DATA ANALYSIS

USING PYTHON

SUBMITTED BY: SHIZA KHAN, S. AELIYA ZAIDI & NOOR UL AIN

SUBMITTED TO: UMAID AHMAD

DATED: 4th JAN, 2021

ABSTRACT:

Data-analytics is becoming a very influential tool for decision-making today both in industry and academia. The incorporation of data-driven concepts in the core curriculum would be very beneficial to graduates, making them competitive in today's market. The main objective of this work is to develop tools and case studies that: (a) increase students' understanding and appreciation of data for decision making, (b) introduce students to basic data analytics and machine learning methods, (c) introduce students to basic data visualization tools, and (d) exhibit that the interactions between rigorous simulations and data could lead to improved solutions. This report about data analysis using Python will be very useful to the ones looking for inner workings of data processing.

This report comprises two parts including theoretical part which demonstrates the introduction of data analysis, data analysis vs data science, data wrangling, normalization, formatting, binning. Additionally, practical part exemplifies the data analysis of sales which is completed by Python based data analysis. Finally, this report concludes the preeminent data analysis method by describing its features and recommend for individuals who want to develop career in data science as well as refers to future of data analysis.

CONCEPT DEFINITIONS

CSV: comma-separated values
MSE: Mean squared error
MLR: Multiple linear Regression
SLR: Simple Linear Regression
SAS: Statistical Analysis Software
MATLAB: Matrix Laboratory
SCIPY: (Scientific python) is a free and open-source Python library used for scientific computing and technical computing.
NUMPY: Numeric Python
MATPLOTLIB: is a plotting library for the Python
IQR: Interquartile Range

Key words

CSV, Matplotlib, NumPy, Pandas, Plotly, SciPy, Seaborn, Sci-kit learn

ABSTRACT

CONCEPT DEFINITIONS

CONTENTS

INTRODUCTION:

Data Analytics refers to the techniques used to analyze data to enhance productivity and business gain. Data is extracted from various sources and is cleaned and categorized to analyze various behavioral patterns. The techniques and the tools used vary according to the organization or individual. So, in short, if you understand your Business Administration and have the capability to perform Exploratory Data Analysis, to gather the required information, then you are good to go with a career in Data Analytics.

Data analytics is the investigation of separating unrefined data to settle on choices about the information. A noteworthy number of the strategies and methods of information investigation have been automated into mechanical systems and figurines that work over unrefined data for human use. Data analytics methods can uncover patterns and measurements that would some way or another be lost in the mass of data. This data would then be able to be utilized to improve procedures to expand the general productivity of a business or framework. Data analytics is a wide term that incorporates numerous various kinds of information examination. Any kind of data can be exposed to information examination methods to get the knowledge that can be utilized to improve things. For instance, fabricating organizations regularly record the runtime, personal time, and work line for different machines and after that investigate the information to even more likely arrangement the remaining tasks at hand, so the machines work nearer to crest limit.

DATA ANALYSIS:

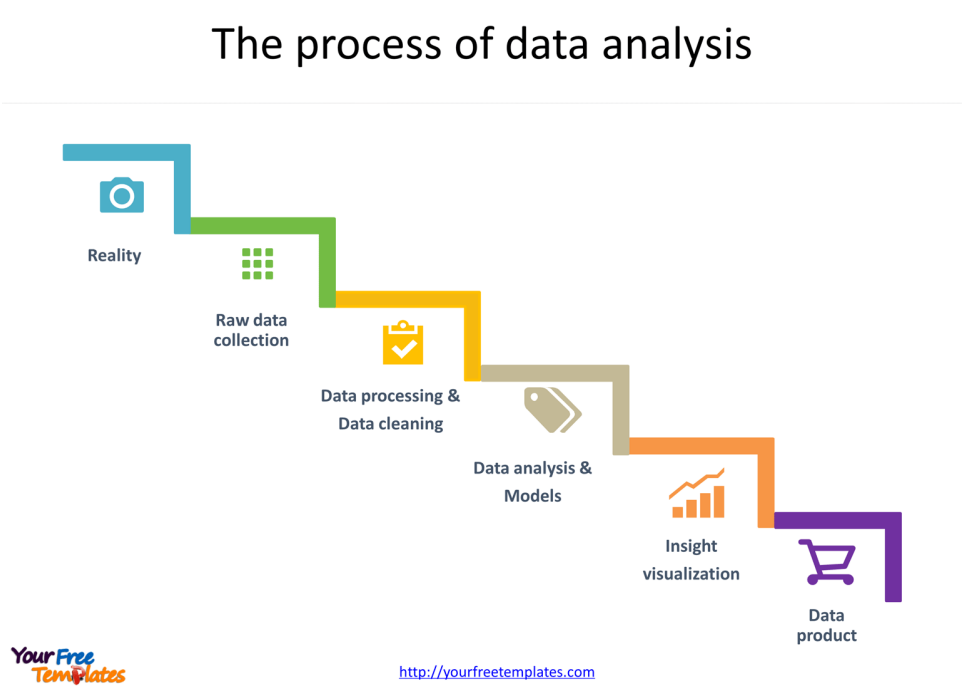
Data analysis is defined as a process of cleaning, transforming, and modeling data to discover useful information for business decision-making. The purpose of Data Analysis is to extract useful information from data and taking the decision based upon the data analysis.

A simple example of Data analysis is whenever we take any decision in our day-to-day life is by thinking about what happened last time or what will happen by choosing that particular decision. This is nothing but analyzing our past or future and making decisions based on it. For that, we gather memories of our past or dreams of our future. So that is nothing but data analysis. Now same thing analyst does for business purposes, is called Data Analysis.

ROLE OF DATA ANALYSIS:

The responsibilities of a data analyst typically include the following:

- Designing and maintaining data systems and databases; this includes fixing coding errors and other data-related problems.
- Mining data from primary and secondary sources, then reorganizing said data in a format that can be easily read by either human or machine.
- Using statistical tools to interpret data sets, paying particular attention to trends and patterns that could be valuable for diagnostic and predictive analytics efforts.
- Demonstrating the significance of their work in the context of local, national, and global trends that impact both their organization and industry.
- Preparing reports for executive leadership that effectively communicate trends, patterns, and predictions using relevant data.
- Collaborating with programmers, engineers, and organizational leaders to identify opportunities for process improvements, recommend system modifications, and develop policies for data governance.
- Creating appropriate documentation that allows stakeholders to understand the steps of the data analysis process and duplicate or replicate the analysis if necessary.



Information analysis is a practice for constructing original data and solving various models in the data through numerical estimation and calculation. Analyzing data facilitates to create new information and accessible various bits of information. Obtaining data from an organization's database or deleting data from external sources for research is one of the major occupations of any data analyst. Clearing information is the basic stage of the entire data classification process, and a stupid and unimproved way to review, identify, and use the data. When investigating deterministic data to select key decisions, data analysts should start with prudent data cleansing methods. It is important to check the phenomena established by data cleansing, and cleansing incorporates data deletion operations that may destroy score or organize information in a single way.

PYTHON IN DATA ANALYSIS

Python is a deciphered, significant level, universally useful programming language invented by Guido van Rossum. First released in 1991, Python's structure reasoning underscores code coherence with its eminent utilization of critical whitespace. Its language builds an article arranged methodology implies to assist software engineers with composing clear, coherent code for bit and huge scale of projects. Python is progressively composed, and trash gathered. It underpins different programming ideal models, including procedural, object-oriented, and practical programming. Python is frequently portrayed as a battery included language because of its exhaustive standard library.

Data Science has increased a terrific agreement of infamy over the most recent couple of years. This current field's important facility is to transfer important data into advertising and business methodologies which enables an organization to develop. The data is put away and inquired about to find in a coherent arrangement. Previously, just the top IT organizations were associated with this field but today organizations from different parts and fields, for example, online business, medicinal services, endowment, and others are utilizing information examination. There are various gadgets open for information investigation, for instance, Hadoop, R programming, SAS and SQL. Finally, Python is the most standard and easy to use instrument for information investigation which is known as a Swiss Army cutting edge of the coding scene since it underlines organized programming and object-oriented programming similarly as the helpful programming language then others. According to the Stack Overflow of 2018, the most standard programming language on world and called the most sensible language for data science mechanical assemblies and applications is Python.

DATA SCIENCE

As the world entered the era of big data, the need for its storage also grew. It was the main challenge and concern for the enterprise industries until 2010. The main focus was on building a framework and solutions to store data. Now when Hadoop and other frameworks have successfully solved the problem of storage, the focus has shifted to the processing of this data. Data Science is the secret sauce here. Data Science is the future of Artificial Intelligence. Therefore, it is very important to understand what is Data Science and how can it add value to your business.

Data Science is a blend of various tools, algorithms, and machine learning principles with the goal to discover hidden patterns from the raw data. Data Science is primarily used to make decisions and predictions making use of predictive causal analytics, prescriptive analytics (predictive plus decision science) and machine learning.

PYTHON FOR DATA SCIENCE

Compared to quantitative and logical processing, Python has a unique attribution and easy to use. It is a longstanding trade leader and often widely used in various industries such as oil and gas, signal management and funds. In addition, Python has been used to strengthen Google's internal foundations and build applications like YouTube. Python is commonly used, a very popular device and an adaptable language that has been released to the public. Its massive library is used for information control, and it is very easy for beginners to learn. In addition to remain an autonomous stadium, it effectively incorporates any current frameworks that can be used to solve the most surprising problems. Most banks use it to process information, organizations use it for perception and preparation, and climate indicator organizations for instance predictive analytics also use it. Python is preferred over other Data Science tools attributable to the different fields. Python is considered a primitive language, and any substitute or scientist with simple basic data can start experimenting. The time required to analyze the code and the various restrictions on removing programming signatures are also limited. Compared to other programming languages (such as C, Java, and C #), the ideal opportunity to execute code is fewer, which helps designers and architects to invest more energy in computing. Python provides an extensive database library that includes artificial knowledge and artificial intelligence. The most popular libraries include Scikit Learn, TensorFlow, Seaborn, Pytorch and Matplotlib. Many operational, Data Science and Artificial Intelligence resources are available online for effective access. Compared to other programming languages such as Java and R, Python is considered a more adaptable and faster language. It provides the flexibility to solve problems that cannot be solved through other programming languages. Many organizations use it to create various types of applications and fast devices. There are different perceptual alternatives in Python. Matplotlib library provides a solid foundation on which can create different libraries, such as Ggplot, Panda Drawing and PyTorch. These packages help develop a framework that can be used for the web and graphic design.

IMPORTANCE OF DATA SCIENCE

Below are some reasons which show that data science will always be a significant part of the economy of the global world.

1. With the help of Data Science, the companies will be able to recognize their client in a more improved and enhanced way. Clients are the foundation of any product and play an essential role in their success and failure. Data Science enables companies to connect with their

customers in a modified manner and thus confirms the better quality and power of the product.

2. Data Science allows products to tell their story powerfully and engagingly. This is one of the reasons which makes it popular. When product and companies use this data inclusively, they can share their story with their viewers and thus creating better product connections.

3. One of the important features of Data Science is that its results can be applied to almost all types of industries such as travel, healthcare and education. With the help of Data Science, the industries can analyze their challenges easily and can also address them effectively.

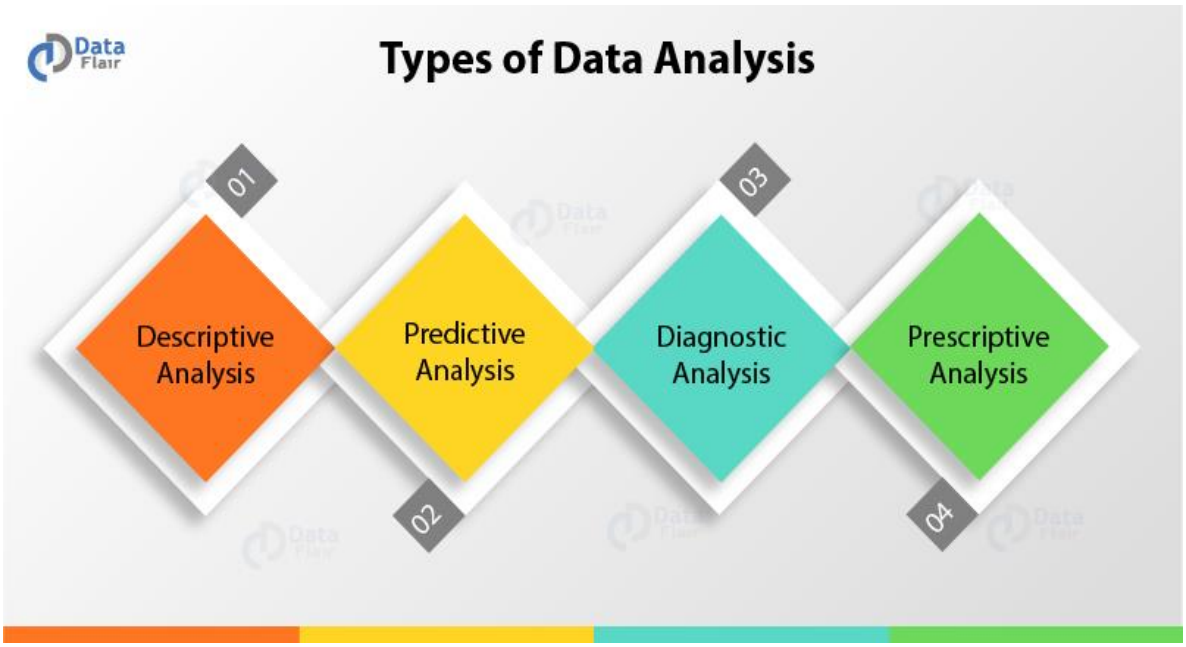
4. Presently, data science is available in almost all the fields and there is a vast amount of data present in the world today and if it is used properly it can lead the product to the success or failure. If data is used correctly it will hold the importance for achieving goals for the product in the future.

5. Big data is continuously emerging and growing. Using various tools which are developed regularly, big data helps the organization to resolve complex issues related to IT, human resource and resource management efficiently and successfully.

6. Data science is gaining popularity in every industry and thus playing a significant role in functioning and growth of any product. Therefore, the requirement of data scientist is also increased as they have to perform an important task of handling data and delivering solutions for the specific problems.

7. Data science also influenced the retail industries. Let’s take an example to understand this, the older people were having a fantastic interaction with the local seller. The seller was also capable of fulfilling the requirements of the clients in a personalized way. But now due to the emergence and increase of supermarkets, this attention got lost. But with the help of data analytics, it is possible for the sellers to connect with their clients.

8. Data Science helps organizations to build this connection with the clients. With the help of data science, organizations and their products will be able to create a better and deep understanding of how customers can utilize their products.



PYTHON PACKAGE FOR DATA SCIENCE

Python Libraries and Packages are several helpful modules and capacities that reduce the utilization of code in everyday life. There are more than 137,000 Python libraries and 198,826 Python bundles prepared to facilitate engineers' standard programming experience. These libraries and bundles are anticipated for an assortment of cutting-edge arrangements. Python libraries and Python bundles believe a fundamental job in ordinary AI. Indeed, their utilization is not restricted to AI when it happened. Information Science, picture and information control, information perception – everything is a piece of their moderate applications. Python Packages are a set of Python modules, while Python libraries are a group of Python functions aimed to carry out special tasks.

NUMPY

One of the most fundamental packages in Python, NumPy is a general-purpose array-processing package. It provides high-performance multidimensional array objects and tools to work with the arrays. NumPy is an efficient container of generic multi-dimensional data.

NumPy’s main object is the homogeneous multidimensional array. It is a table of elements or numbers of the same datatype, indexed by a tuple of positive integers. In NumPy, dimensions are called axes and the number of axes is called rank. NumPy’s array class is called ndarray aka array.

NumPy is used to process arrays that store values of the same datatype. NumPy facilitates math operations on arrays and their vectorization. This significantly enhances performance and speeds up the execution time correspondingly. Some of the functions of NumPy are:

- Basic array operations: add, multiply, slice, flatten, reshape, index arrays
- Advanced array operations: stack arrays, split into sections, broadcast arrays
- Work with Date Time or Linear Algebra
- Basic Slicing and Advanced Indexing in NumPy Python

PANDAS

Pandas is an open-source Python package that provides high-performance, easy-to-use data structures and data analysis tools for the labeled data in Python programming language. Pandas stand for Python Data Analysis Library. Pandas is a perfect tool for data wrangling or munging. It is designed for quick and easy data manipulation, reading, aggregation, and visualization.

Pandas take data in a CSV or TSV file or a SQL database and create a Python object with rows and columns called a data frame. The data frame is very similar to a table in statistical software, say Excel or SPSS. Functions of Pandas are as follows:

- Indexing, manipulating, renaming, sorting, merging data frame
- Update, Add, Delete columns from a data frame
- Impute missing files, handle missing data or NANs
- Plot data with histogram or box plot

This makes Pandas a foundation library in learning Python for Data Science.

MATPLOTLIB

This is undoubtedly a quintessential Python library. You can create stories with the data visualized with Matplotlib. Another library from the SciPy Stack, Matplotlib plots 2D figures. Matplotlib is the plotting library for Python that provides an object-oriented API for embedding plots into applications. It is a close resemblance to MATLAB embedded in Python programming language.

Histogram, bar plots, scatter plots, area plot to pie plot, Matplotlib can depict a wide range of visualizations. With a bit of effort and tint of visualization capabilities, with Matplotlib, you can create just any visualizations:

Line plots

Scatter plots

Area plots

Bar charts and Histograms

Pie charts

Stem plots

Contour plots

Quiver plots

Spectrograms

Matplotlib also facilitates labels, grids, legends, and some more formatting entities.

SCIPY

The SciPy library is one of the core packages that make up the SciPy stack. Now, there is a difference between SciPy Stack and SciPy, the library. SciPy builds on the NumPy array object and is part of the stack which includes tools like Matplotlib, Pandas, and SymPy with additional tools.

SciPy library contains modules for efficient mathematical routines as linear algebra, interpolation, optimization, integration, and statistics. The main functionality of the SciPy library is built upon NumPy and its arrays. SciPy makes significant use of NumPy.

SciPy uses arrays as its basic data structure. It has various modules to perform common scientific programming tasks as linear algebra, integration, calculus, ordinary differential equations, and signal processing.

SEABORN

It is defined as the data visualization library based on Matplotlib that provides a high-level interface for drawing attractive and informative statistical graphics. Putting it simply, seaborn is an extension of Matplotlib with advanced features.

So, what is the difference between Matplotlib and Seaborn? Matplotlib is used for basic plotting; bars, pies, lines, scatter plots and stuff whereas, seaborn provides a variety of visualization patterns with less complex and fewer syntax.

Functions of Seaborn are as follows:

- Determine relationships between multiple variables (correlation)
- Observe categorical variables for aggregate statistics

- Analyze uni-variate or bi-variate distributions and compare them between different data subsets
- Plot linear regression models for dependent variables
- Provide high-level abstractions, multi-plot grids

Seaborn is a great second-hand for R visualization libraries like corrplot and ggplot.

SCIKIT LEARN

Introduced to the world as a Google Summer of Code project, Scikit Learn is a robust machine learning library for Python. It features ML algorithms like SVMs, random forests, k-means clustering, spectral clustering, mean shift, cross-validation and more. Even NumPy, SciPy and related scientific operations are supported by Scikit Learn with Scikit Learn being a part of the SciPy Stack.

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. Supervised learning models like Naive Bayes to grouping unlabeled data such as KMeans, Scikit learn would be your go-to.

Functions of Scikit Learn are as follows

- Classification: Spam detection, image recognition
- Clustering: Drug response, Stock price
- Regression: Customer segmentation, Grouping experiment outcomes
- Dimensionality reduction: Visualization, Increased efficiency
- Model selection: Improved accuracy via parameter tuning
- Pre-processing: Preparing input data as a text for processing with machine learning algorithms.

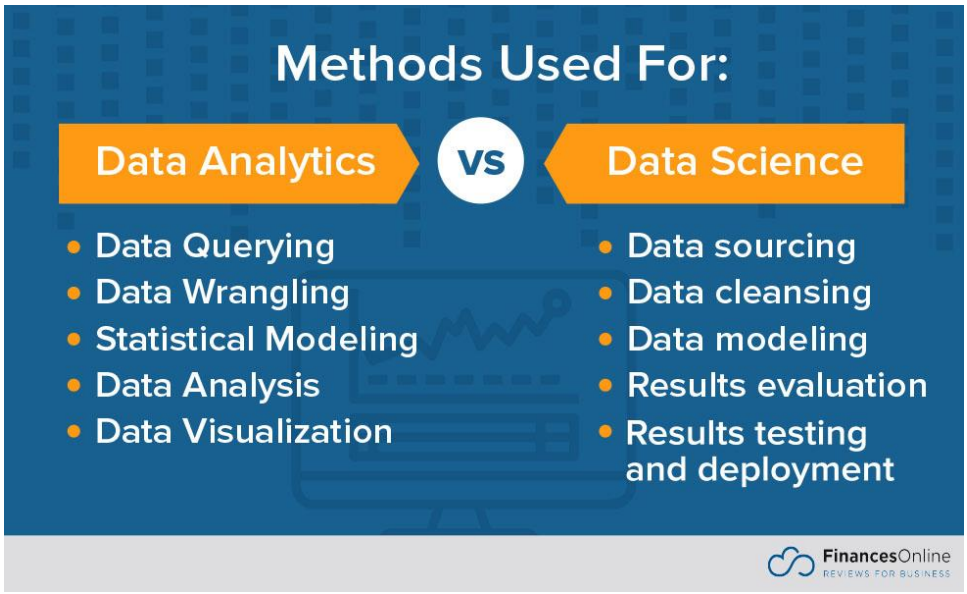
Scikit Learn focuses on modeling data; not manipulating data. We have NumPy and Pandas for summarizing and manipulation.

DATA SCIENCE VS DATA ANALYSIS

While data analysts and data scientists both work with data, the main difference lies in what they do with it. Data analysts examine large data sets to identify trends, develop charts, and create visual presentations to help businesses make more strategic decisions. Data scientists, on the other hand, design and construct new processes for data modeling and production using prototypes, algorithms, predictive models, and custom analysis.

The two fields can be considered different sides of the same coin, and their functions are highly interconnected. Data science lays important foundations and parses big datasets to create initial observations, future trends, and potential insights that can be important. This information by itself is useful for some fields, especially modeling, improving machine

learning, and enhancing AI algorithms as it can improve how information is sorted and understood. However, data science asks important questions that we were unaware of before while providing little in the way of hard answers. By adding data analytics into the mix, we can turn those things we know we don't know into actionable insights with practical applications.



Another basic component of establishing separate Data Analysis and Science is the definition point or goal of each control. Although it stated the concept, it is very important and worth emphasizing: the basic goal of data science is to use a large number of advanced easy to use methods and pieces of knowledge to find the questions that need to be asked to guide, development, progress and development. Using existing data to reveal projects and knowledge images in clear areas as the basic point and Data Analysis aims to obtain extraordinary information that depends on clear points, activities and key performance indicators.

DATA WRANGLING

Data Wrangling, sometimes referred to as Data Munging, is the process of transforming and mapping data from one raw data form into another format with the intent of making it more appropriate and valuable for a variety of downstream purposes such as analytics. The goal of data wrangling is to assure quality and useful data. Data analysts typically spend the majority of their time in the process of data wrangling compared to the actual analysis of the data.

Data wrangling is increasingly ubiquitous at today's top firms. Data has become more diverse and unstructured, demanding increased time spent culling, cleaning, and organizing data ahead of broader analysis. At the same time, with data informing just about every business decision, business users have less time to wait on technical resources for prepared data.

This necessitates a self-service model, and a move away from IT-led data preparation, to a more democratized model of self-service data preparation or data wrangling. This self-service model with data wrangling tools allows analysts to tackle more complex data more quickly, produce more accurate results, and make better decisions. Because of this ability, more businesses have started using data wrangling tools to prepare before analysis.

PRE-PROCESSING DATA

Data preprocessing is an important step in the data mining process. The phrase "garbage in, garbage out" is particularly applicable to data mining and machine learning projects. Data-gathering methods are often loosely controlled, resulting in out-of-range values (e.g., Income: −100), impossible data combinations (e.g., Sex: Male, Pregnant: Yes), and missing values, etc. Analyzing data that has not been carefully screened for such problems can produce misleading results. Thus, the representation and quality of data is first and foremost before running any analysis. Often, data preprocessing is the most important phase of a machine learning project.

Data preprocessing incorporates cleaning, instance determination, standardization, change, highlight extraction, and choice, and so on. The result of data preprocessing is the last preparing set. Data pre-processing may influence how the results of the last data preparation can be deciphered. This viewpoint ought to be deliberately viewed as when the translation of the outcomes is a key point, such in the multivariate preparation of compound data.

DEALING WITH MISSING VALUES

Most datasets contain missing data, mistakenly encoded data, or other data that cannot be utilized for demonstrating. Occasionally missing data is only that — missing. There is no genuine incentive in each field, for instance, an unfilled string in a CSV record. Different occasions data is encoded with an extraordinary watchword or a string. Some basic encodings are NA, N/A, None, and - 1. Before utilizing data with missing data fields, it must change those fields so they can be utilized for examination and display. There are AI calculations and bundles that can naturally distinguish and manage missing data; however, it is however a reasonable practice to change that data physically.

DATA FORMATTING

Data is displayed in various sizes and shapes, which can be digital data, content, mixed media, data query or several different types of data. Information design is considered an organization for encoding data which is encoded in different encodings that various applications and projects can inspect, perceive and use. When selecting a data set, there are a few issues to check, such as data attributes or data size, company foundation and use case. When checking the read and write speed of a data file, some tests are performed to select

the correct data set. In most cases, three different kinds of data elements are also referred to as GIS data sets. Data locations of these elements are administered interchangeably and used for various objectives.

DATA NORMALIZATION

Data normalization is generally considered the development of clean data. Diving deeper, however, the meaning or goal of data normalization is twofold:

- Data normalization is the organization of data to appear similar across all records and fields.
- It increases the cohesion of entry types leading to cleansing, lead generation, segmentation, and higher quality data.

Simply put, this process includes eliminating unstructured data and redundancy (duplicates) in order to ensure logical data storage. When data normalization is done correctly, you will end up with standardized information entry. For example, this process applies to how URLs, contact names, street addresses, phone numbers, and even codes are recorded. These standardized information fields can then be grouped and read swiftly.

METHODS OF NORMALIZING DATA

Min-max normalization is one of the most common ways to normalize data. For every feature, the minimum value of that feature gets transformed into a 0, the maximum value gets transformed into a 1, and every other value gets transformed into a decimal between 0 and 1.

For example, if the minimum value of a feature was 20, and the maximum value was 40, then 30 would be transformed to about 0.5 since it is halfway between 20 and 40. The formula is as follows:

value – min / max – min

SIMPLE FEATURE SCALING IN PYTHON

Feature Scaling or Standardization: It is a step of Data Pre Processing which is applied to independent variables or features of data. It basically helps to normalise the data within a particular range. Sometimes, it also helps in speeding up the calculations in an algorithm.

MIN() AND MAX() IN PYTHON

This article brings you a very interesting and lesser known function of Python, namely max() and min(). Now when compared to their C++ counterpart, which only allows two arguments, that too strictly being float, int or char, these functions are not only limited to 2 elements, but can hold many elements as arguments and also support strings in their

arguments, hence allowing to display lexicographically smallest or largest string as well. Detailed functionality are explained below.

max()

This function is used to compute the maximum of the values passed in its argument and lexicographically largest value if strings are passed as arguments.

Syntax :
max(a,b,c,..,key,default)
Parameters :
a,b,c,.. : similar type of data.
key : key function where the iterables are passed and comparsion is performed
default : default value is passed if the given iterable is empty
Return Value :
Returns the maximum of all the arguments.
Exceptions :
Returns TypeError when conflicting types are compared.

```
# Python code to demonstrate the working of max()

# printing the maximum of 4,12,43.3,19,100

print("Maximum of 4,12,43.3,19 and 100 is : ",end="")

print (max( 4,12,43.3,19,100 ) )
```

Output :

```
Maximum of 4,12,43.3,19 and 100 is : 100
```

min()

This function is used to compute the minimum of the values passed in its argument and lexicographically smallest value if strings are passed as arguments.

Syntax :
min(a,b,c,.., key,default)
Parameters :
a,b,c,.. : similar type of data.
key : key function where the iterables are passed and comparsion is performed

default : default value is passed if the given iterable is empty
Return Value :
Returns the minimum of all the arguments.
Exceptions :
Returns TypeError when conflicting types are compared.

```
# Python code to demonstrate the working of min()

# printing the minimum of 4,12,43.3,19,100

print("Minimum of 4,12,43.3,19 and 100 is : ",end="")

print (min( 4,12,43.3,19,100 ) )
```

Output :

Minimum of 4,12,43.3,19 and 100 is : 4

Z-SCORE IN PYTHON

Z score is an important concept in statistics. Z score is also called standard score. This score helps to understand if a data value is greater or smaller than mean and how far away it is from the mean. More specifically, Z score tells how many standard deviations away a data point is from the mean.

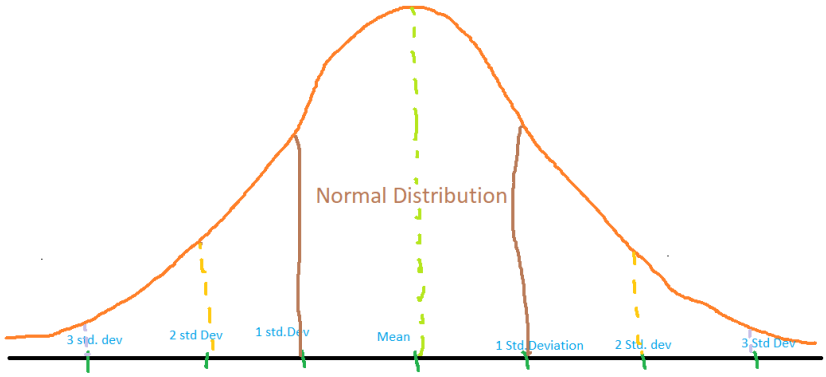
Z score = (x -mean) / std. Deviation

A normal distribution is shown below and it is estimated that

68% of the data points lie between +/- 1 standard deviation.

95% of the data points lie between +/- 2 standard deviation

99.7% of the data points lie between +/-3 standard deviation

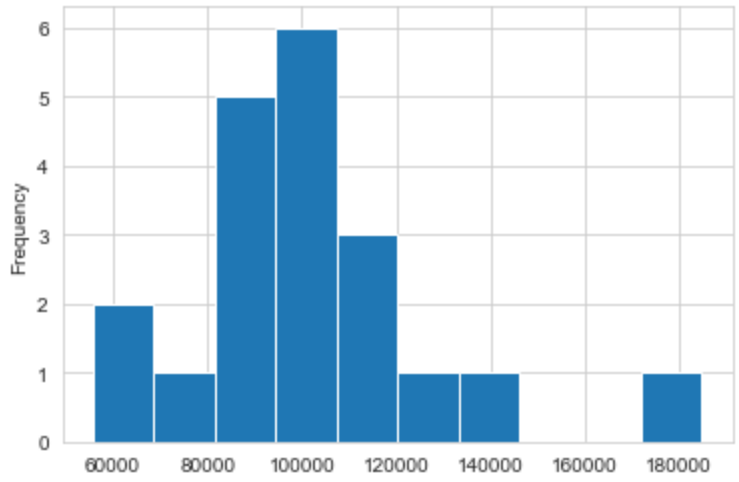


BINNING IN PYTHON

One of the most common instances of binning is done behind the scenes for you when creating a histogram. The histogram below of customer sales data, shows how a continuous set of sales numbers can be divided into discrete bins (for example: $60,000 - $70,000) and then used to group and count account instances.

Here is the code that show how we summarize 2018 Sales information for a group of customers. This representation illustrates the number of customers that have sales within certain ranges.

```
import pandas as pd
import numpy as np
import seaborn as sns

sns.set_style('whitegrid')

raw_df = pd.read_excel('2018_Sales_Total.xlsx')
df = raw_df.groupby(['account number', 'name'])['ext price'].sum().reset_index()

df['ext price'].plot(kind='hist')
```



EXPLORATORY DATA ANALYSIS IN PYTHON

Exploratory Data Analysis or (EDA) is understanding the data sets by summarizing their main characteristics often plotting them visually. This step is very important especially when we arrive at modeling the data in order to apply Machine learning. Plotting in EDA consists of Histograms, Box plot, Scatter plot and many more. It often takes much time to explore the data. Through the process of EDA, we can ask to define the problem statement or definition on our data set which is very important.

GROUPBY IN PYTHON

groupby() function is used to split the data into groups based on some criteria. pandas objects can be split on any of their axes. The abstract definition of grouping is to provide a mapping of labels to group names.

MACHINE LEARNING MODEL EVALUATION

Machine learning continues to be an increasingly integral component of our lives, whether we're applying the techniques to research or business problems. Machine learning models ought to be able to give accurate predictions in order to create real value for a given organization.
While training a model is a key step, how the model generalizes on unseen data is an equally important aspect that should be considered in every machine learning pipeline. We need to know whether it actually works and, consequently, if we can trust its predictions. Could the model be merely memorizing the data it is fed with, and therefore unable to make good predictions on future samples, or samples that it hasn't seen before?
In this article, we explain the techniques used in evaluating how well a machine learning model generalizes to new, previously unseen data. We'll also illustrate how common model evaluation metrics are implemented for classification and regression problems using Python.

Model Evaluation Techniques

The above issues can be handled by evaluating the performance of a machine learning model, which is an integral component of any data science project. Model evaluation aims to estimate the generalization accuracy of a model on future (unseen/out-of-sample) data.

Methods for evaluating a model's performance are divided into 2 categories:
namely, holdout and Cross-validation. Both methods use a test set (i.e data not seen by the model) to evaluate model performance. It's not recommended to use the data we used to build the model to evaluate it. This is because our model will simply remember the whole training set, and will therefore always predict the correct label for any point in the training set. This is known as overfitting.

Holdout

The purpose of holdout evaluation is to test a model on different data than it was trained on. This provides an unbiased estimate of learning performance.In this method, the dataset is randomly divided into three subsets:

1. **Training set** is a subset of the dataset used to build predictive models.
2. **Validation set** is a subset of the dataset used to assess the performance of the model built in the training phase. It provides a test platform for fine-tuning a model's

parameters and selecting the best performing model. Not all modeling algorithms need a validation set.

3. **Test set**, or unseen data, is a subset of the dataset used to assess the likely future performance of a model. If a model fits to the training set much better than it fits the test set, overfitting is probably the cause.

The holdout approach is useful because of its speed, simplicity, and flexibility. However, this technique is often associated with high variability since differences in the training and test dataset can result in meaningful differences in the estimate of accuracy.

Cross-Validation

Cross-validation is a technique that involves partitioning the original observation dataset into a training set, used to train the model, and an independent set used to evaluate the analysis.

The most common cross-validation technique is k-fold cross-validation, where the original dataset is partitioned into k equal size subsamples, called folds. The k is a user-specified number, usually with 5 or 10 as its preferred value. This is repeated k times, such that each time, one of the k subsets is used as the test set/validation set and the other k-1 subsets are put together to form a training set. The error estimation is averaged over all k trials to get the total effectiveness of our model.

For instance, when performing five-fold cross-validation, the data is first partitioned into 5 parts of (approximately) equal size. A sequence of models is trained. The first model is trained using the first fold as the test set, and the remaining folds are used as the training set. This is repeated for each of these 5 splits of the data and the estimation of accuracy is averaged over all 5 trials to get the total effectiveness of our model.

As can be seen, every data point gets to be in a test set exactly once and gets to be in a training set k-1 times. This significantly reduces bias, as we're using most of the data for fitting, and it also significantly reduces variance, as most of the data is also being used in the test set. Interchanging the training and test sets also adds to the effectiveness of this method.

Model Evaluation Metrics

Model evaluation metrics are required to quantify model performance. The choice of evaluation metrics depends on a given machine learning task (such as classification, regression, ranking, clustering, topic modeling, among others). Some metrics, such

as precision-recall, are useful for multiple tasks. Supervised learning tasks such as classification and regression constitutes a majority of machine learning applications. In this article, we focus on metrics for these two supervised learning models.

Classification Metrics

In this section we will review some of the metrics used in classification problems, namely: Classification Accuracy

Confusion matrix

Logarithmic Loss

Area under curve (AUC)

F-Measure

Classification Accuracy

Accuracy is a common evaluation metric for classification problems. It's the number of correct predictions made as a ratio of all predictions made. We use sklearn module to compute the accuracy of a classification task, as shown below:

```
#import
modules
        import warnings
        import pandas as pd
        import numpy as np
        from sklearn import model_selection
        from sklearn.linear_model import LogisticRegression
        from sklearn import datasets
        from sklearn.metrics import accuracy_score
        #ignore warnings
        warnings.filterwarnings('ignore')
        # Load digits dataset
        iris = datasets.load_iris()
        # # Create feature matrix
        X = iris.data
        # Create target vector
        y = iris.target
        #test size
        test_size = 0.33
        #generate the same set of random numbers
        seed = 7
        #cross-validation settings
        kfold = model_selection.KFold(n_splits=10, random_state=seed)
```

```
#Model instance
model = LogisticRegression()
#Evaluate model performance
scoring = 'accuracy'
results = model_selection.cross_val_score(model, X, y, cv=kfold, scoring=scoring)
print('Accuracy -val set: %.2f%% (%.2f)' % (results.mean()*100, results.std()))


#split data
X_train, X_test, y_train, y_test = model_selection.train_test_split(X, y,
test_size=test_size, random_state=seed)
#fit model
model.fit(X_train, y_train)
#accuracy on test set
result = model.score(X_test, y_test)
print("Accuracy - test set: %.2f%%" % (result*100.0))
```

The classification accuracy is **88%** on the validation set.

By using cross-validation, we'd be "testing" our machine learning model in the "training" phase to check for overfitting and to get an idea about how our machine learning model will generalize to independent data (test data set).

Cross-validation techniques can also be used to compare the performance of different machine learning models on the same data set and can also be helpful in selecting the values for a model's parameters that maximize the accuracy of the model—also known as parameter tuning.

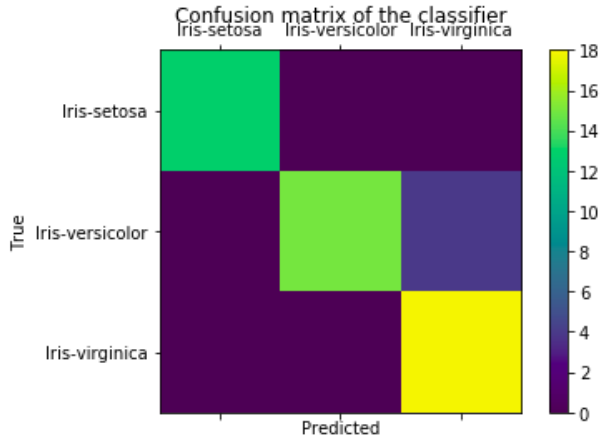
Confusion Matrix

A confusion matrix provides a more detailed breakdown of correct and incorrect classifications for each class. We use the Iris dataset to classify and compute the confusion matrix for the predictions:

```
#import
modules
        import warnings
        import pandas as pd
        from sklearn import model_selection
        from sklearn.linear_model import LogisticRegression
        from sklearn.metrics import confusion_matrix
        import matplotlib.pyplot as plt
        %matplotlib inline
```

```
#ignore warnings
warnings.filterwarnings('ignore')
# Load digits dataset
url = "http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
df = pd.read_csv(url)
# df = df.values
X = df.iloc[:,0:4]
y = df.iloc[:,4]
# print (y.unique())
#test size
test_size = 0.33
#generate the same set of random numbers
seed = 7
#Split data into train and test set.
X_train, X_test, y_train, y_test = model_selection.train_test_split(X, y,
test_size=test_size, random_state=seed)
#Train Model
model = LogisticRegression()
model.fit(X_train, y_train)
pred = model.predict(X_test)


#Construct the Confusion Matrix
labels = ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']
cm = confusion_matrix(y_test, pred, labels)
print(cm)
fig = plt.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(cm)
plt.title('Confusion matrix of the classifier')
fig.colorbar(cax)
ax.set_xticklabels([''] + labels)
ax.set_yticklabels([''] + labels)
plt.xlabel('Predicted')
plt.ylabel('True')
```

```
[[13  0  0]
 [ 0 15  4]
 [ 0  0 18]]
```



The short explanation of how to interpret a confusion matrix is as follows: The diagonal elements represent the number of points for which the predicted label is equal to the true label, while anything off the diagonal was mislabeled by the classifier. Therefore, the higher the diagonal values of the confusion matrix the better, indicating many correct predictions.

In our case, the classifier predicted all the 13 setosa and 18 virginica plants in the test data perfectly. However, it incorrectly classified 4 of the versicolor plants as virginica.

Logarithmic Loss

Logarithmic loss (logloss) measures the performance of a classification model where the prediction input is a probability value between 0 and 1. Log loss increases as the predicted probability diverges from the actual label. The goal of machine learning models is to minimize this value. As such, smaller logloss is better, with a perfect model having a log loss of 0.

```
#Classification
LogLoss
                import warnings
                import pandas
                from sklearn import model_selection
                from sklearn.linear_model import LogisticRegression
                from sklearn.metrics import log_loss


                warnings.filterwarnings('ignore')
                url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-
                indians-diabetes.data.csv"
                dataframe = pandas.read_csv(url)
                dat = dataframe.values
```

```
X = dat[:,:-1]
y = dat[:,-1]
seed = 7
#split data
X_train, X_test, y_train, y_test = model_selection.train_test_split(X,
y,test_size=test_size, random_state=seed)
model.fit(X_train, y_train)
#predict and compute logloss
pred = model.predict(X_test)
accuracy = log_loss(y_test, pred)
print("Logloss: %.2f" % (accuracy))
```

Area under Curve (AUC)

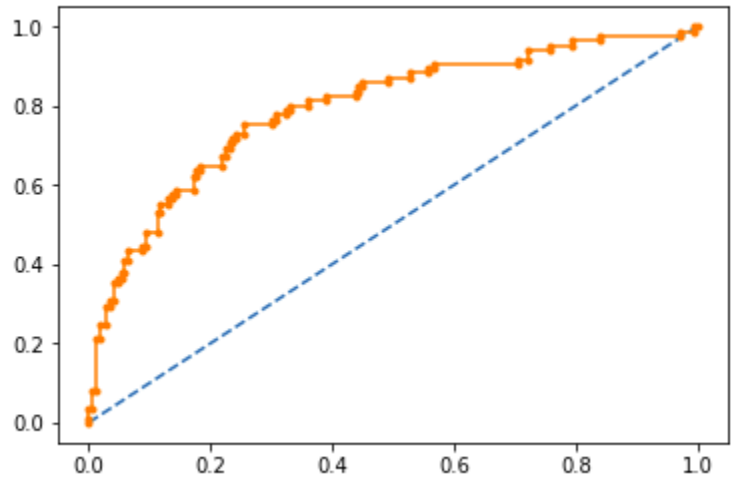
Area under ROC Curve is a performance metric for measuring the ability of a binary classifier to discriminate between positive and negative classes.

```
#Classification
Area under
curve
```

```
import warnings
import pandas
from sklearn import model_selection
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import roc_auc_score, roc_curve


warnings.filterwarnings('ignore')


url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-
indians-diabetes.data.csv"
dataframe = pandas.read_csv(url)
dat = dataframe.values
X = dat[:,:-1]
y = dat[:,-1]
seed = 7
#split data
X_train, X_test, y_train, y_test = model_selection.train_test_split(X, y,
test_size=test_size, random_state=seed)
model.fit(X_train, y_train)
```

```
# predict probabilities
probs = model.predict_proba(X_test)
# keep probabilities for the positive outcome only
probs = probs[:, 1]


auc = roc_auc_score(y_test, probs)
print('AUC - Test Set: %.2f%%' % (auc*100))


# calculate roc curve
fpr, tpr, thresholds = roc_curve(y_test, probs)
# plot no skill
plt.plot([0, 1], [0, 1], linestyle='--')
# plot the roc curve for the model
plt.plot(fpr, tpr, marker='.')
# show the plot
plt.show()
```

```
AUC - Test Set: 79.55%
```



In the example above, the AUC is relatively close to 1 and greater than 0.5. A perfect classifier will have the ROC curve go along the Y axis and then along the X axis.

F-Measure

F-measure (also F-score) is a measure of a test's accuracy that considers both the precision and the recall of the test to compute the score. Precision is the number of correct positive results divided by the total predicted positive observations. Recall, on the other hand, is the number of correct positive results divided by the number of all relevant samples (total actual positives).

Regression Matrix

In this section we review 2 of the most common metrics for evaluating regression problems namely, Root Mean Squared Error and Mean Absolute Error.

The Mean Absolute Error (or MAE) is the sum of the absolute differences between predictions and actual values. On the other hand, Root Mean Squared Error (RMSE) measures the average magnitude of the error by taking the square root of the average of squared differences between prediction and actual observation.

The Python code snippet below shows how the two regression metrics can be implemented.

```
from sklearn import model_selection
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error
from math import sqrt
url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/housing.data"
dataframe = pandas.read_csv(url, delim_whitespace=True)
df = dataframe.values
X = df[:,:-1]
y = df[:,-1]
seed = 7
model = LinearRegression()
#split data
X_train, X_test, y_train, y_test = model_selection.train_test_split(X, y,
test_size=test_size, random_state=seed)
model.fit(X_train, y_train)
#predict
pred = model.predict(X_test)
print("MAE test score:", mean_absolute_error(y_test, pred))
print("RMSE test score:", sqrt(mean_squared_error(y_test, pred)))
```

CONCLUSION

Ideally, the estimated performance of a model tells us how well it performs on unseen data. Making predictions on future data is often the main problem we want to solve. It's important to understand the context before choosing a metric because each machine learning model tries to solve a problem with a different objective using a different dataset.

DATA ANALYTIC TOOLS

Data analytics is the science of analyzing raw data in order to make conclusions about that information. Many of the techniques and processes of data analytics have been automated into mechanical processes and algorithms that work over raw data for human consumption.

Data analytics techniques can reveal trends and metrics that would otherwise be lost in the mass of information. This information can then be used to optimize processes to increase the overall efficiency of a business or system.

UNDERSTANDING DATA ANALYTICS

Data analytics is a broad term that encompasses many diverse types of data analysis. Any type of information can be subjected to data analytics techniques to get insight that can be used to improve things.

For example, manufacturing companies often record the runtime, downtime, and work queue for various machines and then analyze the data to better plan the workloads so the machines operate closer to peak capacity.

PYTHON

- Python was initially designed as an Object-Oriented Programming language for software and web development and later enhanced for data science. Python is the fastest-growing programming languages today.
- It is a powerful Data Analysis tool and has a great set of friendly libraries for any aspect of scientific computing.
- Python is free, open-source software, and it is easy to learn.
- Python's data analysis library Pandas was built over NumPy, which is one of the earliest libraries in Python for data science.

R PROGRAMMING

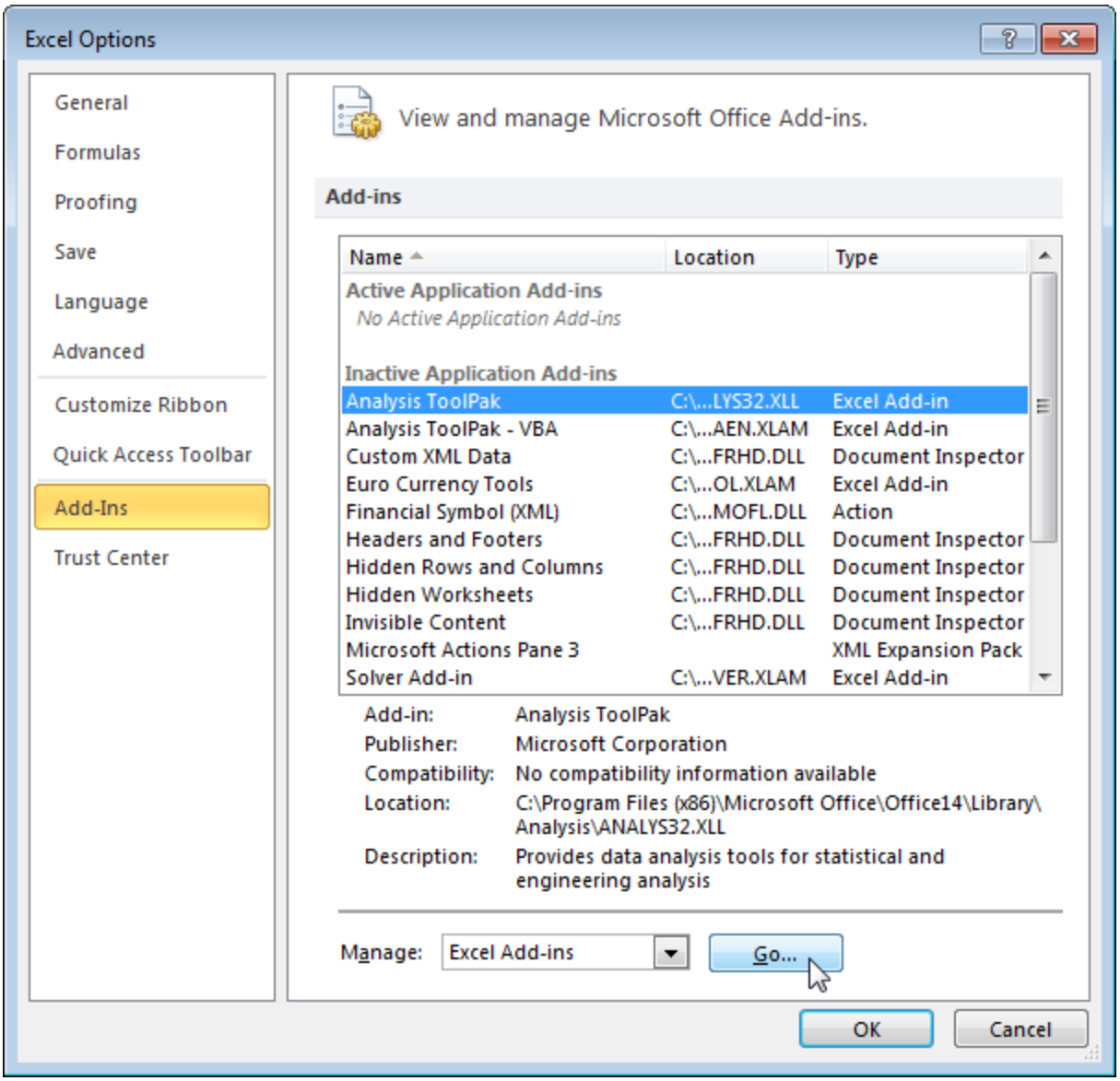
- R is the leading programming language for statistical modeling, visualization, and data analysis. It is majorly used by statisticians for statistical analysis, Big Data and machine learning.
- R is a free, open-source programming language and has a lot of enhancements to it in the form of user written packages
- R has a steep learning curve and needs some amount of working knowledge of coding. However, it is a great language when it comes to syntax and consistency.
- R is a winner when it comes to EDA(By definition - In statistics, exploratory data analysis(EDA) is an approach to analyzing data sets to summarize their main characteristics, often with visual methods).
- Data manipulation in R is easy with packages such as plyr, dplyr, and tidy.
- R is excellent when it comes to data visualization and analysis with packages such as ggplot, lattice, ggvis, etc.
- R has a huge community of developers for support.

- R is used by
    - **Facebook -** For behavior analysis related to status updates and profile pictures.
    - **Google -** For advertising effectiveness and economic forecasting.
    - **Twitter -** For data visualization and semantic clustering
    - **Uber -** For statistical analysis

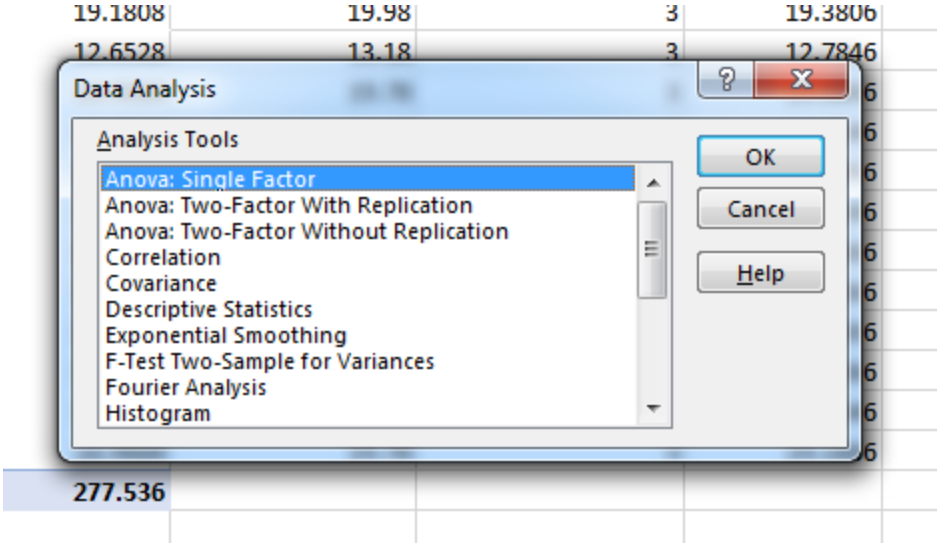
EXCEL

- Excel is a spreadsheet and a simple yet powerful tool for data collection and analysis.
- Excel is not free; it is a part of the Microsoft Office “suite” of programs.
- Excel does not need a UI to enter data; you can start right away.
- It is readily available, widely used and easy to learn and start on data analysis
- The Data Analysis Toolpak in Excel offers a variety of options to perform statistical analysis of your data. The charts and graphs in Excel give a clear interpretation and visualization of your data, which helps in decision making as they are easy to understand.

The Analysis Toolpak feature needs to be enabled and configured in Excel, as shown.



Once the Toolpak has been set up, you will see the list of tools. You can choose the tool based on your goals and the information that you want to analyze.



- Excel is used by more than 750 million users across the world.

TABLEAU

- Tableau is a BI(Business Intelligence) tool developed for data analysts where one can visualize, analyze, and understand their data.
- Tableau is not free software, and the pricing varies as per different data needs
- It is easy to learn and deploy Tableau
- Tableau provides fast analytics; it can explore any type of data – spreadsheets, databases, data on Hadoop and cloud services
- It is easy to use as it has a powerful drag and drop features that anyone with an intuitive mind can handle.
- The data visualization with smart dashboards can be shared within seconds.
- Top companies that use Tableau are Amazon, Citibank, Barclays, LinkedIn, and many more.

APACHE SPARK

- Spark Is an integrated analytics engine for Big Data processing designed for developers, researchers, and data scientists.
- It is free, open-source and a wide range of developers contribute to its development
- It is a high-performance tool and works well for batch and streaming data.
- Learning Spark is easy, and you can use it interactively from the Scala, Python, R, and SQL shells too.
- Spark can run on any platform such as Hadoop, Apache Mesos, standalone, or in the cloud. It can access diverse data sources.
- Spark includes libraries such as

    - for SQL and structured data - SparkSQL
    - Machine learning - MLlib
    - Live dataStream processing - SparkStreaming
    - Graph analytics - GraphX.

- Uber, Slack, Shopify, and many other companies use Apache Spark for data analytics.

EXCEL TOOLS FOR DATA ANALYSIS

Excel is probably the most commonly used spreadsheet for PCs. Newly purchased computers often arrive with Excel already loaded. It is easily used to do a variety of calculations, includes a collection of statistical functions, and a Data Analysis ToolPak. As a result, if you suddenly find you need to do some statistical analysis, you may turn to it as the obvious choice. We decided to do some testing to see how well Excel would serve as a Data Analysis application.

To present the results, we will use a small example. The data for this example is fictitious. It was chosen to have two categorical and two continuous variables, so that we could test a variety of basic statistical techniques. Since almost all real data sets have at least a few missing data points, and since the ability to deal with missing data correctly is one of the features that we take for granted in a statistical analysis package, we introduced two empty cells in the data:

Treatment	Outcome	X	Y
1	1	10.2	9.9
1	1	9.7	
2	1	10.4	10.2
1	2	9.8	9.7
2	1	10.3	10.1
1	2	9.6	9.4
2	1	10.6	10.3
1	2	9.9	9.5
2	2	10.1	10
2	2		10.2

Each row of the spreadsheet represents a subject. The first subject received Treatment 1, and had Outcome 1. X and Y are the values of two measurements on each subject. We were unable to get a measurement for Y on the second subject, or on X for the last subject, so these cells are blank. The subjects are entered in the order that the data became available, so the data is not ordered in any particular way.

We used this data to do some simple analyses and compared the results with a standard statistical package. The comparison considered the accuracy of the results as well as the ease with which the interface could be used for bigger data sets - i.e. more columns. We used SPSS as the standard, though any of the statistical packages OIT supports would do equally well for this purpose. In this article when we say "a statistical package," we mean SPSS, SAS, STATA, SYSTAT, or Minitab.

Most of Excel's statistical procedures are part of the Data Analysis tool pack, which is in the Tools menu. It includes a variety of choices including simple descriptive statistics, t-tests, correlations, 1 or 2-way analysis of variance, regression, etc. If you do not have a Data Analysis item on the Tools menu, you need to install the Data Analysis ToolPak. Search in Help for "Data Analysis Tools" for instructions on loading the ToolPak.

Two other Excel features are useful for certain analyses, but the Data Analysis tool pack is the only one that provides reasonably complete tests of statistical significance. Pivot Table in the Data menu can be used to generate summary tables of means, standard deviations, counts, etc. Also, you could use functions to generate some statistical measures, such as a correlation coefficient. Functions generate a single number, so using functions you will likely have to combine bits and pieces to get what you want. Even so, you may not be able to generate all the parts you need for a complete analysis.

DESCRIPTIVE STATISTICS

The quickest way to get means and standard deviations for a entire group is using Descriptives in the Data Analysis tools. You can choose several adjacent columns for the Input Range (in this case the X and Y columns), and each column is analyzed separately. The labels in the first row are used to label the output, and the empty cells are ignored. If you have more, non-adjacent columns you need to analyze, you will have to repeat the process for each group of contiguous columns. The procedure is straightforward, can manage many columns reasonably efficiently, and empty cells are treated properly.

To get the means and standard deviations of X and Y for each treatment group requires the use of Pivot Tables (unless you want to rearrange the data sheet to separate the two groups). After selecting the (contiguous) data range, in the Pivot Table Wizard's Layout option, drag Treatment to the Row variable area, and X to the Data area. Double click on "Count of X" in the Data area, and change it to Average. Drag X into the Data box again, and this time change Count to StdDev. Finally, drag X in one more time, leaving it as Count of X. This will give us the Average, standard deviation and number of observations in each treatment group for X. Do the same for Y, so we will get the average, standard deviation and number of observations for Y also. This will put a total of six items in the Data box (three for X and three for Y). As you can see, if you want to get a variety of descriptive statistics for several variables, the process will get tedious.

A statistical package lets you choose as many variables as you wish for descriptive statistics, whether or not they are contiguous. You can get the descriptive statistics for all the subjects together, or broken down by a categorical variable such as treatment. You can select the statistics you want to see once, and it will apply to all variables chosen.

CORRELATIONS

Using the Data Analysis tools, the dialog for correlations is much like the one for descriptives - you can choose several contiguous columns, and get an output matrix of all pairs of correlations. Empty cells are ignored appropriately. The output does NOT include the number of pairs of data points used to compute each correlation (which can vary, depending on where you have missing data), and does not indicate whether any of the correlations are statistically significant. If you want correlations on non-contiguous columns, you would either have to include the intervening columns, or copy the desired columns to a contiguous location.

A statistical package would permit you to choose non-contiguous columns for your correlations. The output would tell you how many pairs of data points were used to compute each correlation, and which correlations are statistically significant.

EXCEL DATA ANALYSIS TOOLPAK

The data analysis functions can be used on only one worksheet at a time. When you perform data analysis on grouped worksheets, results will appear on the first worksheet and empty formatted tables will appear on the remaining worksheets. To perform data analysis on the remainder of the worksheets, recalculate the analysis tool for each worksheet.

The Analysis ToolPak includes the tools described in the following sections. To access these tools, click Data Analysis in the Analysis group on the Data tab. If the Data Analysis command is not available, you need to load the Analysis ToolPak add-in program.

Data sets are configurable objects that store your data (think of them as Excel worksheets, or database tables, or tabular lists). You can give each data set a name, define its columns, set user permissions, and import data.

When you sign up, your account already has a number of pre-defined data sets, aimed at financial reporting. They contain example data of a fictitious company. You don't have to start from scratch, you have working examples which you can start with, and you can add or change data sets at any point in time.

Some common examples of data sets are:

- Financial data
- List of companies
- List of customers
- List of business units
- Chart of Accounts

Data sets don't need to exactly replicate the structure of your existing data, because you can convert your data during the import anyway. It is best to design them in such a way that they are optimal for reporting purposes.

COMPARING EXCEL BASED DATA WITH PYTHON DATA ANALYSIS

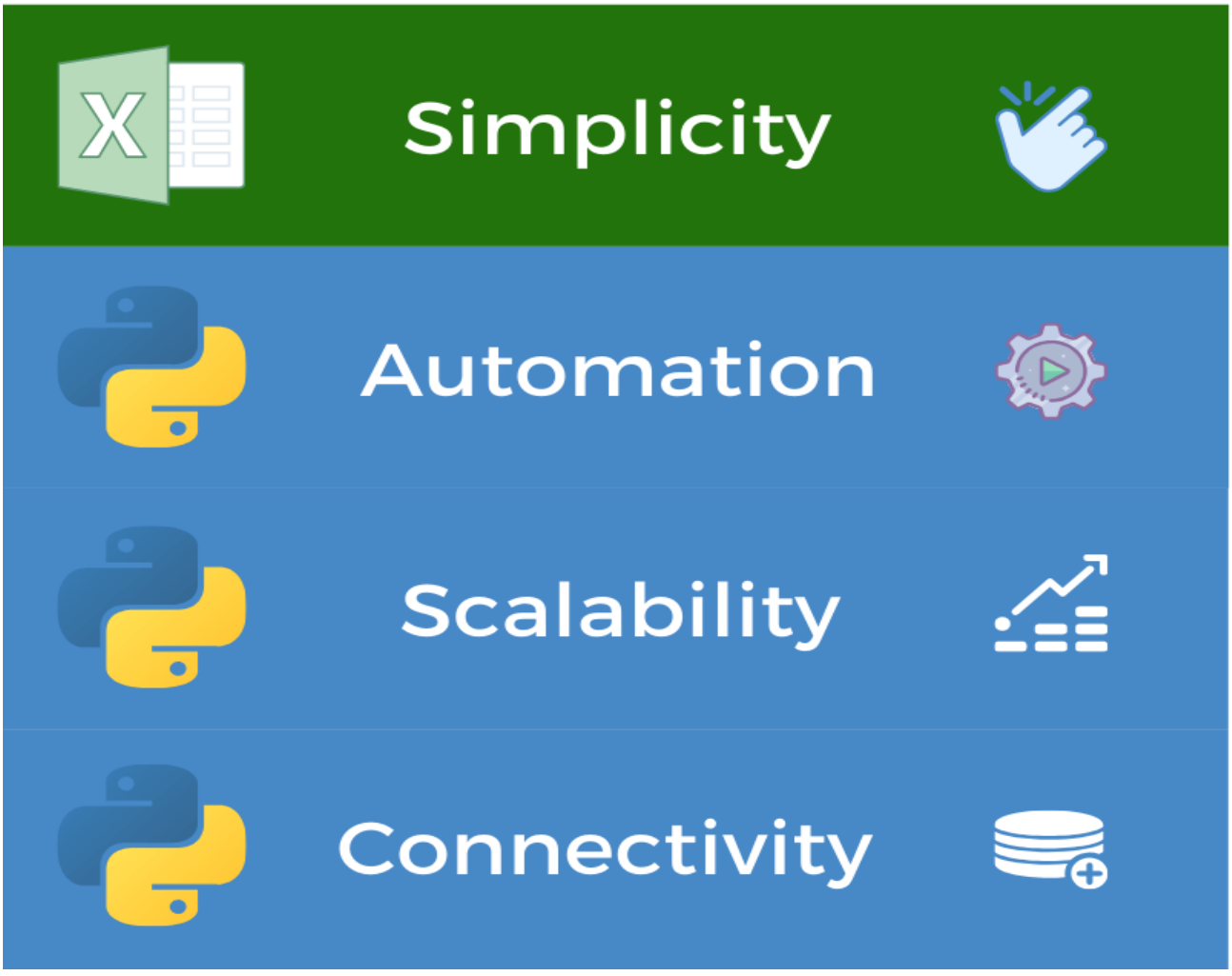
Excel spreadsheets are the standard in the business world for all kinds of data analysis tasks. While Excel’s simplicity makes it so commonplace, it also brings about some limitations. Python, on the other hand, is a programming language that is commonly used for data analysis and data science. We’ll go head-to-head on Python vs. Excel across a couple of important dimensions.

PYTHON

VERSUS

EXCEL

COMPARING THE 2 ANALYTICS TOOLS



REFRENCES:

https://www.edureka.co/blog/what-is-data-analytics/

https://www.guru99.com/what-is-data-analysis.html#:~:text=Data%20analysis%20is%20defined%20as,based%20upon%20the%20data%20analysis.

https://en.wikipedia.org/wiki/Python_(programming_language)

https://www.edureka.co/blog/what-is-data-science/

https://www.zarantech.com/blog/importance-of-data-science/

https://towardsdatascience.com/top-10-python-libraries-for-data-science-cd82294ec266

https://www.northeastern.edu/graduate/blog/data-analytics-vs-data-science/

https://www.sisense.com/blog/data-science-vs-data-analytics/#:~:text=Data%20analysis%20works%20better%20when,answers%20to%20questions%20being%20asked.

https://www.trifacta.com/data-wrangling/

https://en.wikipedia.org/wiki/Data_wrangling

https://en.wikipedia.org/wiki/Data_pre-processing

https://www.bmc.com/blogs/data-normalization/

https://www.geeksforgeeks.org/max-min-python/

https://www.geeksforgeeks.org/python-how-and-where-to-apply-feature-scaling/

https://www.geeksforgeeks.org/z-score-for-outlier-detection-python/#:~:text=Z%20score%20is%20also%20called,x%20%2Dmean)%20%2F%20std.

https://pbpython.com/pandas-qcut-cut.html

https://towardsdatascience.com/exploratory-data-analysis-in-python-c9a77dfa39ce

https://heartbeat.fritz.ai/introduction-to-machine-learning-model-evaluation-fa859e1b2d7f

https://people.umass.edu/evagold/excel.html

https://support.microsoft.com/en-us/office/use-the-analysis-toolpak-to-perform-complex-data-analysis-6c67ccf0-f4a9-487c-8dec-bdb5a2cefab6

https://www.nobledesktop.com/learn/python/python-vs-excel