

DATA ANALYSIS - AUTISM SPECTRUM DISORDER

PROJECT REPORT

**SUBMITTED TO
Ms. SHALINI KUMARI**

**SUBMITTED BY PRIYANKA K
BATCH NO. 4868
Bengaluru**

Table Of Contents

1. Abstraction

2. Introduction

3. Analysis

3.1 Statistical Analysis

3.2 Data Visualization

4. Modules

4.1 NumPy

4.2 Pandas

4.3 Matplotlib

4.4 Seaborn

5. Technologies Used

5.1 Jupyter Notebook

5.2 Anaconda

6. Results and Screenshots

7. Conclusion

Chapter 1

Abstraction

Autism spectrum disorder (ASD) is an early developmental disorder characterized by mutation of enculturation associated with attention deficit disorder in the visual perception of emotional expressions. An estimated one in more than 100 people has autism. Autism affects almost four times as many boys than girls. Data analysis and classification of ASD is still challenging due to unsolved issues arising from many severity levels and range of signs and symptoms. To understanding the functions which involved in autism, neuroscience technology analyzed responses to stimuli of autistic audio and video. The study focuses on analyzing the data set of children with ASD using practical component analysis method. To satisfy this aim, the proposed method consists of two main stages including: (1) data set preparation, (2) Data analysis.

Chapter 2

Introduction

Autism spectrum disorder (ASD) is a condition that can be characterized by a constant deficit in social communication, social interaction, and the presence of restrictive and repetitive behavior. It is an early developmental disorder characterized by alterations in socialization associated with a deficit in the visual perception of faces and emotional expressions. This deficit in the perception of faces and emotional expressions seems to be linked to the peculiarities of the gaze in autistic pathology. The study of this behavioral disorder is carried out by the measurement of different ocular parameters during the perception of neutral and emotional faces (expressing joy or sadness).

Some symptoms in ASD typically appear after 2 years of age therefore, the early diagnosis could be a better opportunity to get treatment and healing. It is generally recognized that traditional clinical methods have difficulty in well distinguishing patients from healthy controls. Therefore, data analysis and classification of ASD is still challenging due to unsolved issues arising from many severity levels and range of signs and symptoms. Thus a large percentage of the population is diagnosis after developmental windows in which behavioral therapy would have had maximal impact on future development and quality of life.

The average age of diagnosis in the United States is 5.7 years and an estimated 27% remain undiagnosed at 8 years of age. At these late stages in development, many of the opportunities to intervene with therapy have evaporated.

The purpose of this special report is to summarize the latest understanding of autism's commonly associated physical and mental health conditions, including how best to identify, treat and in some cases prevent them to improve overall health and quality of life.

Chapter 3

Analysis

3.1 Statistical Analysis

Statistical analysis is the collection and interpretation of data in order to uncover patterns and trends. It is a component of data analytics. Statistical analysis can be used in situations like gathering research interpretations, statistical modeling or designing surveys and studies. It can also be useful for business intelligence organizations that have to work with large data volumes.

In the context of business intelligence (BI), statistical analysis involves collecting and scrutinizing every data sample in a set of items from which samples can be drawn. A sample, in statistics, is a representative selection drawn from a total population.

The goal of statistical analysis is to identify trends. A retail business, for example, might use statistical analysis to find patterns in unstructured and semi-structured customer data that can be used to create a more positive customer experience and increase sales.

3.2Data visualization

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

In the world of Big Data, data visualization tools and technologies are essential to analyze massive amounts of information and make data-driven decisions.

To communicate information clearly and efficiently, data visualization uses statistical graphics, plots, information graphics and other tools. Numerical data may be encoded using dots, lines, or bars, to visually communicate a quantitative message. Effective visualization helps users analyze and reason about data and evidence. It makes complex data more accessible, understandable and usable. Users may have particular analytical tasks, such as making comparisons or understanding causality, and the design principle of the graphic (i.e., showing comparisons or showing causality) follows the task. Tables are generally used where users will look up a specific measurement, while charts of various types are used to show patterns or relationships in the data for one or more variables.

Chapter 4

Modules

4.1 NumPy

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python. NumPy is a Python library and is written partially in Python, but most of the parts that require fast computation are written in C or C++.

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. It also discusses the various array functions, types of indexing, etc.

4.2 Pandas

Pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language. It is already well on its way toward this goal.

Pandas are well suited for many different kinds of data:

- Tabular data with heterogeneously-typed columns, as in an SQL table or Excel

spreadsheet

- ☐ Ordered and unordered (not necessarily fixed-frequency) time series data.
- ☐ Arbitrary matrix data (homogeneously typed or heterogeneous) with row and column

labels

- ☐ Any other form of observational / statistical data sets. The data actually need not be

labeled at all to be placed into a pandas data structure

4.3 Matplotlib

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc. scikit-learn is a library, i.e. a collection of classes and functions that users import into Python programs. Using scikit-learn therefore requires basic Python programming knowledge. No command-line interface, let alone a graphical user interface, is offered for non-programmer users Scikit-learn is the most useful library for machine learning in Python. It is on NumPy, SciPy and matplotlib, this library contains a lot of efficient tools for machine learning and statistical

modeling including classification, regression, clustering and dimensionality reduction. Scikit-learn provide a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

4.4 Seaborn

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

This also has various feature helps us to visualize data in the form of drawings and graphs. It is based on matplotlib, which is a comprehensive library that helps us in creating static, dynamic and interactive visualizations of data in the python.

It also provides us with a high-level interface and setup for creating drawings and making them more attractive. With the help of this, we can put in more information into the graphs.

As we know, this library uses Matplotlib in order to generate the data in a very attractive and innovative way which makes it easier to understand. So it will also have a role in visualizing random distributions.

Chapter 5

Technologies Used

5.1 Jupyter Notebook

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebooks are a spin-off project from the Python project, which used to have an Python Notebook project itself.

Project Jupyter is a nonprofit organization created to "develop open-source software, open-standards, and services for interactive computing across dozens of programming languages".

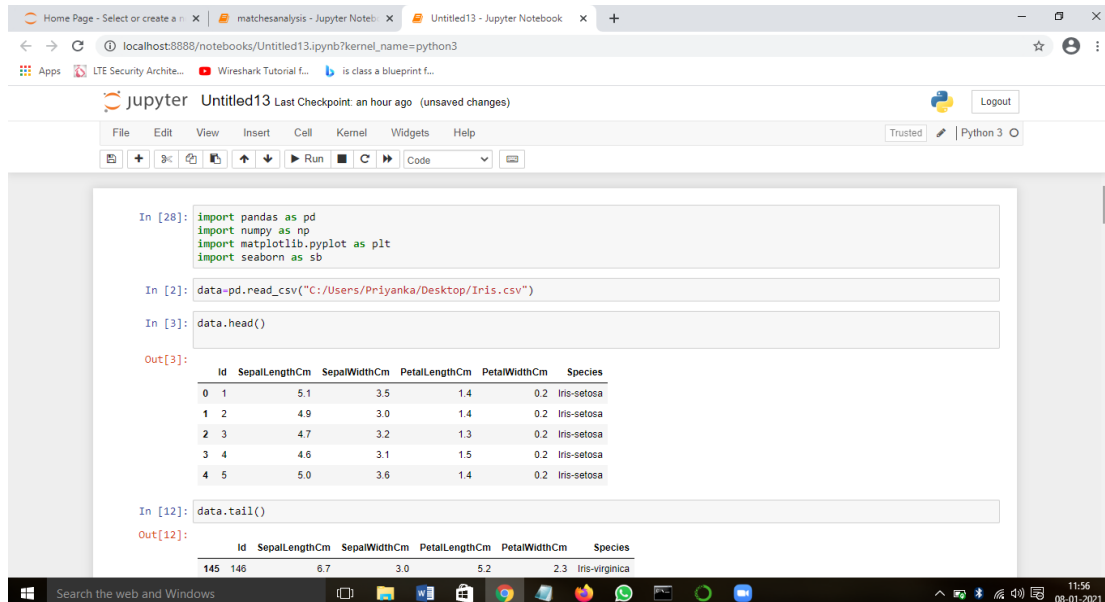
Spun-off from IPython in 2014 by Fernando Pérez, Project Jupyter supports execution environments in several dozen languages. The Jupyter Notebook is an incredibly powerful tool for interactively developing and presenting data science projects.

5.2 Anaconda

Anaconda (Python distribution) is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Access and manage the most powerful data science and machine learning libraries, packages, and tools the open-source community has to offer. Empower the data scientists to deploy models and scale their operations with ease. Secure, govern, and monitor the open-source machine learning pipeline.

Chapter 6

Results and Screenshots



A screenshot of a Jupyter Notebook interface. The browser tabs show 'Home Page - Select or create a...', 'matchesanalysis - Jupyter Note...', and 'Untitled13 - Jupyter Notebook'. The address bar shows 'localhost:8888/notebooks/Untitled13.ipynb?kernel_name=python3'. The Jupyter interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution. The notebook content shows the following code and output:

```
In [28]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb

In [2]: data=pd.read_csv("C:/Users/Priyanka/Desktop/Iris.csv")

In [3]: data.head()

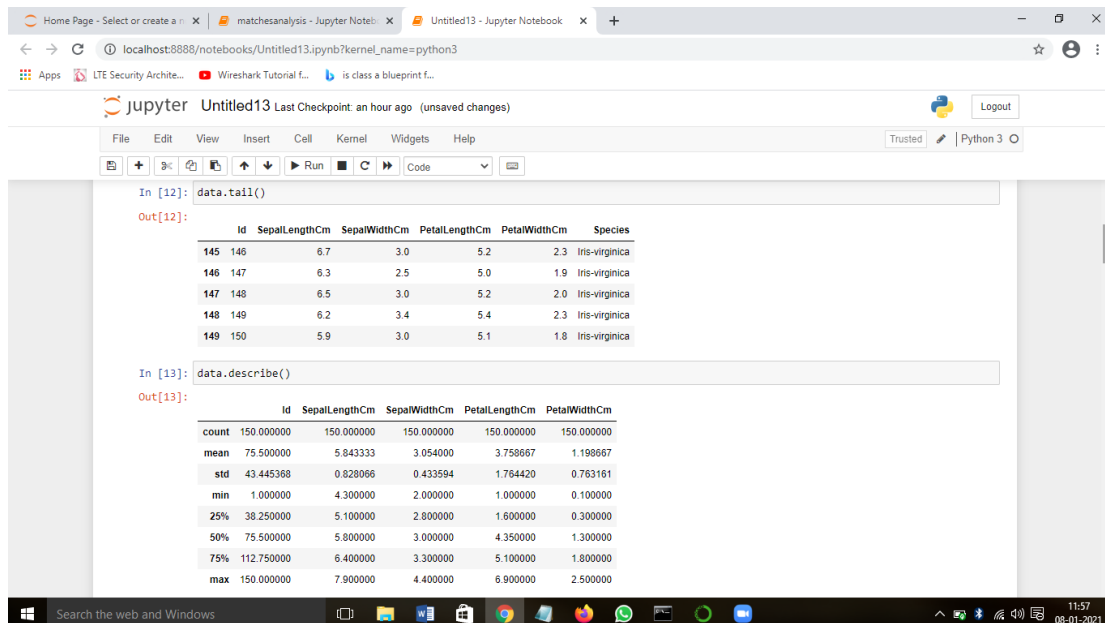
Out[3]:
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

```
In [12]: data.tail()

Out[12]:
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
145	146	6.7	3.0	5.2	2.3	Iris-virginica



A screenshot of a Jupyter Notebook interface, continuing from the previous one. The browser tabs and address bar are the same. The notebook content shows the following code and output:

```
In [12]: data.tail()

Out[12]:
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

```
In [13]: data.describe()

Out[13]:
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

```
Home Page - Select or create a notebook | matchesanalysis - Jupyter Notebooks | Untitled13 - Jupyter Notebook | +
localhost:8888/notebooks/Untitled13.ipynb?kernel_name=python3
jupyter Untitled13 Last Checkpoint: an hour ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
In [14]: data.dtypes
Out[14]:
Id                int64
SepalLengthCm     float64
SepalWidthCm       float64
PetalLengthCm      float64
PetalWidthCm       float64
Species           object
dtype: object

In [15]: print(data.isnull().sum())
Id                0
SepalLengthCm     0
SepalWidthCm       0
PetalLengthCm      0
PetalWidthCm       0
Species           0
dtype: int64

In [4]: data.shape
Out[4]: (150, 6)

In [5]: data.columns
Out[5]: Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
              'Species'],
              dtype=object)
```

```
Home Page - Select or create a notebook | matchesanalysis - Jupyter Notebooks | Untitled13 - Jupyter Notebook | +
localhost:8888/notebooks/Untitled13.ipynb?kernel_name=python3
jupyter Untitled13 Last Checkpoint: an hour ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
In [11]: data['Species'].value_counts()
Out[11]:
Iris-setosa      50
Iris-virginica   50
Iris-versicolor  50
Name: Species, dtype: int64

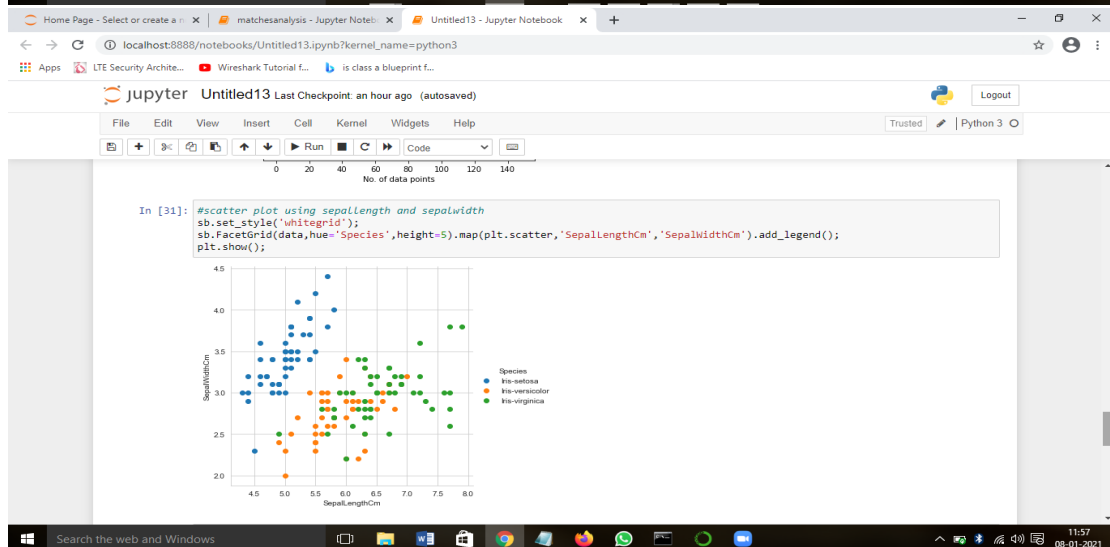
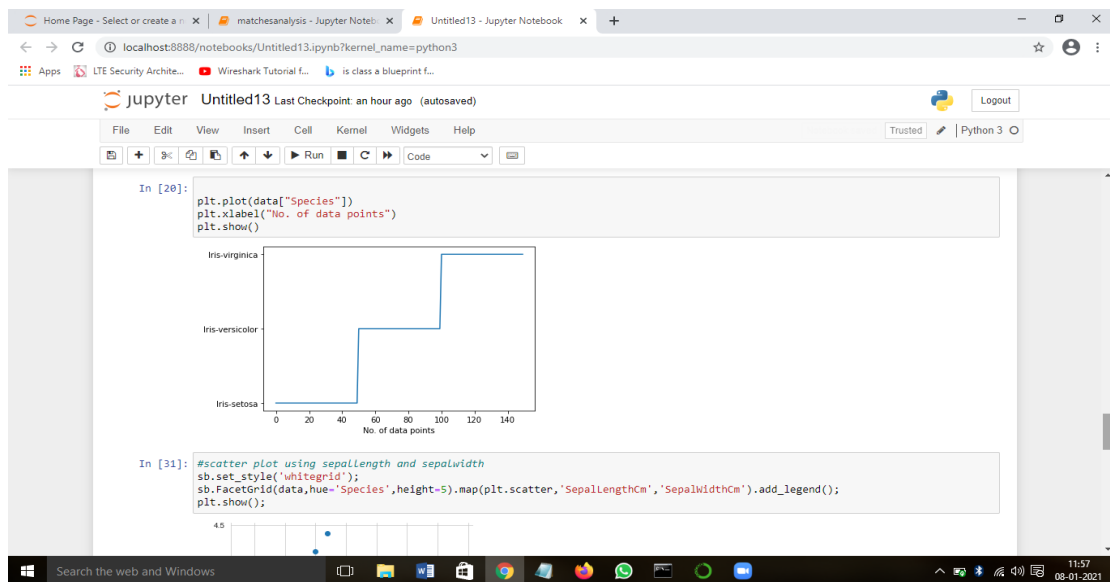
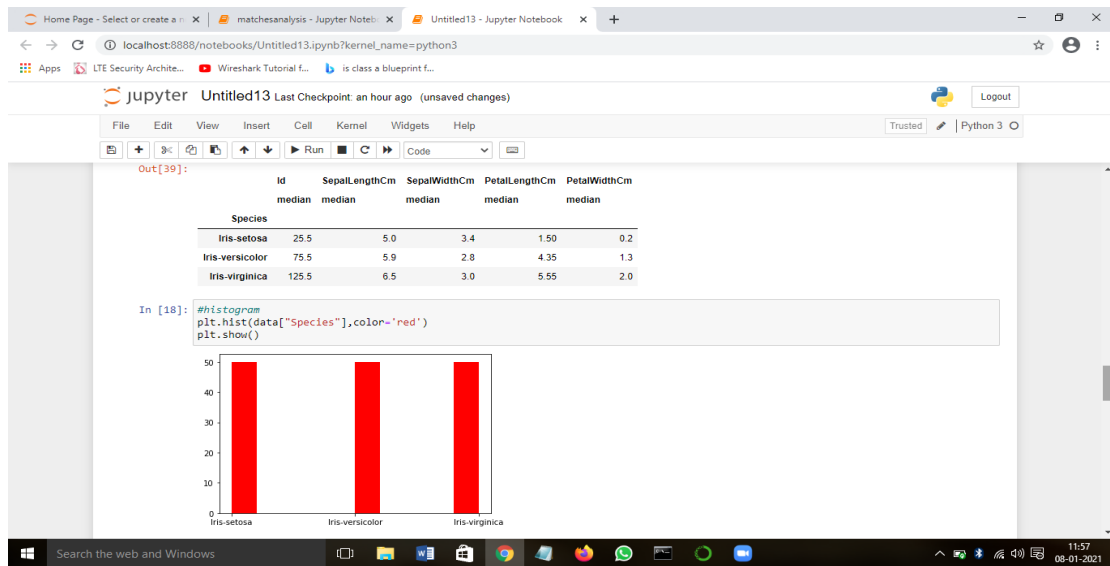
In [16]: data.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  ---
0    Id              150 non-null    int64
1    SepalLengthCm   150 non-null    float64
2    SepalWidthCm    150 non-null    float64
3    PetalLengthCm   150 non-null    float64
4    PetalWidthCm    150 non-null    float64
5    Species         150 non-null    object
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB

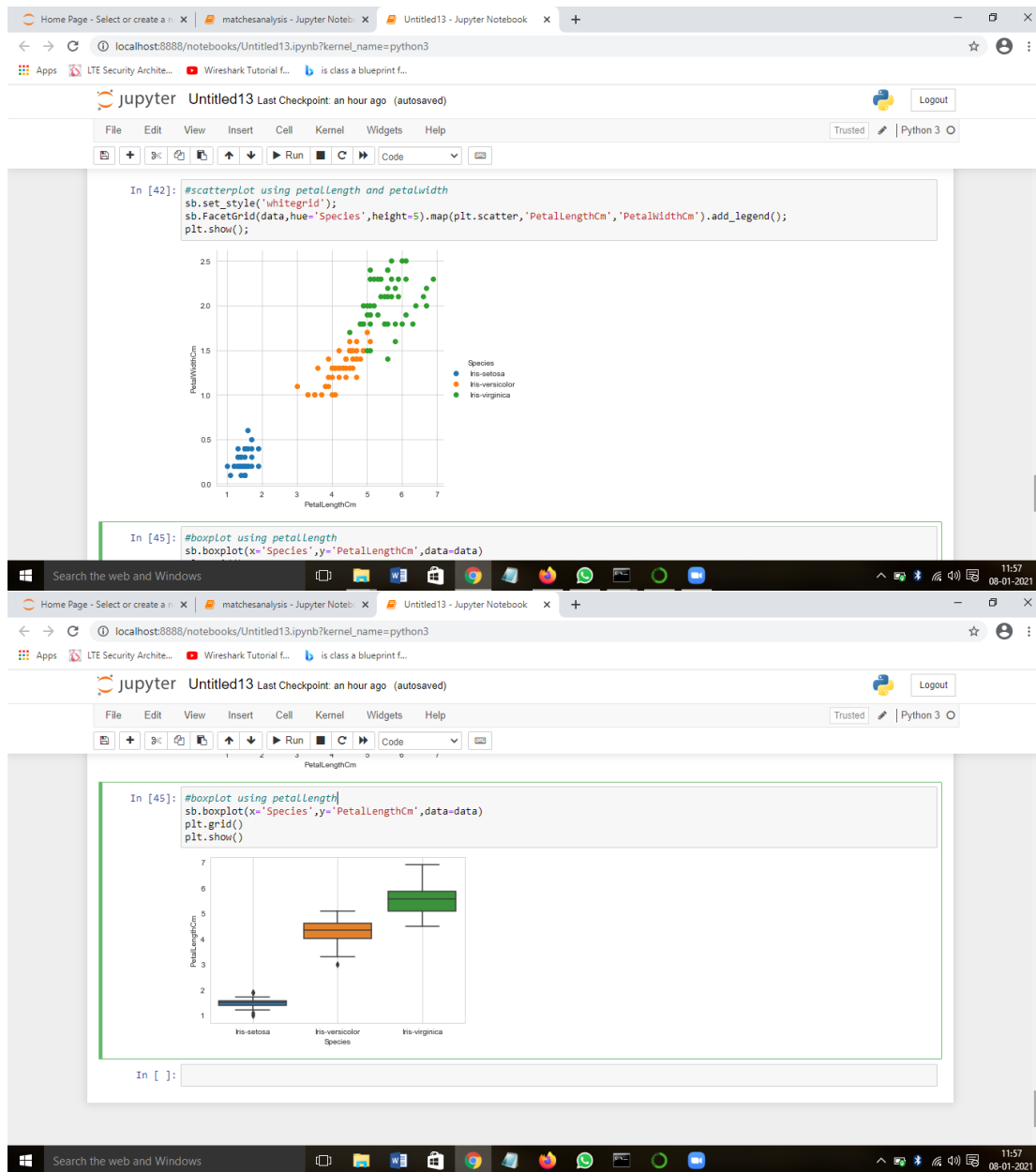
In [35]: data.groupby('Species').agg(['min'])
Out[35]:
              Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
Species
min min              min              min              min
Iris-setosa    1              4.3              2.3              1.0              0.1
Iris-versicolor  51              4.9              2.0              3.0              1.0
Iris-virginica  101             4.9              2.2              4.5              1.4
```

```
Home Page - Select or create a notebook | matchesanalysis - Jupyter Notebooks | Untitled13 - Jupyter Notebook | +
localhost:8888/notebooks/Untitled13.ipynb?kernel_name=python3
jupyter Untitled13 Last Checkpoint: an hour ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
In [35]: data.groupby('Species').agg(['min'])
Out[35]:
              Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
Species
min min              min              min              min
Iris-setosa    1              4.3              2.3              1.0              0.1
Iris-versicolor  51              4.9              2.0              3.0              1.0
Iris-virginica  101             4.9              2.2              4.5              1.4

In [36]: data.groupby('Species').agg(['max'])
Out[36]:
              Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
Species
max max              max              max              max
Iris-setosa    50              5.8              4.4              1.9              0.6
Iris-versicolor 100              7.0              3.4              5.1              1.8
Iris-virginica  150              7.9              3.8              6.9              2.5

In [37]: data.groupby('Species').agg(['std'])
Out[37]:
              Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
Species
std std              std              std              std
Iris-setosa    1.777071          1.777071          1.777071          1.777071
Iris-versicolor 1.777071          1.777071          1.777071          1.777071
Iris-virginica  1.777071          1.777071          1.777071          1.777071
```





Chapter 7

Conclusion

Hence we can say from the above that Analysis is very important as It helps us in achieving the following:-

- 1.It helps in detection of mistakes (like missing values and outliers) .
- 2.It determines relationships between explanatory variables.
- 3.Assessing the direction and rough size of relationships.
Between explanatory and outcome variables.
- 4.It makes our data ready for machine learning algorithm.