1.INTRODUCTION

1.1 PYTHON

About Python:

- Python is a high-level, general-purpose, open source, strictly typed programming language. The language provides constructs intended to enable clear programs on both a small and large scale.
- Python was created By Guido van Rossum.
- The Python Software Foundation (PSF) is the organization behind Python.

Python versions:

- First released in 1991.
- Python 2.0 was released on 16 October 2000
- Python 3.0 was released on 3 December 2008

Current Versions:

- 3.6.3
- 2.7.14

Python features:

Some of the features of python include :-

- Easy to understand
- Dynamic
- Object oriented
- Multipurpose
- Strongly typed
- Open Sourced

Python is mainly used in many domains:

- Web Development
- Data Analysis
- Machine Learning
- Internet Of Things
- GUI Development
- Image processing
- Data visualization
- Game Development

IDLE:

IDLE is an integrated development environment for Python, which has been bundled with the default implementation of the language.

1.2 Anaconda

Anaconda is a open source Distribution for data science and machine learning using python. It includes hundreds of popular data science packages and the conda package and virtual environment manager for Windows, Linux, and MacOS. Conda makes it quick and easy to install, run, and upgrade complex data science and machine learning environments like scikit-learn, TensorFlow, and SciPy. Anaconda Distribution is the foundation of millions of data science projects as well as Amazon Web Service Machine Learning AMIs and Anaconda for Microsoft on Azure and Windows.

1.3 Packages

1.3.1 NumPy

NumPy is the fundamental package for scientific computing with Python. It

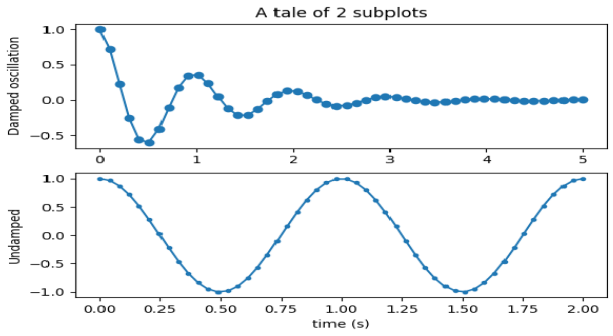
contains among other things:

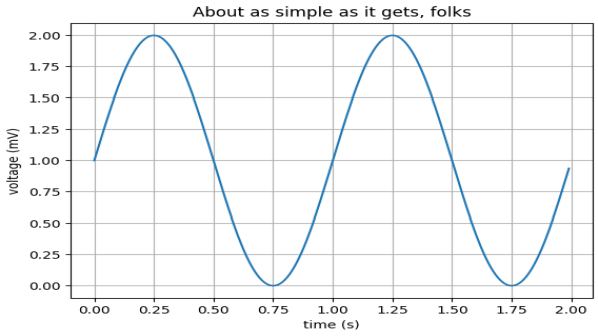
- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

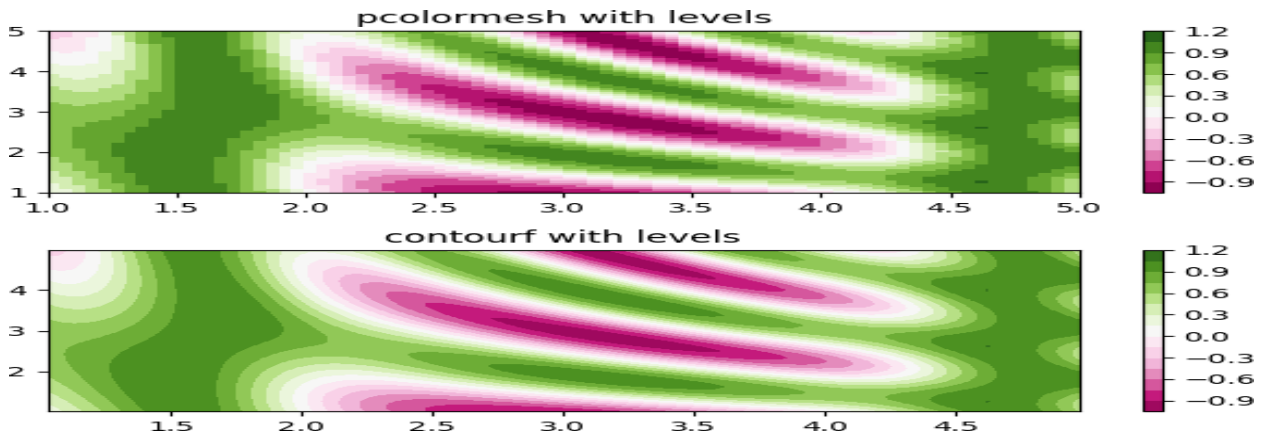
Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

1.3.2 Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, the jupyter notebook, web application servers, and four graphical user interface toolkits.







Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatterplots, etc., with just a few lines of code.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

1.3.3 Scikit-learn

Scikit-learn provides machine learning libraries for python.Some of the features of Scikit-learn includes:

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

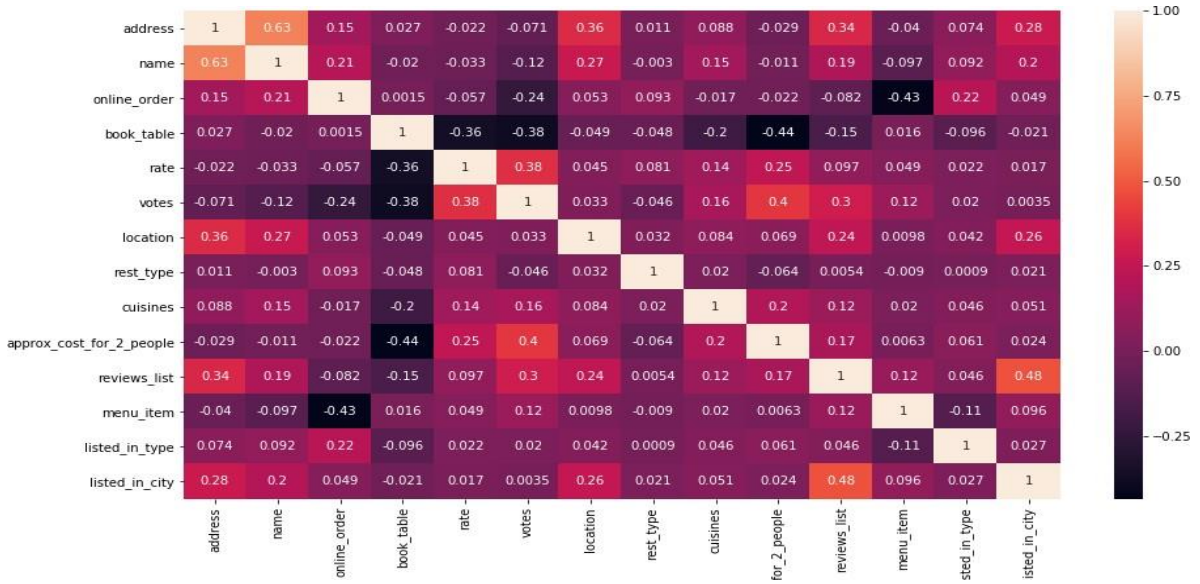
1.3.4 Pandas

Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

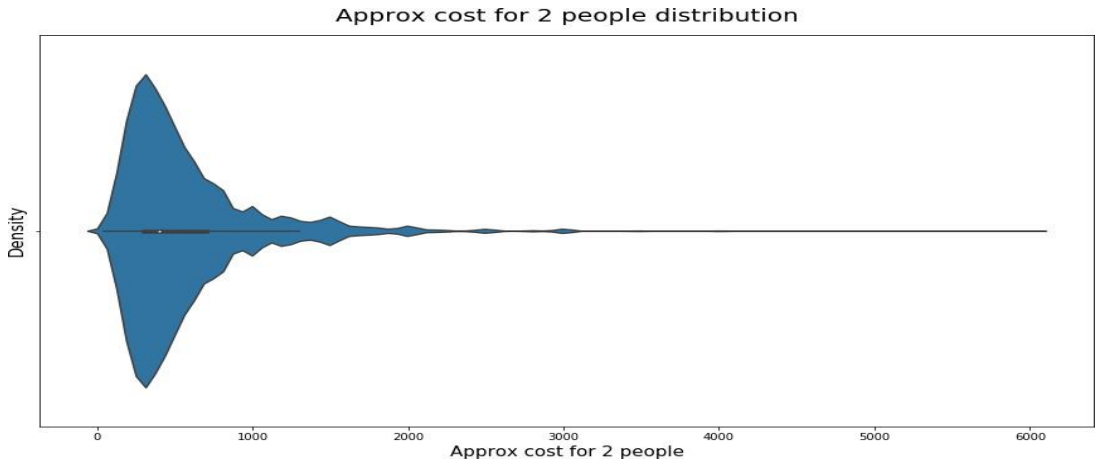
Pandas library is well suited for data manipulation and analysis using python. In particular, it offers data structures and operations for manipulating numerical tables
and time series.

1.3.5 Seaborn

Seaborn is a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics. E.g:-



Heatmap(above) & Violinplot(below)



2. TRAINING WORK UNDERTAKEN

2.1 COLLECTING DATA FROM KAGGLE

Kaggle is a platform for predictive modelling and analytics competitions in which statisticians and data miners compete to produce the best models for predicting and describing the datasets uploaded by companies and users. This crowd sourcing approach relies on the fact that there are countless strategies that can be applied to any predictive modelling task and it is impossible to know beforehand which technique or analyst will be most effective. On 8 March 2017, Google announced that they were acquiring Kaggle. They will join the Google Cloud team and continue to be a distinct brand. In January 2018, Booz Allen and Kaggle launched Data Science Bowl, a machine learning competition to analyze cell images and identify nuclei.

2.2 DATA SCIENCE

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from data in various forms, both structured and unstructured, similar to data mining. Data science is a "concept to unify statistics, data analysis, machine learning and their related methods" in order to "understand and analyze actual phenomena" with data. It employs techniques and theories drawn from many fields within the context of mathematics, statistics, information science, and computer science.

Turing award winner JiGray imagined data science as a "fourth paradigm" of science (empirical, theoretical, computational and now data-driven) and asserted that "everything about science is changing because of the impact of information technology" and the data deluge. When Harvard Business Review called it "The Sexiest Job of the 21st Century" the term became a buzzword, and is now often applied to business analytics, business intelligence, predictive modeling, or any arbitrary use of data, or used as a glamorized term for statistics. In many cases, earlier approaches and solutions are now simply rebranded as "data science" to be more attractive, which can cause the term to become "dilute[d] beyond usefulness." While many university programs now offer a data science degree, there exists no consensus on a definition or suitable curriculum contents. Because of the current popularity of this term, there are many "advocacy efforts" surrounding the field. To its discredit, however, many data science and big data projects fail to deliver useful results, often as a result of poor management and utilization of resources.

2.3 SOURCE CODE & OUTPUT

Import Packages

```
import numpy as np
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report,accuracy_score,confusion_matrix
```

Loading the Dataset

```
url = r'D:\Summer Training TISL ML\Project\Placement_Data_Full_Class.csv'  #reading the data from the csv file
plc = pd.read_csv(url)
plc.head()
```

Out[2]:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status	salary
0	1	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	Placed	270000.0
1	2	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed	200000.0
2	3	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	Placed	250000.0
3	4	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	Not Placed	NaN
4	5	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed	425000.0

Data Pre-processing

Finding the features with null values

```
plc.isnull().sum()
```

```
sl_no           0
gender          0
ssc_p           0
ssc_b           0
hsc_p           0
hsc_b           0
hsc_s           0
```

degree_p 0
degree_t 0
workex 0
etest_p 0
specialisation 0
mba_p 0
status 0
salary 67
dtype: int64

Removing the unnecessary columns

plc.drop(['sl_no','ssc_b','specialisation','mba_p','salary'],axis=1,inplace=True)

Out[6]:

	gender	ssc_p	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	status
0	M	67.00	91.00	Others	Commerce	58.00	Sci&Tech	No	55.0	Placed
1	M	79.33	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	Placed
2	M	65.00	68.00	Central	Arts	64.00	Comm&Mgmt	No	75.0	Placed
3	M	56.00	52.00	Central	Science	52.00	Sci&Tech	No	66.0	Not Placed
4	M	85.80	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.8	Placed

Assigning numerical values to the string literals

plc['gender'] = plc['gender'].map({'M':0,'F':1}) #assigning numerical values
plc['hsc_b'] = plc['hsc_b'].map({'Others':0,'Central':1})
plc['hsc_s'] = plc['hsc_s'].map({'Arts':0,'Commerce':1,'Science':2})

Out[7]:

	gender	ssc_p	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	status
0	0	67.00	91.00	0	1	58.00	Sci&Tech	No	55.0	Placed
1	0	79.33	78.33	0	2	77.48	Sci&Tech	Yes	86.5	Placed
2	0	65.00	68.00	1	0	64.00	Comm&Mgmt	No	75.0	Placed
3	0	56.00	52.00	1	2	52.00	Sci&Tech	No	66.0	Not Placed
4	0	85.80	73.60	1	1	73.30	Comm&Mgmt	No	96.8	Placed

Generating the range using the “cut” function

plc['hsc_pBand'] = pd.cut(plc['hsc_p'],5)

Out[16]:

	gender	ssc_p	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	status	hsc_pBand
0	0	67.00	91.00	0	1	58.00	1	0	55.0	1	(85.56, 97.7]
1	0	79.33	78.33	0	2	77.48	1	1	86.5	1	(73.42, 85.56]
2	0	65.00	68.00	1	0	64.00	0	0	75.0	1	(61.28, 73.42]
3	0	56.00	52.00	1	2	52.00	1	0	66.0	0	(49.14, 61.28]
4	0	85.80	73.60	1	1	73.30	0	0	96.8	1	(73.42, 85.56]

Checking the dependency of "hsc_pBand" on the placement status

```
plc[['hsc_pBand','status']].groupby(['hsc_pBand'],as_index=False).mean()
```

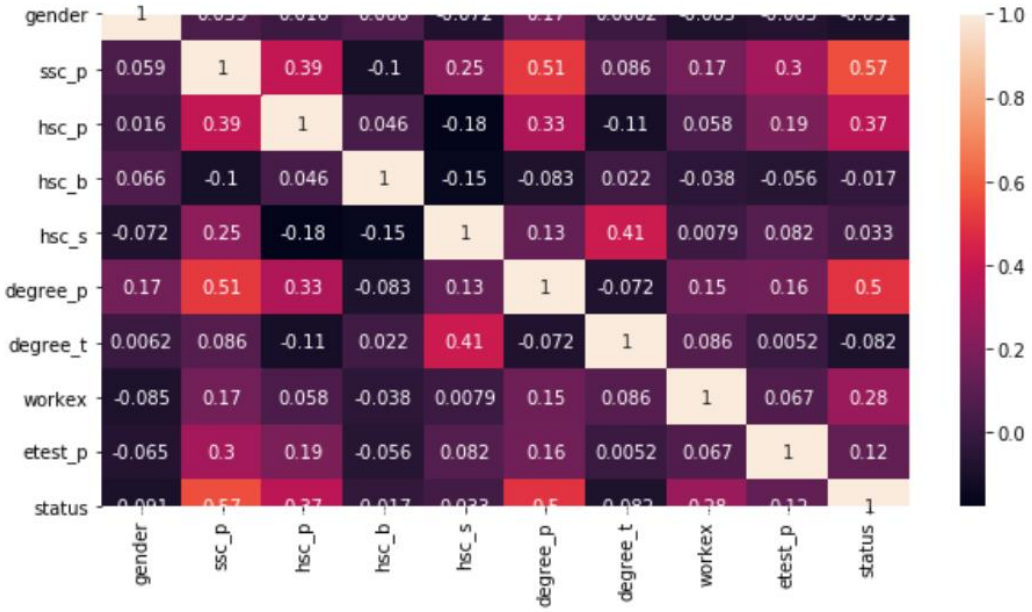
	hsc_pBand	status
0	(36.939, 49.14]	0.000000
1	(49.14, 61.28]	0.530612
2	(61.28, 73.42]	0.733333
3	(73.42, 85.56]	0.888889
4	(85.56, 97.7]	1.000000

Dividing the values of "hsc_p" into various ranges and assigning numerical values

```
plc.loc[ plc['hsc_p'] <= 40, 'hsc_p'] = 0
plc.loc[(plc['hsc_p'] > 40) & (plc['hsc_p'] <= 65), 'hsc_p'] = 1
plc.loc[(plc['hsc_p'] > 65) & (plc['hsc_p'] <= 80), 'hsc_p'] = 2
plc.loc[(plc['hsc_p'] > 80) & (plc['hsc_p'] <= 90), 'hsc_p'] = 3
plc.loc[ plc['hsc_p'] > 90, 'hsc_p'] = 4
```

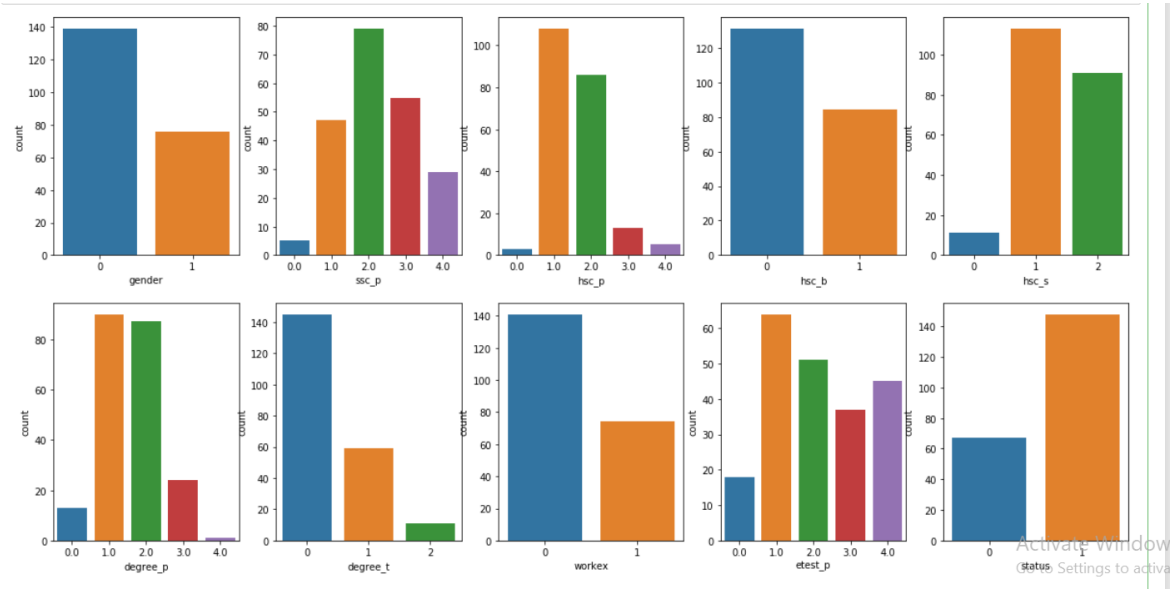
Generating the “heatmap”

```
fig, corr = plt.subplots(figsize=(10,5))
sb.heatmap(plc.corr(), annot=True)
plt.show()
```



Data Visualization

```
fig2, axs = plt.subplots(ncols=5, nrows=2, figsize=(20, 10))
index = 0
axs = axs.flatten() # to flaten to 1d
for k,v in plc.items():
    sb.countplot(x=v, data=plc, ax=axs[index])
    index += 1
```



Extracting and splitting of data

Extraction

X=plc.iloc[:,:9]
Y=plc.iloc[:,9]

Splitting

x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=.1,random_state=42)

Logistic Regression analysis

log=LogisticRegression()
log.fit(x_train,y_train)
y_pred=log.predict(x_test)
accuracy_score(y_pred,y_test)

0.8181818181818182

log.score(x_train,y_train)

0.8549222797927462

Generating the “confusion matrix” and “the classification report”

confusion_matrix(y_pred,y_test)

```
array([[ 5,  2],
       [ 2, 13]], dtype=int64)
```

print(classification_report(y_pred,y_test))

	precision	recall	f1-score	support
0	0.71	0.71	0.71	7
1	0.87	0.87	0.87	15
accuracy			0.82	22
macro avg	0.79	0.79	0.79	22
weighted avg	0.82	0.82	0.82	22

Result

The accuracy achieved using Logistic Regression is **82%** (accuracy score)

Discussion

The campus placement prediction aims to project a detail study on the campus recruitment of a handful of 200 students. The project achieves an accuracy of **82%** in detecting the placement status of a student . The model uses Logistic Regression approach in establishing the accuracy since the label is binary.

Not placed	0
Placed	1

Our data set consists of Placement data of students in our campus. It includes secondary and higher secondary school percentage and specialization. It also includes degree specialization, type and Work experience and salary offers to the placed students.

In the data wrangling part, we have assigned a numerical form to the string input for the complementary analysis since Machine Learning algorithms cannot work on string inputs.

We have discarded the Linear Regression algorithm since very few of the features have a linear relationship with the label.

Conclusion

This project can be easily implemented under various situations. We can add features as and when required. Reusability and flexibility can be exhibited in the modules. This project is extendable in ways that its original developers may not expect. The model enhances extensibility like Supervised Learning and binary classification. With the advent of upgraded version of the programming language we might be to reduce the code and simplify our understanding.

References

- https://www.kaggle.com/benroshan/factors-affecting-campus-placement
- https://seaborn.pydata.org/introduction.html#:~:text=Seaborn%20is%20a%20library%20for,examining%20relationships%20between%20multiple%20variables
- https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc
- Video lectures on Youtube

We would also like to express our gratitude to Prof. Mousita Dhar for guiding us in this project.