MONASH University

Week 5

FIT5202 Big Data Processing

Transformers, Estimators and Pipeline

Week 5 Agenda

- Week 1-4 Review
- Spark for Machine Learning
- Typical ML Workflow
- Understanding Transformers and Estimators
- Pipeline API
- Tutorial Use Case
    - Adult Income Prediction ML Workflow

Week 1-4 Review

- Introduction
- VM Installation and Setup
- Python Basics
- Spark Introduction
- RDDs and DataFrames

Week 1

- SparkSession vs SparkContext
- Data Partitioning
- RDD vs DataFrame
- Searching in RDDs and DataFrames
- Spark SQL

Week 2

- Spark Join Strategies
  - Broadcast Hash Join
  - Sort Merge Join
  - Shuffled Hash Join
- Parallel Joins
  - Inner, Outer, Left, Right, Left Anti, Left Semi
- Execution plans

Week 3

- Dataframe operations
  - Sort
  - Distinct
  - Groupby
- UDFs

Week 4

Why Spark for ML?

- Unified Analytics Engine
  - Ecosystem for Data Ingestion, Feature Engineering, Model Training and deployment
- No need to downsample data to fit in a single machine
- O(n) scale-out i.e. the model linearly scales with the number of data points

Distributed Framework (Spark MLLib) vs Single Node Framework(sklearn)

A typical ML workflow

1. Feature Engineering
2. Training Models
    1. We split the data into training and test data
3. Model Validation and Selection
    1. Using evaluation metrics
4. Exporting/Deploying the model

Feature Engineering in SparkML

Feature Extractors

- TF-IDF
- Word2Vec
- CountVectorizer
- FeatureHasher

Feature Transformers

- Tokenizer
- StopWordsRemover
- n-gram
- Binarizer
- PCA
- PolynomialExpansion
- Discrete Cosine Transform (DCT)
- StringIndexer
- IndexToString

Feature Selectors

- VectorSlicer
- RFormula
- ChiSqSelector

https://spark.apache.org/docs/latest/ml-features.html

ML Pipeline

- Transformer
  - Take a dataframe as input, returns a new dataframe with one or more columns appended to it
  - To prepare data for model training
  - Include feature transformers (e.g. stringIndexer) and learned (ML) models (predicting labels)
  - .transform() method
- Estimator (Model training)
  - Learns parameters from your DataFrame to produce learned model
  - .fit() method

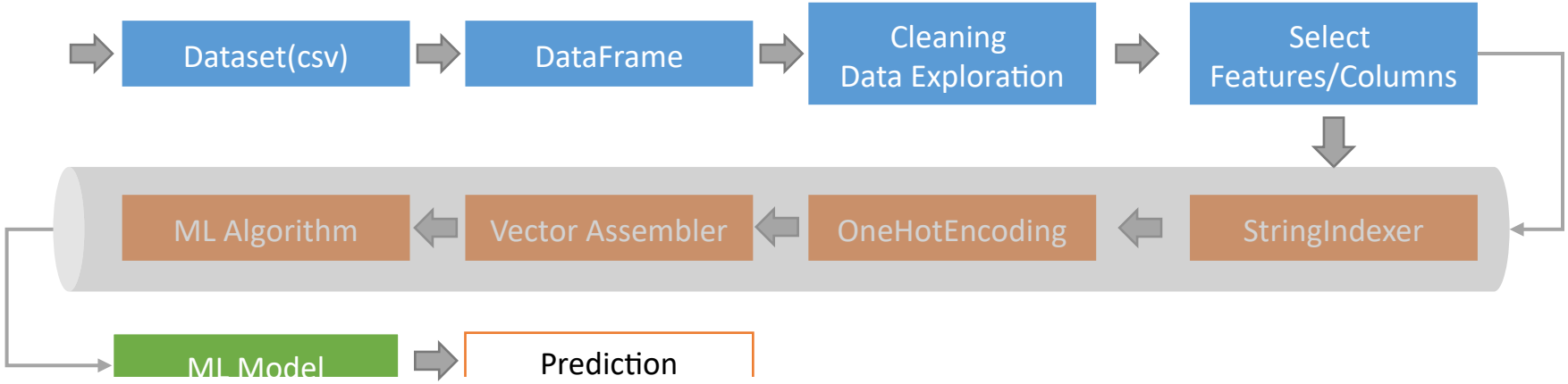
ML models

Some feature models

- Pipeline API : to organize machine learning workflow (Featurization → ML modeling)
  - Organizes a series of transformers and estimators into a single model

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

Adult Income Prediction ML Workflow

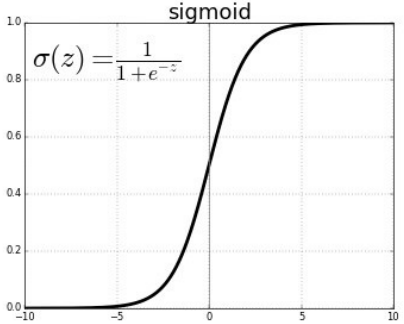


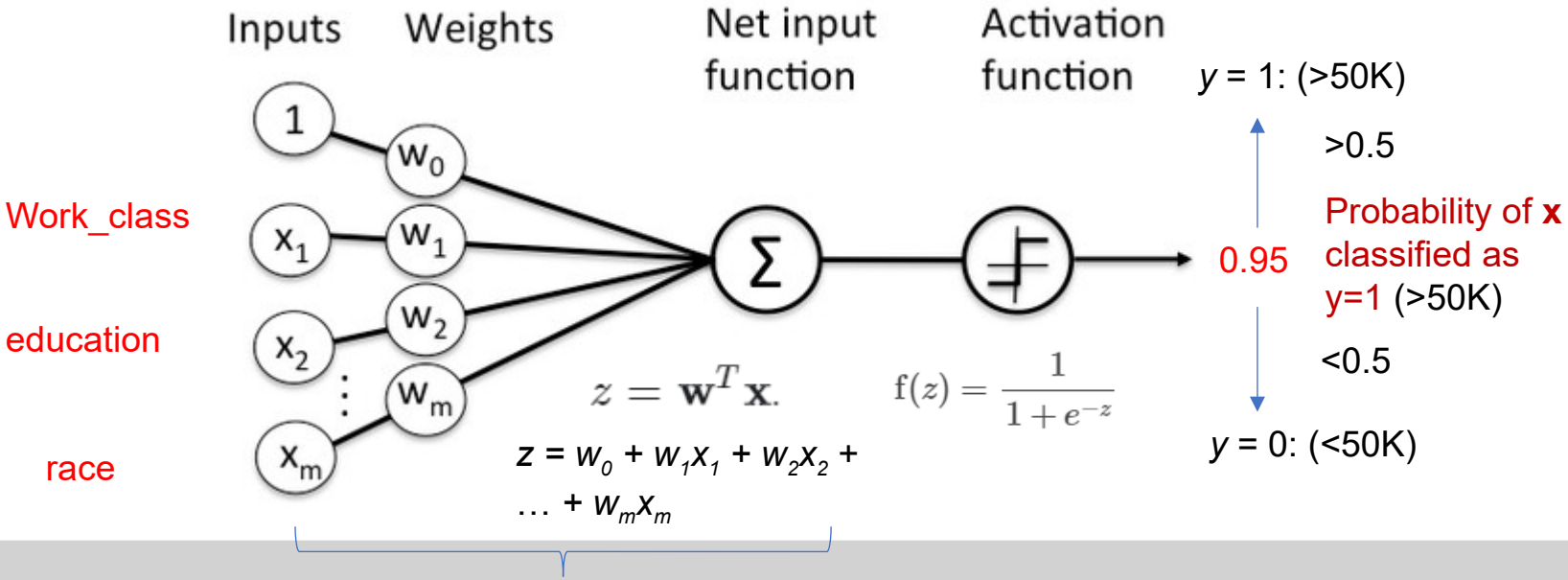
workclass_vec	education_vec	marital-statu	features	onship_vec	race_vec	gender_vec	ela
(8,[0],[1.0])	(15,[5],[1.0])	(6,[1],[	(53,[0,13,24,35,4...	[2],[1.0])	(4,[1],[1.0])	(1,[0],[1.0])	
(8,[0],[1.0])	(15,[0],[1.0])	(6,[0],[	(53,[0,8,23,39,43...	[0],[1.0])	(4,[0],[1.0])	(1,[0],[1.0])	
(8,[2],[1.0])	(15,[6],[1.0])	(6,[0],[	(53,[2,14,23,41,4...	[0],[1.0])	(4,[0],[1.0])	(1,[0],[1.0])	
(8,[0],[1.0])	(15,[1],[1.0])	(6,[0],[	(53,[0,9,23,35,43...	[0],[1.0])	(4,[1],[1.0])	(1,[0],[1.0])	
(8,[3],[1.0])	(15,[1],[1.0])	(6,[1],[	(53,[3,9,24,36,45...	[2],[1.0])	(4,[0],[1.0])	(1,[],[])	
(8,[0],[1.0])	(15,[7],[1.0])	(6,[1],[	(53,[0,15,24,34,4...	[1],[1.0])	(4,[0],[1.0])	(1,[0],[1.0])	
(8,[3],[1.0])	(15,[0],[1.0])	(6,[1],[	(53,[3,8,24,36,46...	[3],[1.0])	(4,[1],[1.0])	(1,[0],[1.0])	
(8,[1],[1.0])	(15,[9],[1.0])	(6,[0],[	(53,[1,17,23,29,4...	[0],[1.0])	(4,[0],[1.0])	(1,[0],[1.0])	
(8,[0],[1.0])	(15,[1],[1.0])	(6,[1],[	(53,[0,9,24,34,46...	[3],[1.0])	(4,[0],[1.0])	(1,[],[])	
(8,[0],[1.0])	(15,[8],[1.0])	(6,[0],[	(53,[0,16,23,30,4...	[0],[1.0])	(4,[0],[1.0])	(1,[0],[1.0])	

Binary Logistic regression

- Model for binary classification

  y = 0: (<50K)
  y = 1: (>50K)





Linear regression with multiple input variables

LogisticRegression

class pyspark.ml.classification.**LogisticRegression**(*, *featuresCol='features', labelCol='label', predictionCol='prediction', maxIter=100, regParam=0.0, elasticNetParam=0.0, tol=1e-06, fitIntercept=True, threshold=0.5, thresholds=None, probabilityCol='probability', rawPredictionCol='rawPrediction', standardization=True, weightCol=None, aggregationDepth=2, family='auto', lowerBoundsOnCoefficients=None, upperBoundsOnCoefficients=None, lowerBoundsOnIntercepts=None, upperBoundsOnIntercepts=None, maxBlockSizeInMB=0.0*) [source]

Logistic regression. This class supports multinomial logistic (softmax) and binomial logistic regression.

fit(*dataset, params=None*)

Fits a model to the input dataset with optional parameters.

New in version 1.3.0.

Parameters: **dataset** : *pyspark.sql.DataFrame*
input dataset.

params : *dict or list or tuple, optional*
an optional param map that overrides embedded params. If a list/tuple of param maps is given, this calls fit on each param map and returns a list of models.

Returns: Transformer or a list of Transformer
fitted model(s)

https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.classification.LogisticRegression.html#pyspark.ml.classification.LogisticRegression.fit

LogisticRegressionModel

class pyspark.ml.classification.**LogisticRegressionModel**(*java_model=None*) [source]

Model fitted by LogisticRegression.

New in version 1.3.0.

Methods

clear(param)	Clears a param from the param map if it has been explicitly set.
copy([extra])	Creates a copy of this instance with the same uid and some extra params.
evaluate(dataset)	Evaluates the model on a test dataset.

transform(*dataset, params=None*)

Transforms the input dataset with optional parameters.

New in version 1.3.0.

Parameters: **dataset** : *pyspark.sql.DataFrame*
input dataset

params : *dict, optional*
an optional param map that overrides embedded params.

Returns: pyspark.sql.DataFrame
transformed dataset

summary

Gets summary (accuracy/precision/recall, objective history, total iterations) of model trained on the training set. An exception is thrown if *trainingSummary is None.*

New in version 2.0.0.

https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.classification.LogisticRegressionModel.html

MONASH University

LogisticRegressionTrainingSummary

class `pyspark.ml.classification.LogisticRegressionTrainingSummary`(*java_obj: Optional[JavaObject] = None*) [source

Abstraction for multinomial Logistic Regression Training results.

New in version 2.0.0.

Methods

`fMeasureByLabel`([beta])	Returns f-measure for each label (category).
`weightedFMeasure`([beta])	Returns weighted averaged f-measure.

Attributes

`accuracy`	Returns accuracy.
`falsePositiveRateByLabel`	Returns false positive rate for each label (category).
`featuresCol`	Field in “predictions” which gives the features of each instance as a vector.
`labelCol`	Field in “predictions” which gives the true label of each instance.
`labels`	Returns the sequence of labels in ascending order.
`objectiveHistory`	Objective function (scaled loss + regularization) at each iteration.
`precisionByLabel`	Returns precision for each label (category).
`predictionCol`	Field in “predictions” which gives the prediction of each class.
`predictions`	Dataframe outputted by the model’s *transform* method.
`probabilityCol`	Field in “predictions” which gives the probability of each class as a vector.

This class provides a summary of the logistic regression model's performance and statistics after training.

https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.classification.LogisticRegressionTrainingSummary.html#pyspark.ml.classification.LogisticRegressionTrainingSummary

Thank You!

See you next week.