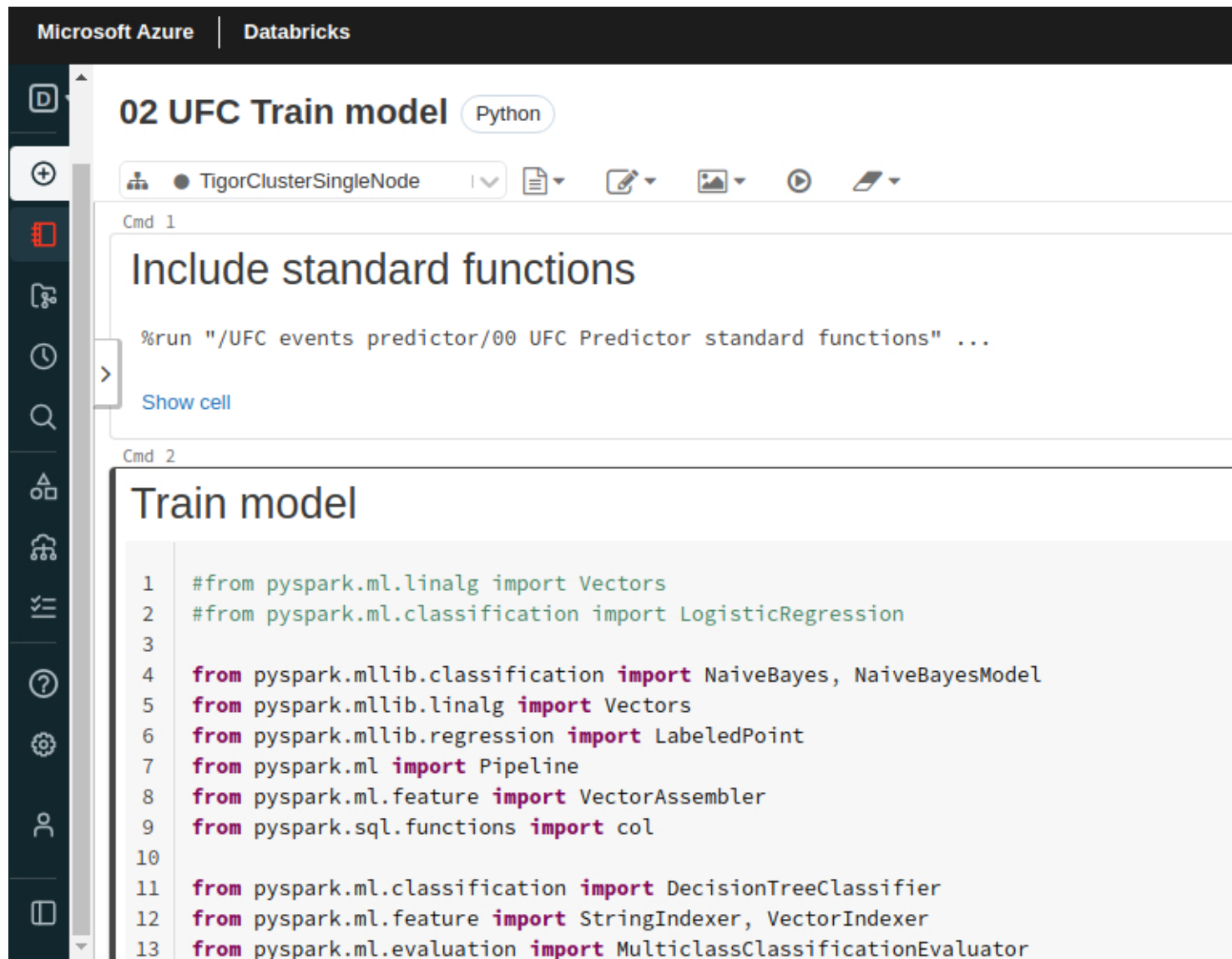


Update 202209: I am working on a system that predicts a stock price and buys and sells automatically. I

MMA Classifier is switched off because Betfair. So the stock market is the new game



Microsoft Azure | Databricks

02 UFC Train model Python

TigorClusterSingleNode

Cmd 1

Include standard functions

```
%run "/UFC events predictor/00 UFC Predictor standard functions" ...
```

Show cell

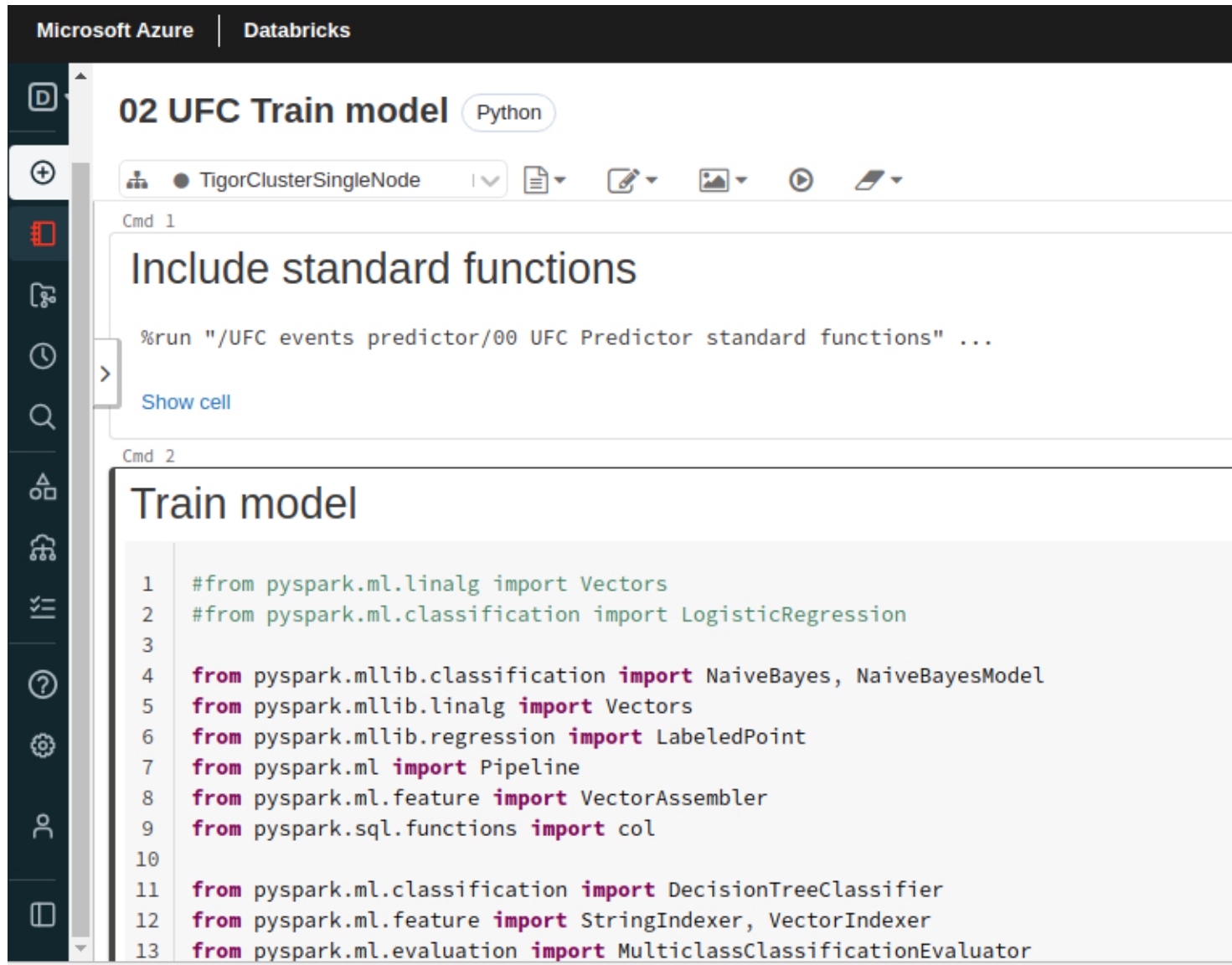
Cmd 2

Train model

```
1 #from pyspark.ml.linalg import Vectors
2 #from pyspark.ml.classification import LogisticRegression
3
4 from pyspark.mllib.classification import NaiveBayes, NaiveBayesModel
5 from pyspark.mllib.linalg import Vectors
6 from pyspark.mllib.regression import LabeledPoint
7 from pyspark.ml import Pipeline
8 from pyspark.ml.feature import VectorAssembler
9 from pyspark.sql.functions import col
10
11 from pyspark.ml.classification import DecisionTreeClassifier
12 from pyspark.ml.feature import StringIndexer, VectorIndexer
13 from pyspark.ml.evaluation import MulticlassClassificationEvaluator
```

Update 202208: After using a lot of tooling on-prem and in the Cloud I was wondering what would I actu

I have moved my MMA classifier to Databricks and Delta and the transition was smooth. It all worked in



The screenshot shows the Databricks web interface. At the top, there's a header with 'Microsoft Azure' and 'Databricks'. Below that, the notebook title is '02 UFC Train model' with a 'Python' language selector. A toolbar contains icons for cluster management, file operations, and execution. The notebook content is divided into two command cells:

Cmd 1

Include standard functions

```
%run "/UFC events predictor/00 UFC Predictor standard functions" ...
```

[Show cell](#)

Cmd 2

Train model

```
1 #from pyspark.ml.linalg import Vectors
2 #from pyspark.ml.classification import LogisticRegression
3
4 from pyspark.mllib.classification import NaiveBayes, NaiveBayesModel
5 from pyspark.mllib.linalg import Vectors
6 from pyspark.mllib.regression import LabeledPoint
7 from pyspark.ml import Pipeline
8 from pyspark.ml.feature import VectorAssembler
9 from pyspark.sql.functions import col
10
11 from pyspark.ml.classification import DecisionTreeClassifier
12 from pyspark.ml.feature import StringIndexer, VectorIndexer
13 from pyspark.ml.evaluation import MulticlassClassificationEvaluator
```

Just like so many data fans before me I have been looking at Machine Learning and how it can apply to

MMA classifier

If it looks like a duck, swims like a duck, and quacks like a d
probably *is* a duck.

Duck test

From Wikipedia, the free encyclopedia

upgrading and server-side data management. This approach is chosen for its ability to handle large volumes of data and its support for various data formats and protocols.

_Machine Learning

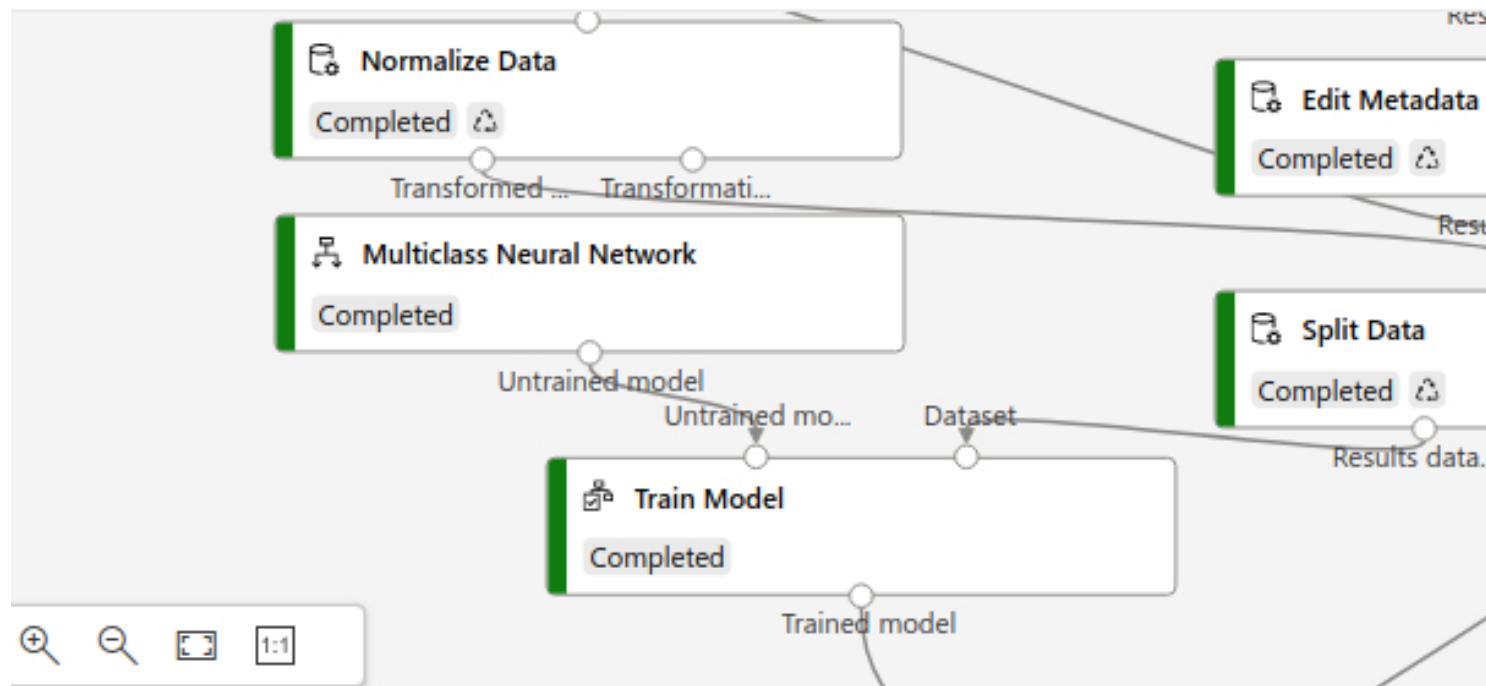
Written by Administrator

Sunday, 10 June 2018 17:54 - Last Updated Sunday, 01 September 2024 09:56

Publish ▶ Resubmit ⊗ Cancel 🗑 Delete

⊕ logs Metrics Images Snapshot Explanations (preview) Fairness (preview)

Canvas



My experience with Machine Learning also includes a project I did at Dutch Railways where I have used

```
.setHandleInvalid( keep );

OneHotEncoder berekendeFractieEncoder = new OneHotEncoder()
    .setInputCols(new String[]{"TreinserieI", "TreinseriegroepI", "MeetmaandI", "TijdstipcombinatieI"})
    .setOutputCols(new String[]{"TreinserieV", "TreinseriegroepV", "MeetmaandV", "TijdstipcombinatieV"})
    .setDropLast(true);

VectorAssembler berekendeFractieVectorAssemblerTreinserie = new VectorAssembler()
    .setInputCols(new String[]{"TreinserieV", "MeetmaandV", "TijdstipcombinatieV"})
    .setOutputCol("featuresTreinserie");

VectorAssembler berekendeFractieVectorAssemblerTreinseriegroep = new VectorAssembler()
    .setInputCols(new String[]{"TreinseriegroepV", "MeetmaandV", "TijdstipcombinatieV"})
    .setOutputCol("featuresTreinseriegroep");

VectorAssembler berekendeFractieVectorAssemblerLandelijk = new VectorAssembler()
    .setInputCols(new String[]{"MeetmaandV", "TijdstipcombinatieV"})
    .setOutputCol("featuresLandelijk");

/* df2 veranderen in df1 om terug te gaan */
Dataset<Row> dsTraining = df3.filter("Datasettype = 0");
Dataset<Row> dsPredict = df3.filter("Datasettype = 1");

LinearRegression linRegTreinserie = new LinearRegression();
LinearRegression linRegTreinseriegroep = new LinearRegression();
LinearRegression linRegLandelijk = new LinearRegression();

linRegTreinserie.setLabelCol("Waargenomenfractie"); //.setMaxIter(100);
linRegTreinserie.setFeaturesCol("featuresTreinserie");
linRegTreinserie.setPredictionCol("BerekendefractieTreinserie");
linRegTreinserie = setLinearRegressionParameters(linRegTreinserie);

linRegTreinseriegroep.setLabelCol("Waargenomenfractie"); //.setMaxIter(100);
linRegTreinseriegroep.setFeaturesCol("featuresTreinseriegroep");
linRegTreinseriegroep.setPredictionCol("BerekendefractieTreinseriegroep");
linRegTreinseriegroep = setLinearRegressionParameters(linRegTreinseriegroep);
```