Data Wrangling using Python

Siddhartha Ghosh, Kandula Neha, Y Praveen Kumar

Abstract**:** ***The term Data Engineering did not get much popularity as the terminologies like Data Science or Data Analytics, mainly because the importance of this technique or concept is normally observed or experienced only during working with data or handling data or playing with data as a Data Scientist or Data Analyst. Though neither of these two, but as an academician and the urge to learn, while working with Python, this topic 'Data engineering' and one of its major sub topic or concept 'Data Wrangling' has drawn attention and this paper is a small step to explain the experience of handling data which uses Wrangling concept, using Python. So Data Wrangling, earlier referred to as Data Munging (when done by hand or manually), is the method of transforming and mapping data from one available data format into another format with the idea of making it more appropriate and important for a variety of relatedm purposes such as analytics. Data wrangling is the modern name used for data pre-processing rather Munging. The Python Library used for the research work shown here is called Pandas. Though the major Research Area is 'Application of Data Analytics on Academic Data using Python', this paper focuses on a small preliminary topic of the mentioned research work named Data wrangling using Python (Pandas Library).***

Index Terms**:** ***Data Engineering, Python, Data Wrangling***

I. INTRODUCTION

This paper starts with an overview of Data Engineering. It will then explain about the use of Python Libraries for executing one of the most important Data Engineering Task – called Data Wrangling.

Data Engineering: Data Engineering is the fabrication and architecting the infrastructure for data (Data can be read as Big Data). It is the collecting and gathering of data, storing it for future, doing real time and batch processing on it and finally provide service to the Data Analyst/Scientist group for further process. Big Data tools are common names in Data Engineering field. The traditional Data Base concepts and Data Base Management Systems stand the fundamentals for Data engineering field.

So Data engineering is responsible for making the channel or streamline for the seamless movement of data from one instance to another. The data engineers who are into it take care of hardware and software requirement along with the IT and Data security and protection aissues. They also promise the fault tolerance in the system and monitor the server logs and administration of the data pipeline.

Data Engineering field includes handling and input errors, taking care of the system, making human-fault-tolerant pipelines, understanding what is necessary to make it better in size, solving continuous integration, knowledge of database administration, doing data cleaning, making a deterministic pipeline and finally gives a strong base to the Data Analytics or Data Scientist group.

Few Data Engineering Techniques: Data Engineering Techniques can be divided under numerous areas, such as File formats

- ✓ Wrangling
- ✓ Ingestion Machines
- ✓ Stream Processing
- ✓ Storage Machines
- ✓ Batch Processing, Batch SQL
- ✓ Storages for Data
- ✓ Management of Clusters
- ✓ Database Transaction
- ✓ Frameworks for Web
- ✓ Visualizations of Data
- ✓ Machine Learning.

Data Engineering and Data Analytics: The Data Analytics or Data Science Techniques cannot be applied on any kind of data set if the data is not in a proper format, data is not cleaned and data is not error free. So Data Engineers play the major role of representing data in a proper shape to a Data Analyst or Data Scientist.

Data wrangling: Data wrangling is the process of reshaping, aggregating, separating, or else we can name it as transforming data from one format to a more useful one.

Clean and wrangle data into a state which can be useful : Data engineers make sure the data the company is using is clean, reliable, and it is made for whatever the purpose we may use to present them. Data engineers mainly rangle data into a situation that can then have queries run against it by software developers.

Data wrangling is considering a scattered and unclear source of data and make it into an useful interesting data set which will catch many eyes. People may ask : How best are they as data set? How much usefulness they have towards the target? Do we have a better way to get data? Once one has thoroughly checked , collected and cleaned the data so that the collected data sets becomes important, we can utilize different AI, ML tools and methods (like Shell scripts) to analyze them and present the details to the developers. So it is important to collect proper data set and make them code ready or machine ready. Data wrangling is a interesting problem when working with big data, mainly if one learned to do it, or he doesn't have the right tools to clean and validate data in an effective and efficient way. Always a nice data engineer can understand the queries a data scientist is trying to understand and make their work easier by creating a interesting, on time, usable data product.

Revised Version Manuscript Received on 16 September, 2019.

Dr. Siddhrtha Ghosh, Professor, CSE Dept of Vidya Jyothi Institute of Technology.

Kandula Neha, Assistant Professor, CSE Dept of Vidya Jyothi Institute of technology.

Praveen Kumar Yechuri. Assistant Professor, CSE Dept of Vidya Jyothi Institute of Technology.

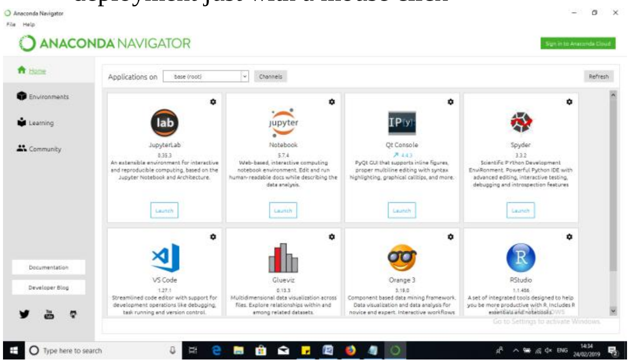
II. THE WORKING ENVIRONMENT

This research work uses the following tools for experiencing Data Wrangling steps.

- Python 3.5
- Anaconda3
- Jupyter Notebook
- Pandas Library

Anaconda3 : As we know Anaconda is a free and open-source tool of the Python and R programming languages for scientific, intelligent computing (data science, artificial intelligence and machine learning applications, big data processing, predictive analytics, etc.). It makes the package management and implementation easy. Anaconda is easy to use and needs machine with 8 GB RAM for best experience. It provides all most all the tools needed to work with pythin and give the best result. Anaconda provides the tools needed to easily:

- It takes data input from CSV files, Excel sheets, databases, and big data
- It manages working environments with Conda part of the software
- It can share, collaborate on, and reproduce projects
- Once the project is ready anaconda make the deployment just with a mouse click



JupyterLab
0.35.3
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook
5.7.4
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Qt Console
4.4.3
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Spyder
3.3.2
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

VS Code
1.27.1
Streamlined code editor with support for development operations like debugging, task running and version control.

Glueviz
0.13.3
Multidimensional data visualization across files. Explore relationships within and among related datasets.

Orange 3
3.19.0
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows

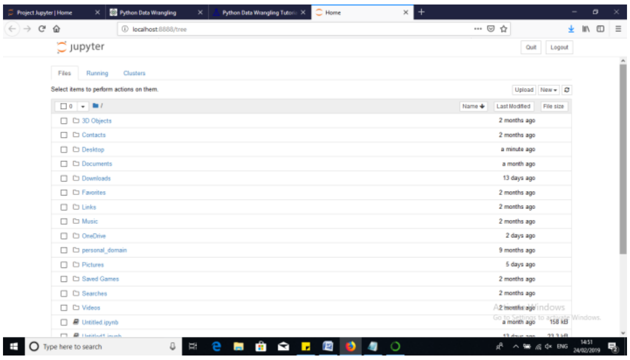
RStudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Fig 2: Anaconda Navigator

Anaconda creates an integrated, end-to-end data experience. This research work uses one important tool mentioned above called Jupyter Notebook.

Jupyter Notebook : Source - https://jupyter.org/

The Jupyter Notebook is an open-source tool that allows one to create and share software development documents that contain live code, equations, visualizations and narrative text. Uses include: cleaning of data and change from one form to another, numerical simulation, statistical modelling, visualization of data, machine learning, and much more. The total thing comes as a package item with Anaconda. Once the latest version of Anaconda is installed there's no need for seperately installing Jupyter Notebook. On launching the Jupyter Notebook the web browser looks like below given pic -



Select items to perform actions on them.

Name	Last Modified	File size
3D Objects	2 months ago	
Contacts	2 months ago	
Desktop	a minute ago	
Documents	a month ago	
Downloads	13 days ago	
Favorites	2 months ago	
Links	2 months ago	
Music	2 months ago	
OneDrive	2 days ago	
personal_domain	9 months ago	
Pictures	5 days ago	
Saved Games	2 months ago	
Searches	2 months ago	
Videos	2 months ago	
Untitled.ipynb	a month ago	158 kB

Fig 3: Jupyter Notebook through a Web Browser

The Notebook which is ueed here has facilty to work for over 40 programming languages, including Python, R, Julia, and Scala.

On choosing a new work environment for Python3 the screen looks like next fig.

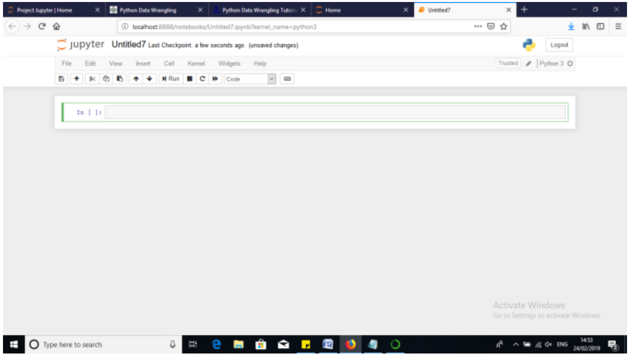


Fig 1: The Jupyter work Area for Python

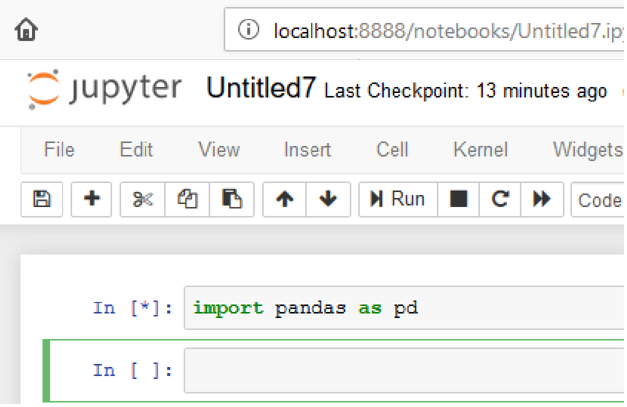
One need to write his/her code in the in [ ]: portion.

About Pandas in Python : Python is a great language for working on data analysis; primarily because of the fantastic ecosystem of data-centric Python packages. Pandas is one of those packages, which imports and analyzes data much easier. Pandas build on packages like NumPy and MatPlotLib to give a single, convenient place to work on most of data analysis and visualization.

Pandas Library features:

- ✓ DataFrame object for data handling and changing with indexing which is clubbed.
- ✓ Tools for reading and writing data between in-memory data structures and different file formats.
- ✓ Handling of mising data with proper integration
- ✓ Rearranging and pointing of data sets.
- ✓ Label-based slicing, fancy indexing, and sub setting of large data sets.
- ✓ Data structure column insertion and deletion.
- ✓ There is a Group by engine which allows split-apply-combine operations on data sets.
- ✓ Data set joining and merging.
- ✓ Hierarchical axis indexing to work with high-dimensional data in a lower-dimensional data structure.
- ✓ Time series-functionality: Date range generation [4] and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging.
- ✓ Provides data filtration.

The pandas is brought into action with a command on Jupyter Notebook as: import pandas as pd.


localhost:8888/notebooks/Untitled7.ip
jupyter Untitled7 Last Checkpoint: 13 minutes ago
File Edit View Insert Cell Kernel Widgets
Run
Code

```
In [*]: import pandas as pd

In [ ]:
```

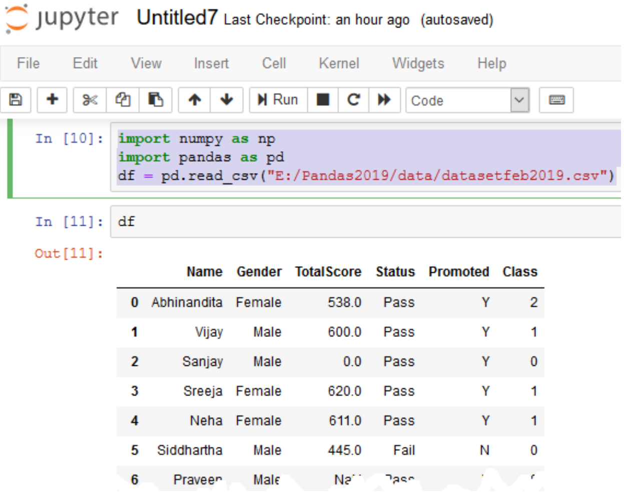
Fig 4: Launching Pandas on Jupyter Notebook

III. The Wrangling work using Pandas

As we know data wrangling involves techniques to bring the data in the data in various formats like - merging, grouping, concatenating etc. for the purpose of analysing or making them ready to be used with another set of data. Latest labguage Python has built-in features to apply these wrangling methods to different data sets to achieve the business goal. In this part of the paper few examples describing these methods will be looked into.

Data Sets and format: The Data Sets used here is mainly to mimic the academic data. The format used here is called CSV – Comma Separated Values. Anyone can make the same data sets using Microsoft Excel or Notepad and then saving the data set as .csv file. If Excel is used one shouldn't forget to close all sheets (other than one data sheet) before saving as .csv. Here a datasetfeb2019.csv is used which can be used in academic organization showing some result of a class. The file location path must be used to access the file.

Now on Jupyter Notebook NumPy library is also used for accessing data.


jupyter Untitled7 Last Checkpoint: an hour ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Run
Code

```
In [10]: import numpy as np
         import pandas as pd
         df = pd.read_csv("E:/Pandas2019/data/datasetfeb2019.csv")

In [11]: df

Out[11]:
```

	Name	Gender	TotalScore	Status	Promoted	Class
0	Abhinandita	Female	538.0	Pass	Y	2
1	Vijay	Male	600.0	Pass	Y	1
2	Sanjay	Male	0.0	Pass	Y	0
3	Sreeja	Female	620.0	Pass	Y	1
4	Neha	Female	611.0	Pass	Y	1
5	Siddhartha	Male	445.0	Fail	N	0
6	Praveen	Male	NaN	Pass	[illegible]	[illegible]

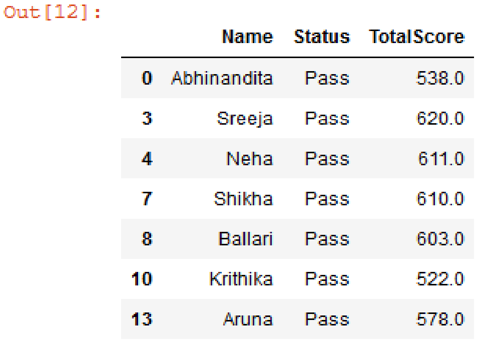
Fig 5: A portion of dataset on Jupyter Notebook

The command used to load the dataset mentioned above is –

```
import numpy as np
import pandas as pd
df =
pd.read_csv("E:/Pandas2019/data/datasetfeb2019.csv")
```

Boolean Indexing: here we find out that how are the values of a column filtered based on conditions from another set of columns? For instance, to find the value of a list of all females who scored above 500 means pass.

Python Code : df.loc[(df["Gender"]=="Female") & (df["TotalScore"]>=500), ["Name", "Status" , "TotalScore"]]



```
Out[12]:
```

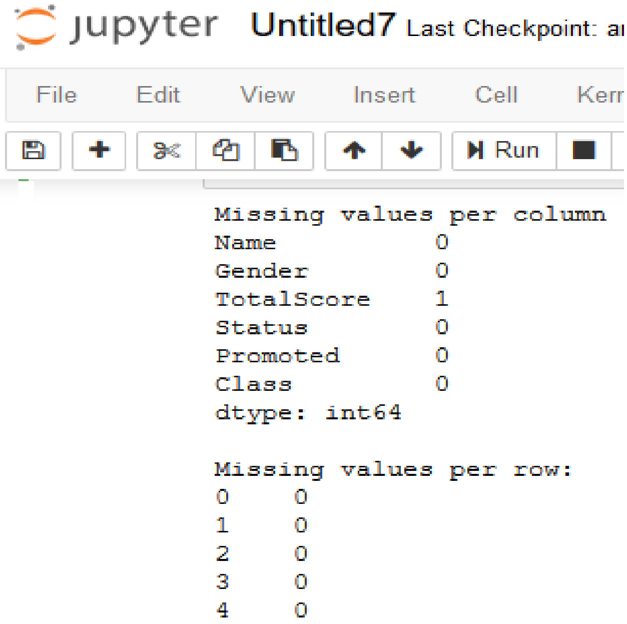
	Name	Status	TotalScore
0	Abhinandita	Pass	538.0
3	Sreeja	Pass	620.0
4	Neha	Pass	611.0
7	Shikha	Pass	610.0
8	Ballari	Pass	603.0
10	Krithika	Pass	522.0
13	Aruna	Pass	578.0

Fig 6: Outcome of the above mentioned Python Code

Apply Function: It is one of the commonly used functions in Pythn to handle data and making new variables. The method apply returns some value after sending each row/column of a data frame with some other function. The function can be both default or user-defined. For instance, here it can be used to find the #missing values in each row and column.

```
#Nnew function creation in Python :
def  n_miss(x):
return sum(x.isnull())
#Applying per column:
print ("Mmissing values per column")
print (df.apply(n_miss, axis=0))
#axis=0
 function is to be applied on each column
#Now applying per row:
print ("\nMissing values per row:")
print (df.apply(n_miss, axis=1).head())
```

#axis=1 defines that function is to be applied on each row

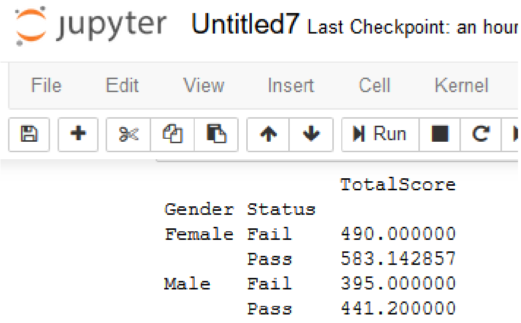

jupyter Untitled7 Last Checkpoint: ai
File Edit View Insert Cell Kerr
Run

```
Missing values per column
Name          0
Gender        0
TotalScore    1
Status        0
Promoted      0
Class         0
dtype: int64

Missing values per row:
0    0
1    0
2    0
3    0
4    0
```

Fig 5: Outcome of Finding Missing Values

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication



```
                     TotalScore
Gender Status
Female Fail          490.000000
       Pass          583.142857
Male   Fail          395.000000
       Pass          441.200000
```

Fig 7: A Pivot Table after Execution

Pivot Table: The Pandas can be used to make Excel style pointing tables. For instance, in this coding case, the one which we are doing, a key column is “Total Score” which has missing values. We can compute it using mean amount of each ‘Gender’ and ‘Status’ grp. Now the mean ‘TotalScoret’ of each group can be determined as:

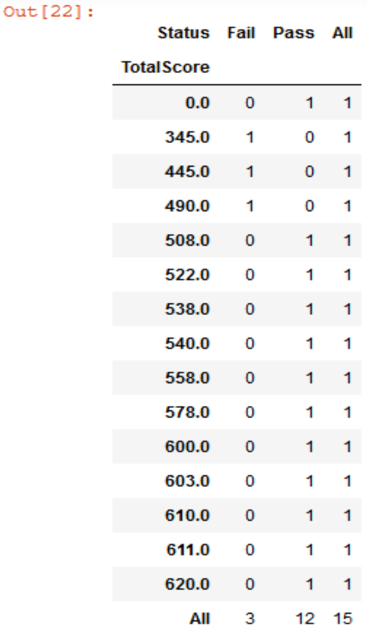
#Create pivot table

impute_grps = df.pivot_tab(values=["TotalSc"], index=["Gender","Status"], aggfunc=np.mean)

print (impute_grps)

Crosstab: This crosstab function is used to get an initial “feel” (view) of the data. Here, we can validate or check some basic hypothesis. For instance, in this case, “TotalScore” is expected to affect the “Status” significantly. The idea can be tested using cross-tabulation as shown in below figure:

pd.crosstab(df["TotalScore"],df["Status"],margins=True)



Out[22]:

Status	Fail	Pass	All
TotalScore			
0.0	0	1	1
345.0	1	0	1
445.0	1	0	1
490.0	1	0	1
508.0	0	1	1
522.0	0	1	1
538.0	0	1	1
540.0	0	1	1
558.0	0	1	1
578.0	0	1	1
600.0	0	1	1
603.0	0	1	1
610.0	0	1	1
611.0	0	1	1
620.0	0	1	1
All	3	12	15

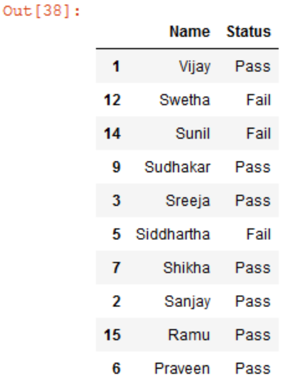
Now we will merge existing data frame df with N2

Sorting of DataFrames: The package Pandas allow us to do easy sorting and simplifying based on multiple columns or verticles. This can be done as:

To get the sorted values for required fields and to have the first 10 rows we can write -

data_sort = df.sort_vals(['Name','TotalScore'], ascending=False)

data_sort[['Name','Status']].head(10)



Out[38]:

	Name	Status
1	Vijay	Pass
12	Swetha	Fail
14	Sunil	Fail
9	Sudhakar	Pass
3	Sreeja	Pass
5	Siddhartha	Fail
7	Shikha	Pass
2	Sanjay	Pass
15	Ramu	Pass
6	Praveen	Pass

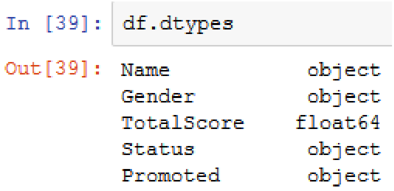
Fig8 : Data after Sorting

Iterating over rows of a dataframe means horizontal wise action : It is not a frequently used operation in Pandas. Still, one doesn’t want to get stuck, while working. At times one may need to iterate through all rows using a loop so we have a technique. For instance, one common problem we face is the incorrect treatment of variables in Python. This generally happens when:

- ✓ In this programming part the nominal variables with numeric categories are treated as numerical, intersting , right.
- ✓ Numeric variables with characters entered in one of the rows (due to a data error which may occur) are considered categorical.

So it’s generally a good idea to manually one defines the column types. If we check the data types of all columns then we should do as :

Finding Current Data Types:



```
In [39]: df.dtypes

Out[39]: Name           object
         Gender         object
         TotalScore    float64
         Status         object
         Promoted       object
```

A good way to handle such issues is to make a dot csv (.csv) file with column names and types. This way, we can make a common function to read the file and assign column data types.

So, there are many more steps and techniques which are found in Data Wrangling which makes the work of others easy. This paper discusses most of the common methods which are mandatory for the people who will work in the field of data Science or Data Analytics using Python.

IV. Conclusion

This paper was an initiative to share the preliminary steps of research experiences while working with Data Sets, Data Science and different Techniques. The paper is kept simple and small thinking that this can be used as preliminary steps for those thousands of learners and researchers who want to work in the field of Data Science and Machine Learning. A good time is spent by every individual, just thinking, where to start and what tools to use. This research work is an eye opener for me and while working with Pandas I could enjoy the modern ways of Analysing Data, mainly here, Wrangling data.

REFERENCES

[1] A. Gandomi and M. Haider, Beyond the hype: Big data concepts, methods, and analytics, International Journal of Information Management, 35(2) (2015), pp.137-144.

[2] C. L. Philip, Q. Chen and C. Y. Zhang, Data-intensive applications, challenges, techniques and technologies: A survey on big data, Information, Sciences, 275 (2014), pp.314-347.

[3] Fernando Pérez, Brian E. Granger, IPython: A System for Interactive Scientific Computing, Computing in Science and Engineering, vol. 9, no. 3, pp. 21-29, May/June 2007, doi: 10.1109/MCSE. 2007.53.

[4] MH. Kuo, T. Sahama, A. W. Kushniruk, E. M. Borycki and D. K. Grunwell, Health big data analytics: current perspectives, challenges and potential solutions, International Journal of Big Data Intelligence, 1 (2014), pp.114-126.

[5]M. K.Kakhani, S. Kakhani and S. R.Biradar, Research issues in big data analytics, International Journal of Application or Innovation in Engineering & Management, 2(8) (2015), pp.228-232.

[6] X. Jin, B. W.Wah, X. Cheng and Y. Wang, Significance and challengesof big data research, Big Data Research, 2(2) (2015), pp.59-64.

AUTHORS PROFILE

Dr. Siddhrtha Ghosh - Presently working as Professor in CSE Dept of Vidya Jyothi Institute of Technology. Has 18 years of teaching, 10 years of research and 2 years of industry experience. Has about 50 publications. Has own IBM USA Best innovative faculty award for Smart Cities in 2010. In 2018 he has been identified as Luminary of Qlik, Data Analytics company. In 2019 he has achieved best teacher award.

Kandula Neha - Presently working as Assistant Professor in CSE Dept of Vidya Jyothi Institute of technology. Has 2 years of experience in Research. Has about 11 publications.

Praveen Kumar Yechuri - Presently working as Assistant Professor in CSE Dept of Vidya Jyothi Institute of Technology. Has 9 years of Teaching Experience . Has about 8 publications.