Spark Machine Learning Pipelines

Anurag Nagar

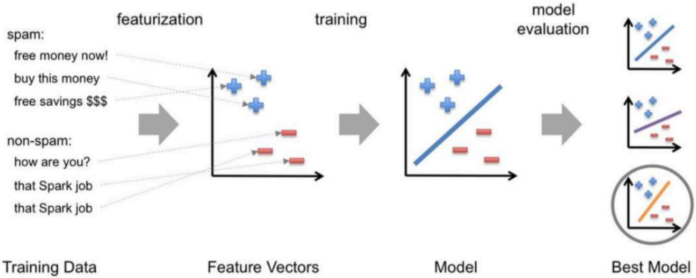
Big Data Class

Introduction

- Machine Learning projects involve multiple steps, such as pre-processing, feature extraction, model building, etc

Introduction

- Machine Learning projects involve multiple steps, such as pre-processing, feature extraction, model building, etc
- There are iterative steps that have to be done multiple times e.g. parameter optimization

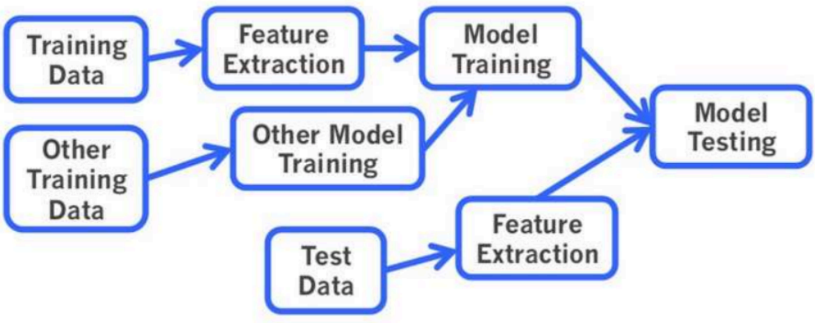


Pipelines

- Pipelines simplify the ML process by modularizing various phases.

Pipelines

- Pipelines simplify the ML process by modularizing various phases.
- Pipelines consist of a series of operations that are run sequentially.



Outline

1. Introduction

2. Pipelines
   - Pipeline Components
   - Transformers
   - Estimators
   - Pipelines
   - PipelineModel
   - Pipeline Example

3. Parameter Tuning
   - Background
   - Parameter Tuning Steps
   - Parameter Tuning Example

Pipeline Components

ML pipeline consists of the following components[1]

1. **Transformers** implements a *transform()* method, which converts one DataFrame into another, generally by appending one or more columns. For example:
   - *A Feature transformer* transforms raw data to feature vectors
   - *A Learning Model* transforms feature vector to a prediction label
2. **Estimators** abstracts the concept of a learning algorithm or any algorithm that fits or trains on data. It implements a *fit()* method which accepts a Dataframe and produces a model.

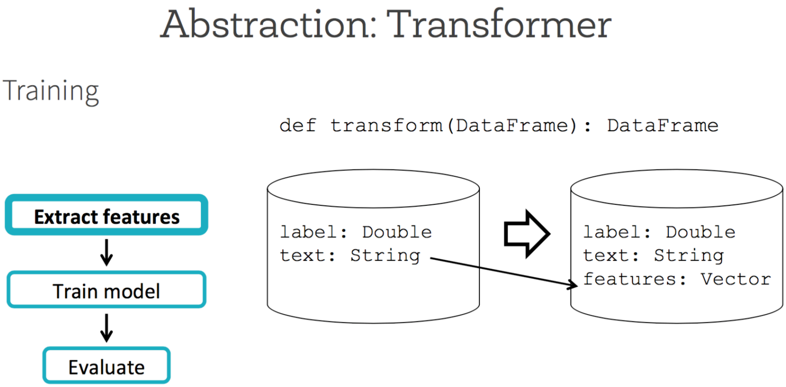
[1] see `https://spark.apache.org/docs/latest/ml-pipeline.html#pipeline-components` for more details

Outline

1. Introduction
2. Pipelines
   - Pipeline Components
   - Transformers
   - Estimators
   - Pipelines
   - PipelineModel
   - Pipeline Example
3. Parameter Tuning
   - Background
   - Parameter Tuning Steps
   - Parameter Tuning Example

Transformers

Feature transformer extracts features from raw data



Abstraction: Transformer

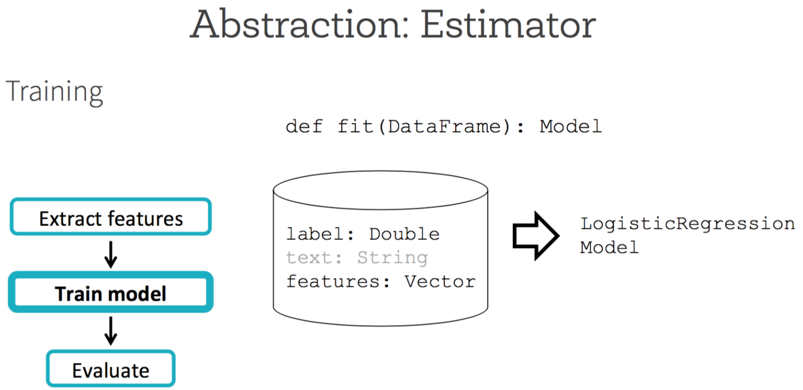
Training

```
def transform(DataFrame): DataFrame
```

Outline

Estimators

Estimators take a Dataframe with feature vectors and produce a learning model



Abstraction: Estimator

Training

```
def fit(DataFrame): Model
```

Extract features

Train model

Evaluate

```
label: Double
text: String
features: Vector
```

```
LogisticRegression
Model
```

Outline

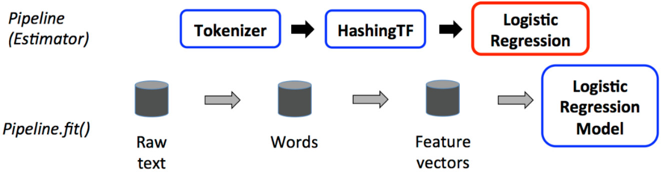
1. Introduction

2. Pipelines
   - Pipeline Components
   - Transformers
   - Estimators
   - Pipelines
   - PipelineModel
   - Pipeline Example

3. Parameter Tuning
   - Background
   - Parameter Tuning Steps
   - Parameter Tuning Example

Pipelines

Multiple stages are joined together serially to form a Pipeline.



Transformers are shown in blue and Estimators are shown in red. Overall, a Pipeline is an Estimator as it produces a model, called PipelineModel.

Outline

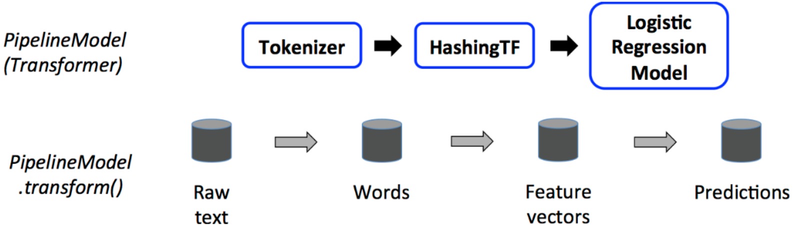
1 Introduction

2 Pipelines
- Pipeline Components
- Transformers
- Estimators
- Pipelines
- PipelineModel
- Pipeline Example

3 Parameter Tuning
- Background
- Parameter Tuning Steps
- Parameter Tuning Example

Pipelines

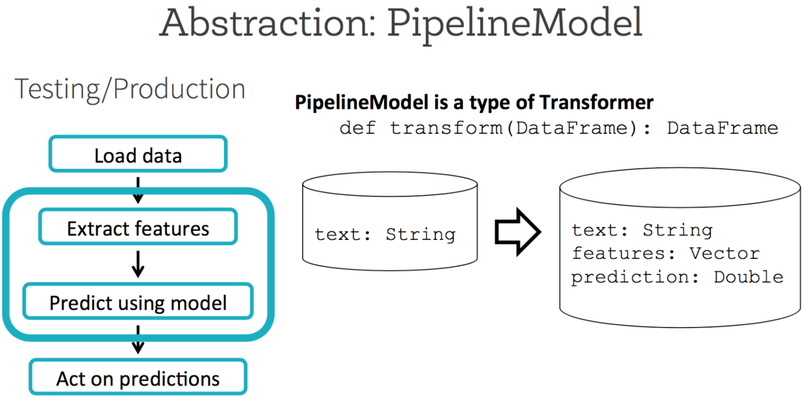
PipelineModel produced during the training phase is used for making predictions in the test phase.
Note that there are only Transformers here.



After calling *PipelineModel.transform()* on the test dataset, we obtain a DataFrame containing predictions.

Model transforms the test data

PipelineModel takes in test dataset and produces **prediction**



Abstraction: PipelineModel

Testing/Production

PipelineModel is a type of Transformer

```
def transform(DataFrame): DataFrame
```

Outline

1. Introduction
2. Pipelines
   - Pipeline Components
   - Transformers
   - Estimators
   - Pipelines
   - PipelineModel
   - Pipeline Example
3. Parameter Tuning
   - Background
   - Parameter Tuning Steps
   - Parameter Tuning Example

Pipeline Example

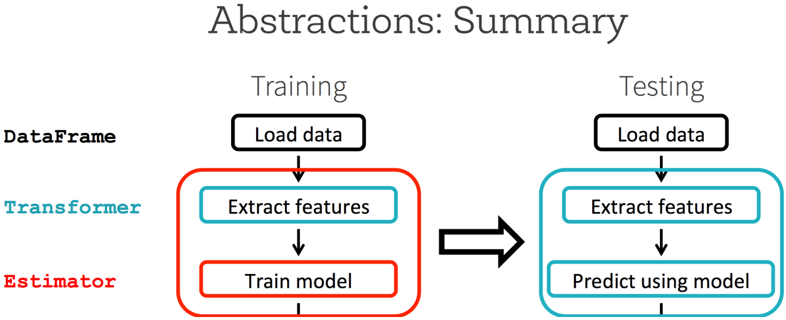
See this link:

`https://spark.apache.org/docs/latest/ml-pipeline.html#example-pipeline`

for a toy example using Pipelines.

Abstraction Summary

Summary of abstractions is shown below:



Outline

1 Introduction

2 Pipelines
- Pipeline Components
- Transformers
- Estimators
- Pipelines
- PipelineModel
- Pipeline Example

3 Parameter Tuning
- Background
- Parameter Tuning Steps
- Parameter Tuning Example

Parameter Tuning

- One of the important tasks in ML is *model selection*, which involves selecting the model with the best set of **parameters**.

Parameter Tuning

- One of the important tasks in ML is *model selection*, which involves selecting the model with the best set of **parameters**.
- This is frequently done by manually trying various combination of parameters for Estimators, such as a Logistic Regression model.

Parameter Tuning

- One of the important tasks in ML is *model selection*, which involves selecting the model with the best set of **parameters**.
- This is frequently done by manually trying various combination of parameters for Estimators, such as a Logistic Regression model.
- Spark provides an automated alternative, both for *Estimators* and for *entire pipelines*.

Parameter Tuning

- One of the important tasks in ML is *model selection*, which involves selecting the model with the best set of **parameters**.
- This is frequently done by manually trying various combination of parameters for Estimators, such as a Logistic Regression model.
- Spark provides an automated alternative, both for *Estimators* and for *entire pipelines*.
- Uses tools such as **CrossValidator** and **TrainValidationSplit** to find best choice of parameters.

Outline

1 Introduction

2 Pipelines

- Pipeline Components
- Transformers
- Estimators
- Pipelines
- PipelineModel
- Pipeline Example

3 Parameter Tuning

- Background
- Parameter Tuning Steps
- Parameter Tuning Example

Parameter Tuning Steps

1. Create a Pipeline with training stages. This should include model creation Estimator.

Parameter Tuning Steps

1. Create a Pipeline with training stages. This should include model creation Estimator.
2. Create a *parameter grid* using the **ParamGridBuilder** class. This is a grid for all values of the parameters that you want to test.

Parameter Tuning Steps

1. Create a Pipeline with training stages. This should include model creation Estimator.
2. Create a *parameter grid* using the **ParamGridBuilder** class. This is a grid for all values of the parameters that you want to test.
3. Define an evaluator, such as *BinaryClassificationEvaluator*, which will be used to evaluate the model.

Parameter Tuning Steps

1. Create a Pipeline with training stages. This should include model creation Estimator.
2. Create a *parameter grid* using the **ParamGridBuilder** class. This is a grid for all values of the parameters that you want to test.
3. Define an evaluator, such as *BinaryClassificationEvaluator*, which will be used to evaluate the model.
4. Create a **CrossValidator** object, which will split data into training and testing parts with a choice for *folds*.

Parameter Tuning Steps

1. Create a Pipeline with training stages. This should include model creation Estimator.
2. Create a *parameter grid* using the **ParamGridBuilder** class. This is a grid for all values of the parameters that you want to test.
3. Define an evaluator, such as *BinaryClassificationEvaluator*, which will be used to evaluate the model.
4. Create a **CrossValidator** object, which will split data into training and testing parts with a choice for *folds*.
5. Call the *CrossValidator.fit()* method and it will try all possible choices of parameters and give you the best choice.

Outline

1 Introduction

2 Pipelines
- Pipeline Components
- Transformers
- Estimators
- Pipelines
- PipelineModel
- Pipeline Example

3 Parameter Tuning
- Background
- Parameter Tuning Steps
- Parameter Tuning Example

Parameter Tuning Example

See this link:

`https://spark.apache.org/docs/latest/ml-tuning.html#cross-validation`

for a toy example of parameter tuning