Spark MLlib Basics

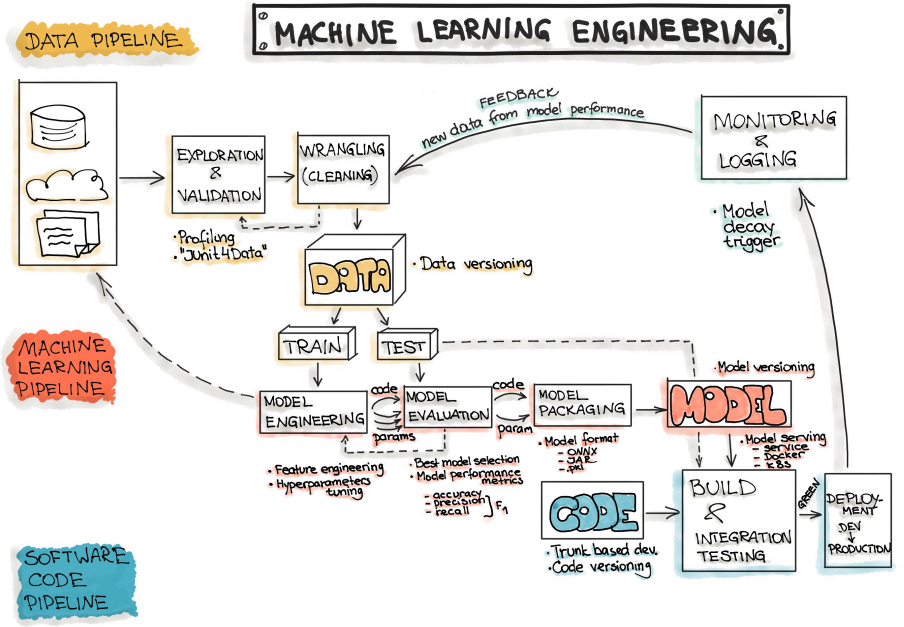
Justin Post

Big Picture

- We've studied the idea of data pipelines
- We've looked at considerations for doing (supervised) modeling and how to judge those models

Next up:

- Using Spark to do our modeling
- Understanding model pipelines
- Documenting the model building process
- Practical considerations for ML and big data
- Streaming Data



Spark Recap

Create a **Spark Session** in `pyspark`

Defines:

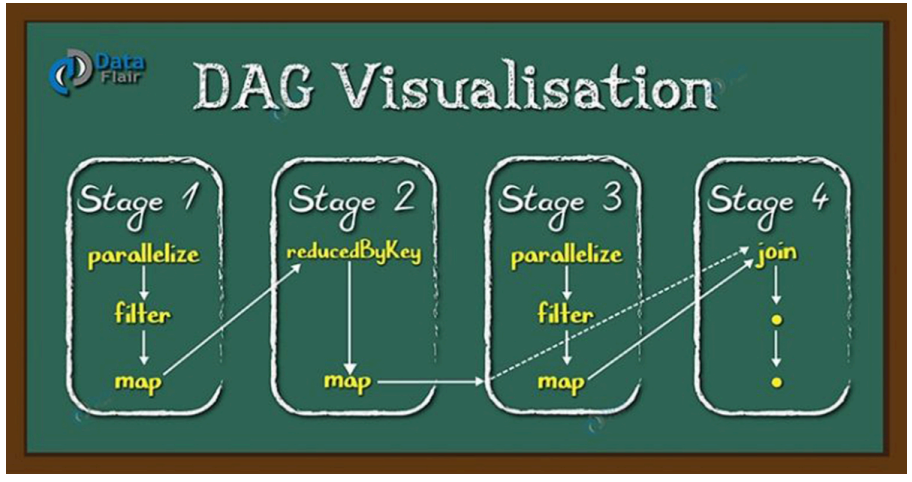
- Cluster and workers
- Spark coordinator (i.e. the **Driver**)
- Name of the app

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.master('local[*]').appName('my_app').getOrCreate()
```

Spark Recap

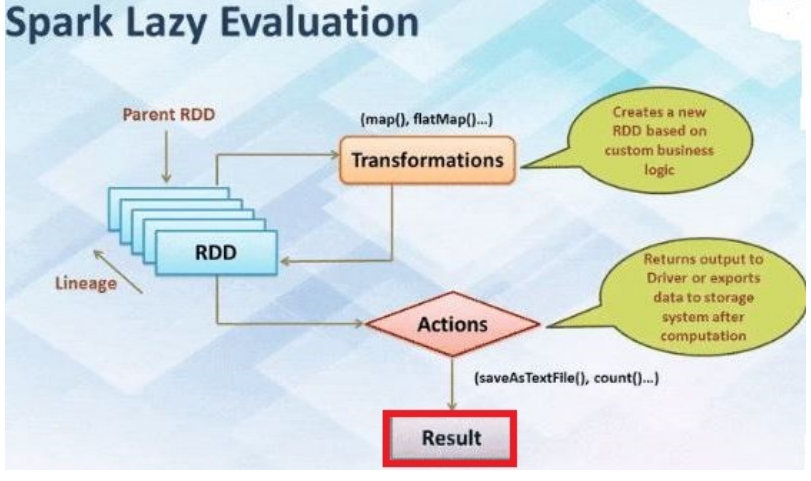
Spark handles big data and is fault tolerant!

- Turns transformations and actions into a directed acyclic graph (DAG) that allows computation to be picked back up if something fails



Spark Recap

- All transformations in Spark are *lazy*
- **Transformations** are built up and computation done only when needed
- Makes computation faster!
  - Spark can realize a dataset created through map will be used in a reduce and return only the result of the reduce rather than the larger mapped dataset



Spark Recap

Two major DataFrame APIs in `pyspark`

- pandas-on-Spark DataFrames through the `pyspark.pandas` module
- Spark SQL DataFrames through `pyspark.sql` module

Spark Recap

Two major DataFrame APIs in `pyspark`

- pandas-on-Spark DataFrames through the `pyspark.pandas` module
- Spark SQL DataFrames through `pyspark.sql` module

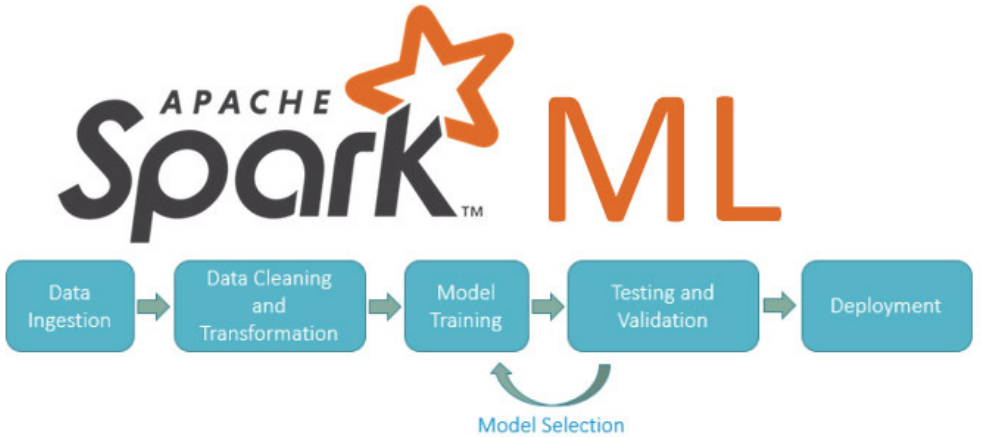
Recommended to use spark SQL for machine learning!

- Common actions to return data
  - `show(n)`, `take(n)`, `collect()`
- Common transformations done with SQL like functions

```
from pyspark.sql.functions import *
df.withColumn("Age_cat",
               when(df.Age>75, "75+")
              .when(df.Age>=70, "70-75")
              .otherwise("<70"))
```

Spark `MLlib`

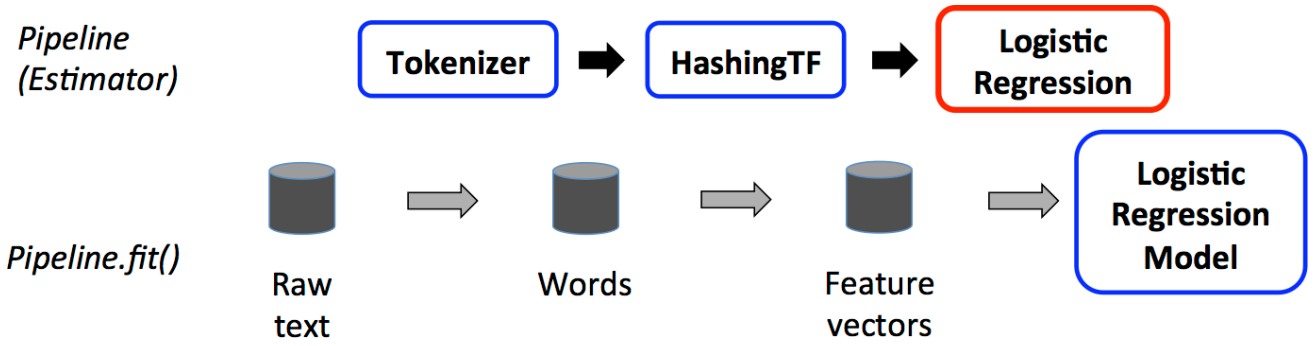
`MLlib` allows for fitting ML models in spark!



- Syntax of model fitting, CV, etc. very similar to `sklearn`!

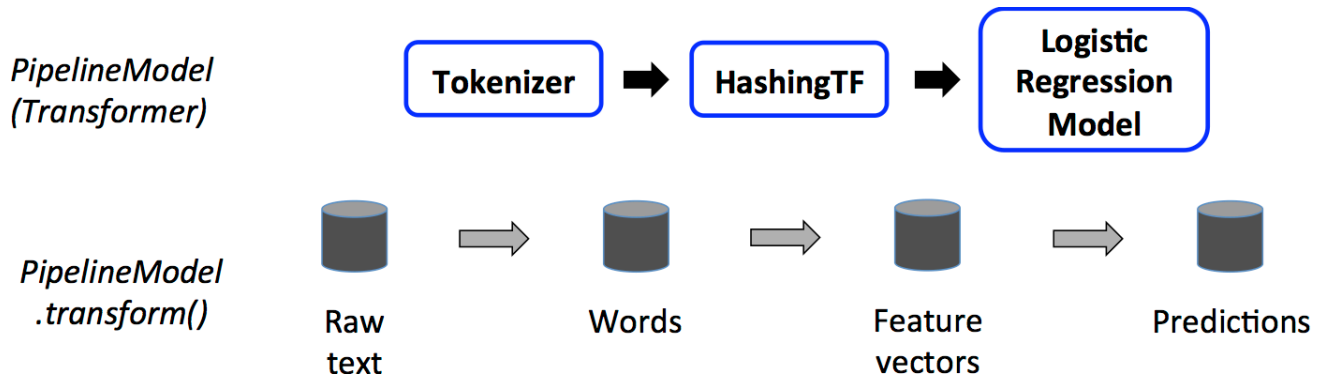
Spark MLlib

- Two major components:
  - Transformers (Create polynomials, standardize data, etc.)
  - Estimators (Models)



Spark MLlib

- Two major components:
  - Transformers (Create polynomials, standardize data, **models**, etc.)
  - Estimators (Models)



How to fit an ML Model in Spark?

- Setting up response and predictors is different:
  - Create a `label` column which represents the response
  - Create a `features` column with all of the predictors in it!

How to fit an ML Model in Spark?

- Setting up response and predictors is different:
  - Create a `label` column which represents the response
  - Create a `features` column with all of the predictors in it!
- Many functions with a `.transform()` method

```
from pyspark.ml.feature import SQLTransformer
sqlTrans = SQLTransformer(statement = "SELECT year, log(km_driven) as log_km_driven FROM __THIS__")
sqlTrans.transform(bike)
```

How to fit an ML Model in Spark?

- Setting up response and predictors is different:
  - Create a `label` column which represents the response
  - Create a `features` column with all of the predictors in it!
- Many functions with a `.transform()` method

```
from pyspark.ml.feature import SQLTransformer
sqlTrans = SQLTransformer(statement = "SELECT year, log(km_driven) as log_km_driven FROM __THIS__")
sqlTrans.transform(bike)
```

- Models and CV function have a `.fit()` method (once fitted a `.transform()` method too!)

```
from pyspark.ml.regression import LinearRegression
lr = LinearRegression(regParam = 0, elasticNetParam = 0).fit(...)
```

Jump Into Pyspark!

- Go through basic example of fitting a linear regression model in Spark `MLlib`

Recap

- Setting up response and predictors:
  - Create a `label` column which represents the response
  - Create a `features` column with all of the predictors in it!
- Many functions with a `.transform()` method
- Models and CV function have a `.fit()` method (once fitted a `.transform()` method too!)