**INTERNSHIP REPORT**

**On**

**Data Science**

**By**

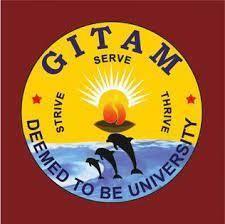
**Sameer A**

**Regd. No.: 322010337004**

**OASIS INFOBYTE**

**Bangalore.**

**(Duration: 01/04/2023 to 30/04/2023)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**Gandhi Institute of Technology and Management (DEEMED TO BE A UNIVERSITY) BENGALURU, KARNATAKA, INDIA SESSION:2020-2024**

### 

### ACKNOWLEDGEMENT

This is to express my sincerest gratitude for the wonderful opportunity I had as an intern in web development at OASIS INFOBYTE.

As my internship comes to an end, I wanted to take a moment to acknowledge and appreciate the invaluable experience I gained during my time with your esteemed organization.

Throughout my internship, I had the privilege of working with an exceptional team of professionals who were not only highly skilled but also incredibly supportive and encouraging.

Their mentorship and guidance were instrumental in my growth as a web developer.

The projects I was involved in challenged me to push my boundaries and allowed me to apply the theoretical knowledge I gained in a practical setting.

In conclusion, I am immensely grateful for the opportunity to intern at OASIS INFOBYTE. The experience has not only deepened my passion for web development but has also equipped me with valuable skills and knowledge that will undoubtedly shape my future career.

Sameer A

322010337004

### Internship Objectives

* This Internship is generally thought of to be reserved for college students looking to gain experience in a particular field. However, a wide array of people can benefit from Training Internship in order to receive real world experience and develop their skills.
* An objective for this position should emphasize the skills you already possess in the area and your interest in learning more
* Internship is utilized in a number of different career fields, including architecture, engineering, healthcare, economics, advertising and many more.
* Some internships are used to allow individuals to perform scientific research while others are specifically designed to allow people to gain first-hand experience working.
* Utilizing internships is a great way to build your resume and develop skills that can be emphasized in your resume for future jobs. When you are applying for a Training Internship, make sure to highlight any special skills or talents that can make you stand apart from the rest of the applicants so that you have an improved chance of landing the position.
* OASIS INFOBYTE internship goal is **to help others to gain personal and professional skills in the area of Data Science**. The purpose is to provide a structure through which students may exercise their right to freely associate in pursuit of a common purpose or goal that enhances community at OASIS INFOBYTE.

**WEEKLY OVERVIEW OF INTERNSHIP ACTIVITIES**

**WEEK 1:**

Task-1: IRIS FLOWER CLASSIFICATION

**WEEK 2:**

Task-2: UNEMPLOYMENT ANALYSIS WITH PYTHON

**WEEK 3:**

TASK-3: CAR PRICE PREDICTION USING MACHINE LEARNING

**WEEK 4:**

Task-4: EMAIL SPAM DETECTION

**Requirements**

### Software:

* + Operating system: Windows 11.
  + Languages: Python
  + Other: Github

### 

### TECHNOLOGY

**Data Science and Machine Learning**

Data science is a field that studies data and how to extract meaning from it. It encompasses a wide range of skills, including statistics, machine learning, programming, and visualization. Data scientists use these skills to collect, clean, analyze, and interpret data in order to make better decisions.

Machine learning is a field of computer science that gives computers the ability to learn without being explicitly programmed. Machine learning algorithms are trained on data, and they can then be used to make predictions or decisions without being explicitly told how to do so. Some of the most common machine learning algorithms include linear regression, logistic regression, support vector machines, and decision trees.

**Overview of Data Science:**

Data science is a field that studies data and how to extract meaning from it. It is a multidisciplinary field that combines principles and practices from the fields of mathematics, statistics, artificial intelligence, and computer engineering to analyze large amounts of data.

Data scientists use these skills to collect, clean, analyze, and interpret data in order to make better decisions. They use statistical and machine learning techniques to identify patterns and trends in data, and they use visualization techniques to communicate their findings to others.

Data science is used in a wide variety of industries, including:

Finance - Data scientists are used to predict financial trends, manage risk, and optimize investment portfolios.

Healthcare - Data scientists are used to develop new treatments, diagnose diseases, and improve patient care.

Retail - Data scientists are used to personalize product recommendations, optimize marketing campaigns, and improve customer service.

Manufacturing - Data scientists are used to improve product quality, optimize production processes, and reduce costs.

Logistics - Data scientists are used to optimize transportation routes, manage inventory, and prevent fraud.

**Overview of Machine Learning:**

Machine learning is a type of artificial intelligence (AI) that allows software applications to become more accurate in predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

Here are some examples of how machine learning is used today:

Image recognition is used to identify objects in images. For example, self-driving cars use image recognition to identify other cars, pedestrians, and traffic signs.

Speech recognition is used to convert spoken words into text. For example, Amazon Echo uses speech recognition to allow users to control their devices with their voice.

Natural language processing is used to understand human language. For example, Google Translate uses natural language processing to translate text from one language to another.

Fraud detection is used to identify fraudulent transactions. For example, credit card companies use fraud detection algorithms to identify suspicious transactions.

Recommendation systems are used to recommend products or services to users. For example, Netflix uses a recommendation system to recommend movies to its users.

Machine learning is a rapidly growing field with many potential applications. As the amount of data that we generate continues to grow, the demand for machine learning algorithms will continue to increase.

Here are some of the different types of machine learning:

Supervised learning is the most common type of machine learning. In supervised learning, the algorithm is trained on a set of labelled data. The labels tell the algorithm what the output should be for each input. For example, if you are training an algorithm to classify images of cats and dogs, the labels would tell the algorithm that the image of a cat should be labelled "cat" and the image of a dog should be labelled "dog".

Unsupervised learning is used to find patterns in unlabelled data. Unsupervised learning algorithms do not have any labels to tell them what the output should be. Instead, they must learn the patterns in the data on their own. For example, an unsupervised learning algorithm could be used to cluster a group of images into different categories, even though the images are not labelled.

Semi-supervised learning is a hybrid of supervised and unsupervised learning. In semi-supervised learning, the algorithm is trained on a set of labelled data and a set of unlabelled data. The labelled data helps the algorithm to learn the patterns in the data, and the unlabelled data helps the algorithm to generalize to new data.

Reinforcement learning is used to train agents to take actions in an environment in order to maximize a reward. Reinforcement learning algorithms do not have any labelled data. Instead, they learn by trial and error. The agent is given a reward for taking actions that lead to a desired outcome, and it is penalized for taking actions that lead to an undesired outcome. Over time, the agent learns to take actions that maximize the reward.

**WEEK-1**

Task-1: IRIS FLOWER CLASSIFICATION

**CODE:**

**#import required libraries**

**import numpy as np**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**#loading dataset**

**df = pd.read\_csv('/content/Iris.csv')**

**df**

**#Display first 10 and last 8 rows of dataset**

**print("First eight rows of dataset are: ")**

**print(df.head(10))**

**print("\n")**

**print("Last six rows of dataset are: ")**

**print(df.tail(8))**

**#Summary statistics of Iris dataset**

**df.describe()**

**#info of dataset**

**df.info()**

**#To get no. of rows and columns in the dataset**

**print("Dimensions of dataset are: ",df.shape)**

**#To get category of each type of species**

**print("No.of flowers in each species",df.value\_counts("Species"))**

**# We will construct a count plot to see the count of each species sns.countplot(x='Species', data=df)**

**plt.show()**

**plt.hist(df['SepalLengthCm'],color='lavender')**

**plt.title("histogram")**

**plt.xlabel("Sepal Length")**

**plt.ylabel("frequency")**

**plt.hist(df['PetalLengthCm'],color = 'black')**

**plt.title("histogram")**

**plt.xlabel("Sepal Length")**

**plt.ylabel("frequency")**

**plt.hist(df['SepalWidthCm'],color='green')**

**plt.title("histogram")**

**plt.xlabel("Sepal Length")**

**plt.ylabel("frequency")**

**DATA MODELLING**

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

**x = df.drop(columns = ['Species'])**

**y = df['Species']**

**x\_train, x\_test, y\_train, y\_test = train\_test\_split(x,y, test\_size = 0.4) y\_test**

**Decision Tree : ID3 decision tree classifier**

**#Importing library for decision classifier**

**from sklearn.tree import DecisionTreeClassifier id3=DecisionTreeClassifier(criterion='entropy')**

**#Fit the data**

**k=id3.fit(x\_train,y\_train)**

**#predict the data**

**y\_pred=id3.predict(x\_test)**

**print(y\_pred)**

**#Confusion matrix**

**from sklearn.metrics**

**import confusion\_matrix,accuracy\_score,classification\_report cm=confusion\_matrix(y\_pred,y\_test)**

**plt.figure(figsize=(10,8))**

**sns.heatmap(cm,annot=True)**

**plt.xlabel('predicted-y')**

**plt.ylabel('actual-y')**

**plt.show()**

**#Accuracy score and model score**

**print(classification\_report(y\_pred,y\_test)) print('accuracy-score',accuracy\_score(y\_pred,y\_test))**

**print('Model score',id3.score(x\_test,y\_test))**

**X\_new = np.array([[3, 2, 1, 0.2, 2], [4.9, 2.2, 3.8, 1.1, 2.2], [5.3, 2.5, 4.6, 1.9,3.1]])**

**#Prediction of the species**

**prediction = id3.predict(X\_new)**

**print("Prediction of Species: {}".format(prediction))**

**Output:**

Refer GitHub links for full code and outputs. I created a repository. Navigate through below link:

[Task - 1 Iris Classification](https://github.com/GeezGod01/OIBSIP/blob/main/TASK-1/OIB_DS_Task_1.ipynb)

**WEEK-2**

Task-2: UNEMPLOYMENT ANALYSIS WITH PYTHON

**CODE:**

**!pip install plotly.express**

**#importing libraries**

**import numpy as np**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**import plotly.express as px**

**#loading Unemployment in India dataset**

**df = pd.read\_csv('/content/Unemployment in India.csv')**

**df**

**#Loading Unemployment rate upto 2020 dataset**

**df1 = pd.read\_csv('/content/Unemployment\_Rate\_upto\_11\_2020.csv')**

**df1**

**#First and last rows of Unemployment in India dataset**

**print("Rows from start are: ")**

**print(df.head(6))**

**print("\n")**

**print("Rows from bottom: ")**

**print(df.tail(8))**

**#First and last rows in the dataset "unemployment rate till 2020"**

**print("Rows from start are: ")**

**print(df1.head(6))**

**print("\n")**

**print("Rows from bottom: ")**

**print(df1.tail(8))**

**#Summary statistics of Unemployment in India dataset**

**print("Shape of the data set ",df.shape)**

**print("Size of the data set",df.size)**

**print("\n")**

**print("Info of the dataset \n",df.info)**

**print("\n")**

**print("Descriptive statistics of the dataset \n",df.describe)**

***#Summary statistics in the dataset "umemployment rate till 2020"***

**print("Shape of the data set ",df1.shape)**

**print("Size of the data set",df1.size)**

**print("\n")**

**print("Info of the dataset \n",df1.info)**

**print("\n")**

**print("Descriptive statistics of the dataset \n",df1.describe)**

**#Names of columns in both datasets**

**print("Column names in the dataset unemployment in India: \n",df.columns) print("\n \n")**

**print("Column names in the dataset unemployment rate till 2020: \n",df1.columns)**

**#To check if both the datasets are null or not**

**print(df.isnull())**

**print("\n")**

**print(df1.isnull())**

**#Value counts for dataset**

**print("For the dataset-Unemployment in India: ") print(df.isnull().value\_counts())**

**print("\n") print("For the dataset-Unemployment rate till 2020: ") print(df1.isnull().value\_counts())**

**fig = plt.figure(figsize = (10, 10))**

**sns.histplot(x=' Estimated Unemployment Rate (%)', data=df, kde=True, hue='Area')**

**plt.title('Unemployment according to Area')**

**plt.xlabel('Unemployment Rate')**

**plt.show()**

**fig = plt.figure(figsize = (5, 5))**

**sns.lineplot(y=' Estimated Unemployment Rate (%)', x=' Date', data=df) plt.title('Unemployment according to Date')**

**plt.xlabel('Date')**

**plt.xticks(rotation=90)**

**plt.ylabel('Unemployment Rate')**

**plt.show()**

**df1.columns**

**fig = plt.figure(figsize = (30, 15))**

**plt.scatter(df1[' Date'], df1[' Estimated Employed'])**

**plt.title('Unemployment according to Region')**

**plt.xlabel('Region')**

**plt.ylabel('Unemployment Rate')**

**plt.show()**

**fig = plt.figure(figsize = (30, 15))**

**sns.histplot(x=' Estimated Labour Participation Rate (%)', data=df, kde=True, hue='Area')**

**plt.title('Labour Participation according to Area')**

**plt.xlabel('Labour Participation Rate')**

**plt.show()**

**fig = plt.figure(figsize = (7, 7))**

**sns.lineplot(y=' Estimated Labour Participation Rate (%)', x=' Date', data=df)**

**plt.title('Labour Participation according to Date')**

**plt.xlabel('Date')**

**plt.xticks(rotation=90)**

**plt.ylabel('Labour Participation Rate')**

**plt.show()**

**y=df[' Estimated Unemployment Rate (%)']**

**x=df['Region'] plt\_1 = plt.figure(figsize=(10, 10))**

**plt.title('Umemployment Rate', fontweight='bold' ,fontsize=20) plt.xlabel("States",fontweight='bold',fontsize=20) plt.ylabel("Estimated Unemployment rate",fontweight='bold',fontsize=20) plt.xticks(rotation='vertical',fontsize=12)**

**sns.histplot(x, color='lavender')**

**fig = plt.figure(figsize = (9, 9))**

**plt.scatter(df1['Region.1'], df1[' Estimated Labour Participation Rate (%)']) plt.title('Labour Participation according to Region')**

**plt.xlabel('Region')**

**plt.ylabel('Labour Participation Rate')**

**plt.show()**

**sns.histplot(x=' Estimated Employed', data=df, kde=True, hue='Area') plt.title('Employment according to Area')**

**plt.xlabel('Employment Rate')**

**plt.show()**

**fig = plt.figure(figsize = (9, 9)) sns.lineplot(y=' Estimated Employed', x=' Date', data=df)**

**plt.title('Employment according to Date')**

**plt.xlabel('Date')**

**plt.xticks(rotation=90)**

**plt.ylabel('Employment Rate')**

**plt.show()**

**fig = plt.figure(figsize = (9, 9))**

**plt.scatter(df1['Region.1'],**

**df1[' Estimated Employed'])**

**plt.title('Employment according to Region')**

**plt.xlabel('Region')**

**plt.ylabel('Employment Rate')**

**plt.show()**

**#Now let’s have a look at the correlation between the features of this dataset:**

**plt.style.use('seaborn-whitegrid')**

**plt.figure(figsize=(12, 10))**

**sns.heatmap(df.corr())**

**plt.show()**

**#Now let’s have a look at the correlation between the features of this dataset:**

**plt.style.use('seaborn-whitegrid')**

**plt.figure(figsize=(12, 10))**

**sns.heatmap(df1.corr())**

**plt.show()**

**df.columns**

**#Now let’s see the unemployment rate according to different regions of India: plt.figure(figsize=(12, 11))**

**plt.title("Indian Unemployment")**

**sns.histplot(x="Estimated Unemployment Rate", hue="Region", data=df1) plt.show()**

**Output:**

Refer GitHub links for full code and outputs. I created a repository. Navigate through the link below .

[Task - 2 Unemployment Analysis](https://github.com/GeezGod01/OIBSIP/blob/main/TASK-2/OIB_DS_Task_2.ipynb)

**WEEK-3**

TASK-3: CAR PRICE PREDICTION USING MACHINE LEARNING

**CODE:**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

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

**from sklearn.linear\_model import LinearRegression 0**

**from sklearn.linear\_model import Lasso**

**from sklearn import metrics**

**df=pd.read\_csv('/content/CarPrice.csv')**

**df.head()**

**df.shape**

**df.info()**

**df.isnull().sum()**

**df=df.drop('car\_ID',axis=1)**

**df=df.drop('fuelsystem',axis=1)**

**df.head()**

**x=df.drop(['CarName','price'],axis=1)**

**y=df['price']**

**print(x)**

**print(y)**

**x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.2,random\_state=2)**

**LINEAR REGRESSION**

**model=LinearRegression()**

**model.fit(x\_train,y\_train)**

**train\_model=model.predict(x\_train)**

**error\_score = metrics.r2\_score(y\_train, train\_model)**

**print("R squared Error : ", error\_score)**

**plt.scatter(y\_train, train\_model)**

**plt.xlabel("Actual Price")**

**plt.ylabel("Predicted Price")**

**plt.title(" Actual Prices vs Predicted Prices")**

**plt.show()**

**test\_model=model.predict(x\_test)**

**error\_score = metrics.r2\_score(y\_test, test\_model)**

**print("R squared Error : ", error\_score)**

**plt.scatter(y\_test, test\_model)**

**plt.xlabel("Actual Price")**

**plt.ylabel("Predicted Price")**

**plt.title(" Actual Prices vs Predicted Prices")**

**plt.show()**

**LASSO REGRESSION**

**las\_model=Lasso()**

**las\_model.fit(x\_train,y\_train)**

**Lasso()**

**train\_model=las\_model.predict(x\_train)**

**error\_score = metrics.r2\_score(y\_train, train\_model)**

**print("R squared Error : ", error\_score)**

**plt.scatter(y\_train, train\_model)**

**plt.xlabel("Actual Price")**

**plt.ylabel("Predicted Price")**

**plt.title(" Actual Prices vs Predicted Prices")**

**plt.show()**

**test\_model = las\_model.predict(x\_test)**

**error\_score = metrics.r2\_score(y\_test, test\_model)**

**print("R squared Error : ", error\_score)**

**plt.scatter(y\_test, test\_model)**

**plt.xlabel("Actual Price")**

**plt.ylabel("Predicted Price")**

**plt.title(" Actual Prices vs Predicted Prices")**

**plt.show()**

**Output:**

Refer GitHub links for full code and outputs. I created a repository. Navigate through the link below.

[Task - 3 Car Price Prediction](https://github.com/GeezGod01/OIBSIP/blob/main/TASK-3/OIB_DS_Task_3.ipynb)

**WEEK-4**

Task-4: EMAIL SPAM DETECTION

**CODE:**

**import numpy as np**

**import pandas as pd**

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

**from sklearn.feature\_extraction.text import TfidfVectorizer**

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

**from sklearn.metrics import accuracy\_score**

**raw\_mail\_data = pd.read\_csv('/content/spam.csv')**

**raw\_mail\_data.head()**

**mail\_data = raw\_mail\_data.where((pd.notnull(raw\_mail\_data)),'')**

**mail\_data.shape**

**mail\_data.loc[mail\_data['Category'] == 'spam', 'Category',] = 0 mail\_data.loc[mail\_data['Category'] == 'ham', 'Category',] = 1**

**X = mail\_data['Message']**

**Y = mail\_data['Category']**

**print(X)**

**print(Y)**

**X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state=3)**

**print(X.shape) print(X\_train.shape) print(X\_test.shape)**

**feature\_extraction = TfidfVectorizer(min\_df = 1, stop\_words='english', lowercase='True')**

**X\_train\_features = feature\_extraction.fit\_transform(X\_train) X\_test\_features = feature\_extraction.transform(X\_test)**

**Y\_train = Y\_train.astype('int')**

**Y\_test = Y\_test.astype('int')**

**print(X\_train)**

**print(X\_train\_features)**

**model = LogisticRegression()**

**model.fit(X\_train\_features, Y\_train)**

**prediction\_on\_training\_data = model.predict(X\_train\_features) accuracy\_on\_training\_data = accuracy\_score(Y\_train, prediction\_on\_training\_data)**

**print('Accuracy on training data : ', accuracy\_on\_training\_data)**

**prediction\_on\_test\_data = model.predict(X\_test\_features) accuracy\_on\_test\_data = accuracy\_score(Y\_test, prediction\_on\_test\_data)**

**print('Accuracy on test data : ', accuracy\_on\_test\_data**

**input\_mail = ["I've been searching for the right words to thank you for this breather. I promise i wont take your help for granted and will fulfil my promise. You have been wonderful and a blessing at all times"] input\_data\_features = feature\_extraction.transform(input\_mail) prediction = model.predict(input\_data\_features)**

**print(prediction)**

**if (prediction[0]==1):**

**print('Ham mail')**

**else:**

**print('Spam mail')**

**Output:**

Refer GitHub links for full code and outputs. I created a repository. Navigate through the link below.

[**Task - 4 Spam Mail Detection**](https://github.com/GeezGod01/OIBSIP/blob/main/TASK-4/OIB_DS_Task_4.ipynb)

Verify Here

* **Linkedin Profile:**

[**Linkedin**](http://www.linkedin.com/in/sameer-asif-a6399a254)

**INTERNSHIP COMPLETION CERTIFICATE**

****