Lucas Beane
Due 1/30/19
Professor Alonzi
DS 6003

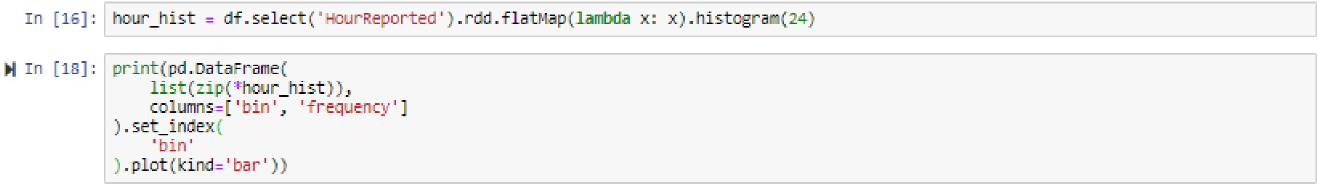
Full Pipeline using Spark

Motivation:

I was first motivated to tackle crime data in Charlottesville because it’s one of the data sets on the Open Data Portal, which is the subject of my capstone. I thought it would be good to get practice working with some real life data that could be applied to problems I’m working on. Some possible problems I considered included predicting type of offense based on location and time of the crime, as well as seeing if time of the year influenced that at all. Unfortunately, working on a multiclass logistic regression with pySpark turned out to be too much with my current toolkit. Instead, I focused on producing results from a nonsense problem of predicting block number from the time of day a crime was perpetrated. The good news is that with an R^2 value of 0.0001, I’ve all but verified that these two features have no correlation. In the future, I hope to learn more of the ins and outs of pySpark in order to better implement such models, particularly in the case of multiclass logistic regression.

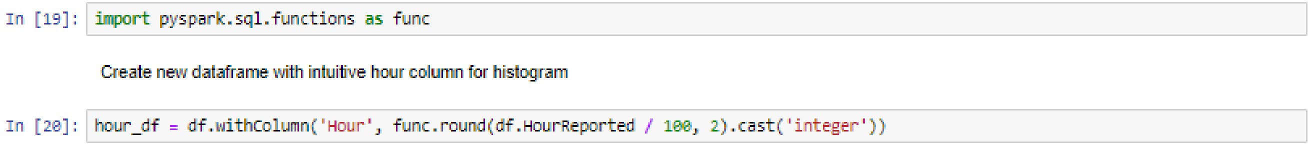
Code Snippet:

How I created a histogram of crimes over time of day:



```
In [16]: hour_hist = df.select('HourReported').rdd.flatMap(lambda x: x).histogram(24)

In [18]: print(pd.DataFrame(
             list(zip(*hour_hist)),
             columns=['bin', 'frequency']
         ).set_index(
             'bin'
         ).plot(kind='bar'))
```

Using other functions to prettify plot:

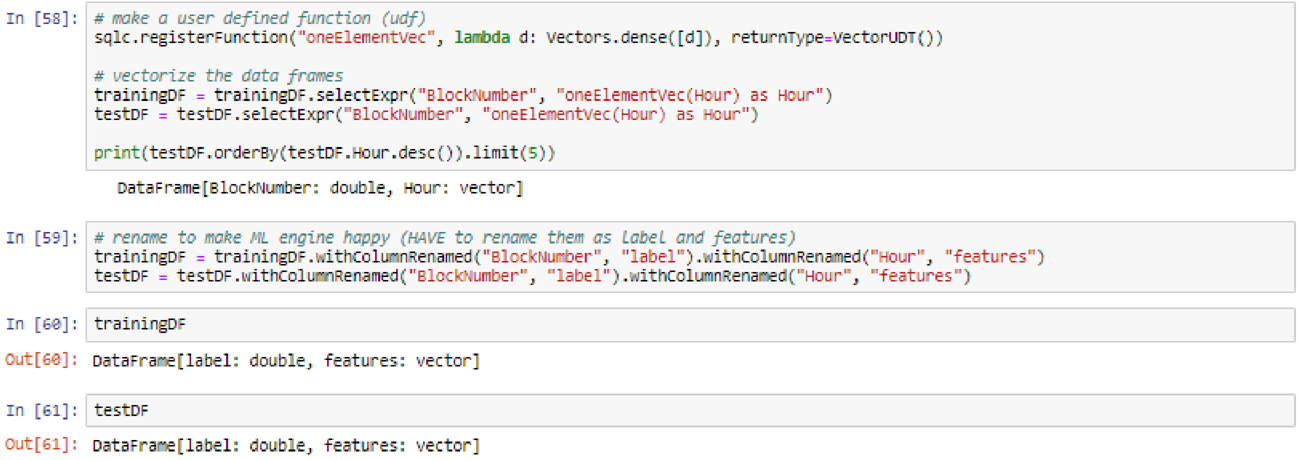


```
In [19]: import pyspark.sql.functions as func
```

Create new dataframe with intuitive hour column for histogram

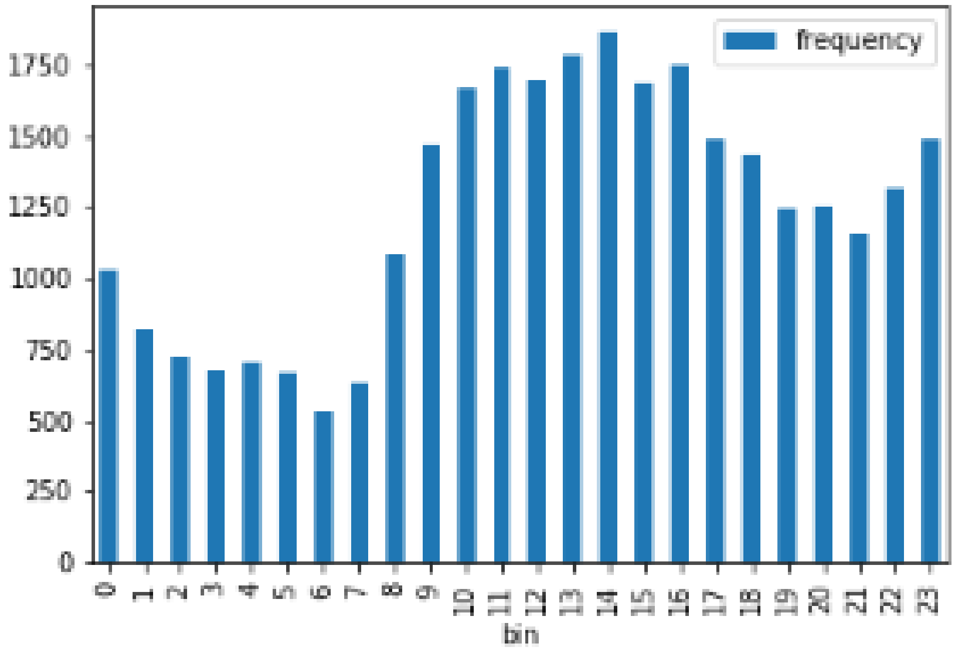
```
In [20]: hour_df = df.withColumn('Hour', func.round(df.HourReported / 100, 2).cast('integer'))
```

Get training and testing sets into correct formats and check for consistency:



```
In [58]: # make a user defined function (udf)
         sqlc.registerFunction("oneElementVec", lambda d: Vectors.dense([d]), returnType=VectorUDT())

         # vectorize the data frames
         trainingDF = trainingDF.selectExpr("BlockNumber", "oneElementVec(Hour) as Hour")
         testDF = testDF.selectExpr("BlockNumber", "oneElementVec(Hour) as Hour")

         print(testDF.orderBy(testDF.Hour.desc()).limit(5))

            DataFrame[BlockNumber: double, Hour: vector]

In [59]: # rename to make ML engine happy (HAVE to rename them as label and features)
         trainingDF = trainingDF.withColumnRenamed("BlockNumber", "label").withColumnRenamed("Hour", "features")
         testDF = testDF.withColumnRenamed("BlockNumber", "label").withColumnRenamed("Hour", "features")

In [60]: trainingDF

Out[60]: DataFrame[label: double, features: vector]

In [61]: testDF

Out[61]: DataFrame[label: double, features: vector]
```

Visualization:



This is a plot of the distribution of crimes in Charlottesville over time of day developed from the pySpark dataframe I created. It shows an odd drop at around midnight, and a spike at around 8 a.m. Though nothing too surprising is shown, it's still an interesting graph to consider.