1_numeric_features

November 24, 2019

1 Screencast Code

The follow code is the same used in the "Numeric Features" screencast. Run each code cell to see how

```
In [1]: from pyspark.sql import SparkSession
        from pyspark.ml.feature import RegexTokenizer, VectorAssembler, Normalizer, StandardScal
        from pyspark.sql.functions import udf
        from pyspark.sql.types import IntegerType

        import re
```

```
In [2]: # create a SparkSession: note this step was left out of the screencast
        spark = SparkSession.builder \
            .master("local") \
            .appName("Word Count") \
            .getOrCreate()
```

2 Read in the Data Set

```
In [3]: stack_overflow_data = 'Train_onetag_small.json'
```

```
In [4]: df = spark.read.json(stack_overflow_data)
```

```
In [5]: df.head()
```

```
Out[5]: Row(Body="<p>I'd like to check if an uploaded file is an image file (e.g png, jpg, jpeg,
```

3 Tokenization

Tokenization splits strings into separate words. Spark has a Tokenizer class as well as RegexTokenizer, which allows for more control over the tokenization process.

```
In [6]: # split the body text into separate words

        regexTokenizer = RegexTokenizer(inputCol="Body", outputCol="words", pattern="\\W")
        df = regexTokenizer.transform(df)
        df.head()
```

```
Out[6]: Row(Body="<p>I'd like to check if an uploaded file is an image file (e.g png, jpg, jpeg,

In [7]: # count the number of words in each body tag

        body_length = udf(lambda x: len(x), IntegerType())
        df = df.withColumn("BodyLength", body_length(df.words))

In [8]: # count the number of paragraphs and links in each body tag

        number_of_paragraphs = udf(lambda x: len(re.findall("</p>", x)), IntegerType())
        number_of_links = udf(lambda x: len(re.findall("</a>", x)), IntegerType())

In [9]: df = df.withColumn("NumParagraphs", number_of_paragraphs(df.Body))
        df = df.withColumn("NumLinks", number_of_links(df.Body))

In [10]: df.head(2)

Out[10]: [Row(Body="<p>I'd like to check if an uploaded file is an image file (e.g png, jpg, jpe
          Row(Body='<p>In my favorite editor (vim), I regularly use ctrl-w to execute a certain
```

4 VectorAssembler

Combine the body length, number of paragraphs, and number of links columns into a vector

```
In [11]: assembler = VectorAssembler(inputCols=["BodyLength", "NumParagraphs", "NumLinks"], outp
         df = assembler.transform(df)

In [12]: df.head()

Out[12]: Row(Body="<p>I'd like to check if an uploaded file is an image file (e.g png, jpg, jpeg
```

5 Normalize the Vectors

```
In [13]: scaler = Normalizer(inputCol="NumFeatures", outputCol="ScaledNumFeatures")
         df = scaler.transform(df)

In [14]: df.head(2)

Out[14]: [Row(Body="<p>I'd like to check if an uploaded file is an image file (e.g png, jpg, jpe
          Row(Body='<p>In my favorite editor (vim), I regularly use ctrl-w to execute a certain
```

6 Scale the Vectors

```
In [15]: scaler2 = StandardScaler(inputCol="NumFeatures", outputCol="ScaledNumFeatures2", withSt
         scalerModel = scaler2.fit(df)
         df = scalerModel.transform(df)

In [16]: df.head(2)

Out[16]: [Row(Body="<p>I'd like to check if an uploaded file is an image file (e.g png, jpg, jpe
          Row(Body='<p>In my favorite editor (vim), I regularly use ctrl-w to execute a certain

In [ ]:
```