

GIRIJANANDA INSTITUTE OF MANAGEMENT AND TECHNOLOGY,GUWAHATI

**DATA SCIENCE USING PYTHON**

WEATHER PREDICTION

TRAINING PROJECT REPORT

**Submitted by:**

HIMASMITA BHARDWAJ

170310007022

Computer Science and Engineering

Bachelors of Technology

ACKNOWLEDGEMENT

It gives us great pleasure to acknowledge the guidance, assistance and support of Ms. Mousita Dhar in making the Project and this Project report successful, which has been structured under her valued suggestion.

She has helped us to accomplish the challenging task in a very short period of time.

Finally, we express the constant support of our friends, family and professors for inspiring us throughout and encouraging us.

Himasmita Bhardwaj

170310007022

CSE

INTRODUCTION

# 1.1Python

**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 isan 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:-



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.

* 1. SOURCE CODE & OUTPUT

**1.IMPORT PACKAGES**

1. **import numpy as np**

2. **import pandas as pd**

3. **import seaborn as sb**

4. **import matplotlib.pyplot as plt**

5. **import seaborn as sns**

6. **from sklearn.linear\_model import** LogisticRegression

7. **from sklearn.model\_selection import** train\_test\_split

8. **from sklearn.metrics import** classification\_report

9. **from sklearn.metrics import** confusion\_matrix

10. **from sklearn.metrics import** r2\_score

**2. LOAD THE DATASET**

11. weather=pd.read\_csv("E:\\my proj dataset\\weather.csv")

12. weather.head()

**3. DATA PRE-PROCESSING**

* Counting missing values for different columns

13. weather.isnull().sum()

Unnamed: 0 0

YEAR 0

JAN 0

FEB 0

MAR 0

APR 0

MAY 0

JUN 0

JUL 0

AUG 0

SEP 0

OCT 0

NOV 0

DEC 0

dtype: int64

* Rearranging my dataset to ‘YEARS’,’MONTHS’ and ‘TEMPERATURE’

14. weather\_new = pd.melt(weather, id\_vars='YEAR', value\_vars=weather.columns[1:])

weather\_new.head()

|  |  |  |
| --- | --- | --- |
| year | variable | Value |
| 1901 | JAN | 17.99 |
| 1902 | JAN | 18.02 |
| 1903 | JAN | 19.45 |
| 1904 | JAN | 17.23 |
| 1905 | JAN | 1717 |
|  |  |  |

* Changing my row name to ‘year’,’month’, and ‘temp’

15. weather\_new.columns=['Year','Month','Temp']

Weather\_new.head

|  |  |  |
| --- | --- | --- |
| Year | Month | Temp |
| 1901 | JAN | 17.99 |
| 1902 | JAN | 18.02 |
| 1903 | JAN | 19.45 |
| 1904 | JAN | 17.23 |
| 1905 | JAN | 1717 |
|  |  |  |

* Applying decision tree regression to my dataset

16. from sklearn.tree import DecisionTreeRegressor

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import r2\_score

weather\_new2 = weather\_new[['Year', 'Month', 'Temp']].copy()

weather\_new2 = pd.get\_dummies(weather\_new2)

y = weather\_new2[['Temp']]

x = weather\_new2.drop(columns='Temp')

dtr = DecisionTreeRegressor()

train\_x, test\_x, train\_y, test\_y = train\_test\_split(x,y,test\_size=0.3)

dtr.fit(train\_x, train\_y)

pred = dtr.predict(test\_x)

r2\_score(test\_y, pred)

OUTPUT: 0.9591824833797405

* Drawing a hitmap to see the correlation
* Using standardscaler to the dataset

from sklearn.preprocessing import StandardScaler

# Let's scale the columns before plotting them against YEAR

sc = StandardScaler()

column\_sels = ['Month\_DEC', 'Month\_FEB', 'Month\_JAN', 'Month\_JUN', 'Month\_MAY']

x = weather\_new2.loc[:,column\_sels]

y = weather\_new2['Temp']

data=sc.fit\_transform(x)

x = pd.DataFrame(data, columns=column\_sels)

X

* Applying Rondom Forest Regression to the dataset

from sklearn.ensemble import RandomForestRegressor

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import r2\_score

weather\_new2 = weather\_new[['Year', 'Month', 'Temp']].copy()

weather\_new2 = pd.get\_dummies(weather\_new2)

y = weather\_new2[['Temp']]

x = weather\_new2.drop(columns='Temp')

rfr = RandomForestRegressor(n\_estimators = 130, random\_state=0,min\_samples\_split=.15)

train\_x, test\_x, train\_y, test\_y = train\_test\_split(x,y,test\_size=0.3)

rfr.fit(train\_x, train\_y)

pred = rfr.predict(test\_x)

r2\_score(test\_y, pred)

OUTPUT: 0.9718500348916566

* Applying liner regression to the dataset

(as we have already imported the linear regression we need not to write it again)

lr=LinearRegression()

lr.fit(X\_train,y\_train)

OUTPUT: LinearRegression(copy\_X=True, fit\_intercept=True, n\_jobs=None,

normalize=False)

y\_pred=lr.predict(X\_test)

lr.coef\_

OUTPUT: array([-1.6919058 , -1.42180246, -1.88714066, 0.83347379, 0.87026434])

lr.intercept\_

OUTPUT: 24.277148836037448

* Finding rms value and r2\_square value

from sklearn.metrics import r2\_score, mean\_squared\_error

r2\_score(y\_test,y\_pred)

OUTPUT: 0.7977342267479862

lr.score(X\_train,y\_train)

OUTPUT: 0.7943308196583088

mean\_squared\_error(y\_test,y\_pred)

OUTPUT: 2.6621557192789957

RESULT

* Decision tree regression : 95%
* Random Forest Regression: 97%
* Linear regression: 79%

**SO FOR MY DATASET THE BEST RESULT IS GIVEN BY RANDOM TREE REGRESSION.**

CONCLUSION

* The main aim of this project was to find the temperature of a significant month of the year.
* This will help us in predicting the future weather of a month.
* We do that by doing training of the data that is already available to us
* This project involved three supervised learning algorithms i.e. **Linear Regression**, **Decision Tree Regression** and **Random Forest Regression.** All of them have different accuracy score. I got most score in Random Forest Regression but it may vary depending upon datasets.
* I conclude that my result was most accurate using the RANDOM FOREST REGRESSION(97%).

REFERENCES

https://www.kaggle.com/

https://www.python.org/

https://anaconda.org/anaconda/python/

http://www.numpy.org/

https://matplotlib.org/

http://scikit-learn.org/

https://pandas.pydata.org/

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 providea high-level interface for drawing attractive statistical graphics. E.g:-