispheres?	
Why does velocity affect time? Does velocity affect space geometry?	

Token

Can I convert montra helicon D to a mountain bike by just changing the tyres?	[can, i, convert, montra, helicon, d, to, a, mountain, bike, by, just, changing, the, tyres?]
How did Otto von Guericke used the Magdeburg hemispheres?	[how, did, otto, von, guericke, used, the, magdeburg, hemispheres?]
Why does velocity affect time? Does velocity affect space geometry?	[why, does, velocity, affect, time?, does, velocity, affect, space, geometry?]

Stop Word Remover

	[can, i, convert, montra, helicon, d, to, a, mountain, bike, by, just, changing, the, tyres?]
	[how, did, otto, von, guericke, used, the, magdeburg, hemispheres?]
	[why, does, velocity, affect, time?, does, velocity, affect, space, geometry?]

Stop Word Remover

[convert, montra, helicon, d, mountain, bike, changing, tyres?]	[can, i, convert, montra, helicon, d, to, a, mountain, bike, by, just, changing, the, tyres?]
[otto, von, guericke, used, magdeburg, hemispheres?]	[how, did, otto, von, guericke, used, the, magdeburg, hemispheres?]
[velocity, affect, time?, velocity, affect, space, geometry?]	[why, does, velocity, affect, time?, does, velocity, affect, space, geometry?]

NGram

[convert, montra, helicon, d, mountain, bike, changing, tyres?]	
[otto, von, guericke, used, magdeburg, hemispheres?]	
[velocity, affect, time?, velocity, affect, space, geometry?]	

NGram

[convert, montra, helicon, d, mountain, bike, changing, tyres?]	[convert montra helicon, montra helicon d, helicon d mountain, d mountain bike, mountain bike changing, bike changing tyres?]
[otto, von, guericke, used, magdeburg, hemispheres?]	[otto von guericke, von guericke used, guericke used magdeburg, used magdeburg hemispheres?]
[velocity, affect, time?, velocity, affect, space, geometry?]	[velocity affect time?, affect time? velocity, time? velocity affect, velocity affect space, affect space geometry?]

TF-IDF

	[convert montra helicon, montra helicon d, helicon d mountain, d mountain bike, mountain bike changing, bike changing tyres?]
	[otto von guericke, von guericke used, guericke used magdeburg, used magdeburg hemispheres?]
	[velocity affect time?, affect time? velocity, time? velocity affect, velocity affect space, affect space geometry?]

TF-IDF

(20,[1,3,5,6,18],[1.638,1.639,1.638,1.64,3.26])	[convert montra helicon, montra helicon d, helicon d mountain, d mountain bike, mountain bike changing, bike changing tyres?]
(20,[0,6,10,12],[1.634,1.640,1.638,1.637])	[otto von guericke, von guericke used, guericke used magdeburg, used magdeburg hemispheres?]
(20,[8,9,14,17,19],[1.641,1.638,1.63744,1.637,1.632])	[velocity affect time?, affect time? velocity, time? velocity affect, velocity affect space, affect space geometry?]

Pipeline

```
from pyspark.ml.feature import Tokenizer

tokenizer = Tokenizer(inputCol="question_text", outputCol="question_token")


from pyspark.ml.feature import StopWordsRemover

remover = StopWordsRemover(inputCol="question_token",
                           outputCol="question_filtered")


from pyspark.ml.feature import NGram

ngram = NGram(n=3, inputCol="question_filtered", outputCol="question_3gram")


from pyspark.ml.feature import HashingTF, IDF
hashingTF = HashingTF(inputCol="question_3gram", outputCol="question_tf",
                      numFeatures=20)
idf = IDF(inputCol="question_tf", outputCol="question_tfidf")
```

Pipeline

```
from pyspark.ml import Pipeline

assembler = VectorAssembler(inputCols = [
 'question_tfidf',
 'length'], outputCol='features')

lgr = LogisticRegression(labelCol="target", featuresCol="features"
                         ,maxIter=100)

pipeline = Pipeline(stages=[tokenizer,remover,ngram,hashingTF,idf,
                            assembler,lgr])
```

Applying Estimation Models

```
paramGrid = ParamGridBuilder().build()

evaluator = MulticlassClassificationEvaluator(labelCol="target",
                                              metricName='f1')
cv = CrossValidator (estimator=pipeline,
                           estimatorParamMaps=paramGrid,
                           evaluator=evaluator,
                           numFolds=5)
train = data.filter(data['target'].isNotNull())
(trainX , validation)= train.randomSplit([0.7,0.3])
test = data.filter(data['target'].isNull())
model = cv.fit(trainX)
results = model.transform(validation).select("qid","target", "prediction")
f1 = evaluator.evaluate(results)
```

Lab 4

Data:SMSSpamCollection

Features:

- Class (1:SPAM, 0:HAM)
- Text

- 請預測新郵件是否為垃圾郵件