**M5 Lab: Computing Pairwise Correlations**

### **Part I: Simple Correlation**

1. Login to your Databricks Community Edition account.
2. Create a new cluster.
3. Create a new notebook. Choose Python for programming. You can also choose Scala or Java if you prefer.
4. Go to the Spark ML Documentation at this page: [https://spark.apache.org/docs/2.3.0/mllib-statistics.html#correlations (Links to an external site.)](https://spark.apache.org/docs/2.3.0/mllib-statistics.html#correlations)
5. Go to the **Correlations** section, copy the code there into the notebook that you have created. Make sure to copy the code written in the same programming language as the one you specified for the notebook

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

1. Run the code. Explain, what this code is doing. What are the obtained results?

Code is computing the correlation using Pearson’s method and ‘spearman’ for Spearman’s method. The obtained result is correlation matrix using person’s method and use spearmen’s for spearman’s method. If the methods are not specified, then Pearson method is used by default.

1. Change the correlation method from “Pearson” to “spearman”. Rerun your code. Do you notice any differences in the obtained results? Research the difference between both correlation methods.

Yes, noticed the differences,

Pearson Correlation:

The Pearson correlation evaluates the linear relationship between two continuous variables. A relationship is linear when a change in one variable is associated with a proportional change in the other variable.

For example, you might use a Pearson correlation to evaluate whether increases in temperature at your production facility are associated with decreasing thickness of your chocolate coating.

Spearman rank-order correlation

The Spearman correlation evaluates the monotonic relationship between two continuous or ordinal variables. In a monotonic relationship, the variables tend to change together, but not necessarily at a constant rate. The Spearman correlation coefficient is based on the ranked values for each variable rather than the raw data.

For example, you might use a Spearman correlation to evaluate whether the order in which employees complete a test exercise is related to the number of months they have been employed.

### **Part II: Pairwise Correlations**

In this part you will load a small dataset file into Databricks and compute the pairwise correlations. That is, the correlation between every pair of attributes.

Uploading the Data File users\_no\_labels

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Creating new table : Users\_no\_labels

Graphical user interface, text, application

Description automatically generated

Create table with UI

Graphical user interface, application

Description automatically generated

Table view

Graphical user interface, application

Description automatically generated

Reading the Data from the File.

Graphical user interface, text, application, email

Description automatically generated

Option 1:

  Read the data from the table instead of the file.

Graphical user interface, text, application

Description automatically generated

 Convert the dataframe into an RDD data structure. That's because the correlation function takes its input as an RDD and not a dataframe.

Graphical user interface, text, application

Description automatically generated

call the correlation function and print the results

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Option 2:

Graphical user interface, text, application, email

Description automatically generated

### **Part III: Working with a Big Dataset**

Both Option1 & Option 2 is done ,For clear visibility of Data Frame created table through notebook.

Graphical user interface, application, table

Description automatically generated

Table

Description automatically generated

A picture containing graphical user interface

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

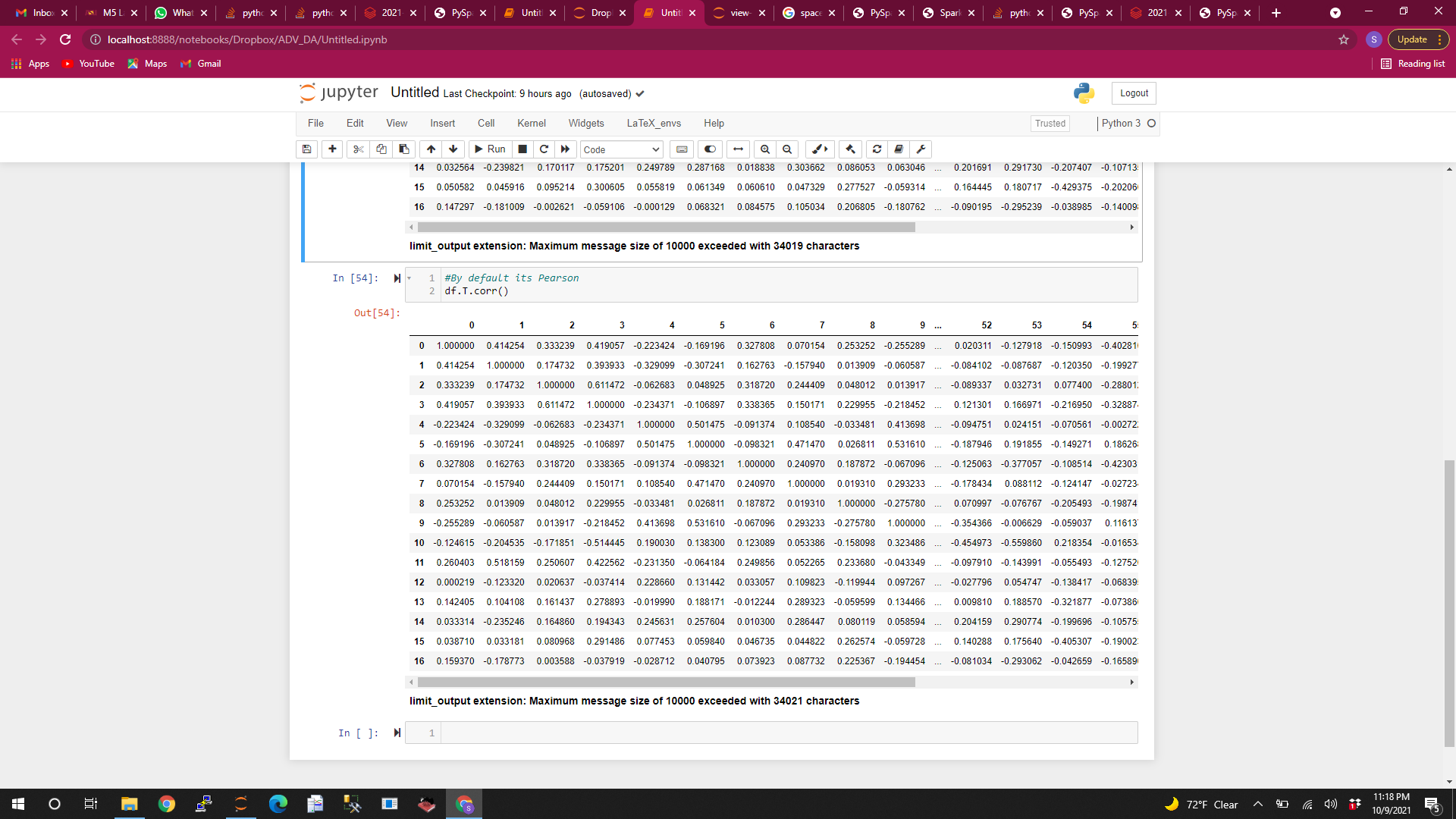
Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

A screenshot of a computer

Description automatically generated



I have created these table in Notebook as the file is easy to work with , because it’s a big dataset file.

How many data points does this dataset contain?

2000 data points.

How many attributes does each data point have?

62 attributes.

Graphical user interface, text, application, email

Description automatically generated

How many pairwise similarities should be computed?

62\*62 = 3844 (Because I have transposed from rows to columns)

Without Transforming (2000, 62) The Results are as follows:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

How many data points does this dataset contain?

62 data points.

How many attributes does each data point have?

2000 attributes.

How many pairwise similarities should be computed?

2000\*2000