Correlation

November 30, 2017

```
In [1]: from pyspark.sql.types import StructType, StructField, FloatType, LongType, StringType
        from pyspark.shell import spark

        feats = []
        f = open('features.txt')
        for line_num, line in enumerate(f):
            if line_num == 0:
                # Timestamp
                feats.append(StructField(line.strip(), LongType(), True))
            elif line_num == 1:
                # Geohash
                feats.append(StructField(line.strip(), StringType(), True))
            else:
                # Other features
                feats.append(StructField(line.strip(), FloatType(), True))

        schema = StructType(feats)
```

```
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.2.0
      /_/

Using Python version 3.6.3 (default, Oct  6 2017 12:04:38)
SparkSession available as 'spark'.
```

```
In [2]: df = spark.read.format('csv').option('sep', '\t').schema(schema).load('inputs/mini-samp
```

```
In [3]: col_names = []
        for i in range(2,len(df.columns)):
            col_names.append(df.columns[i])
```

```
In [4]: from pyspark.ml.stat import Correlation
        from pyspark.ml.feature import VectorAssembler
        vectorAssembler = VectorAssembler(inputCols=col_names,
                                          outputCol="features")
        trans_features = vectorAssembler.transform(df)
        coeff = Correlation.corr(trans_features,'features',method='pearson').collect()[0][0]

In [5]: mtrx = coeff.toArray()

In [6]: import numpy as np
        np.savetxt('./heatmap-generation/correlation_matrix.txt', mtrx)

In [10]: #2.5 min on mini sample data

In [13]: list_corr_pairs_coeffs = []
         feature_pairs = []

         for i in range(0,56):
             for j in range(0,56):

                 if (i != j) and not(((col_names[i]+"_"+col_names[j]) in feature_pairs) or
                                  ((col_names[j]+"_"+col_names[i]) in feature_pairs)):
                     feature_pairs.append(col_names[i] +"_"+col_names[j])
                     corr_pair_coeff = []
                     corr_pair_coeff.append(col_names[i]+" , "+col_names[j])
                     corr_pair_coeff.append(col_names[i])
                     corr_pair_coeff.append(col_names[j])
                     corr_pair_coeff.append(float(mtrx[i][j]))
                     list_corr_pairs_coeffs.append(tuple(corr_pair_coeff))

In [14]: df_corr_coeff_col_names = []
         df_corr_coeff_col_names.append(StructField("Feature_Pair", StringType(), True))
         df_corr_coeff_col_names.append(StructField("Feature1", StringType(), True))
         df_corr_coeff_col_names.append(StructField("Feature2", StringType(), True))
         df_corr_coeff_col_names.append(StructField("Pearson_Coeff", FloatType(), True))
         df_corr_coeff = spark.createDataFrame(list_corr_pairs_coeffs,StructType(df_corr_coeff
         sort_coeff_df = df_corr_coeff.sort(df_corr_coeff.Pearson_Coeff.desc())

In [15]: f = open("Feature_pair_sorted_coeff.txt","w")
         sort_coeff_df_list = sort_coeff_df.collect()
         for i in range(0,len(sort_coeff_df_list)):
             f.write(sort_coeff_df_list[i].Feature_Pair + " " + (str)(sort_coeff_df_list[i].Pea
             f.write("\n")
         f.close()
```