**YHILLS INTERNSHIP REPORT**

**SUBMITTED BY**

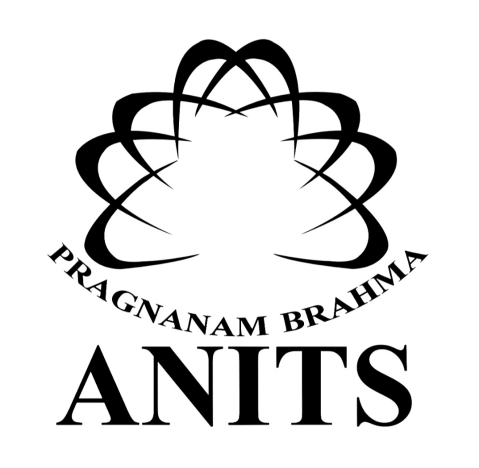
**PAILA TEJESWARA RAO (A21126551040)**

In fulfillment

of the internship

in

**COMPUTER SCIENCE & ENGINEERING(AI&ML,DS)**

****

**ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES**

**(Affiliated to Andhra University)**

**SANGIVALASA, VISAKHAPATNAM – 531162**

**2021-2025**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING(AI,ML&DS)**

**ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES**

**(Affiliated to Andhra University)**

**SANGIVALASA, VISAKHAPATNAM – 531162**

**2021-2025**

**BONAFIDE CERTIFICATE**

This is to certify that this Internship Report on “Data Science YHILLS” is a bonafide work of P TEJESWARA RAO (A21126551040) of 3/4 CSD carried out the Internship under my Supervision.

**Reviewer Class Teacher**

Dr. R NDSS KIRAN Mrs. G SURYAKALA ESWARI

Assistant Professor Assistant Professor

Department of CSE(AI&ML,DS) Department of CSE(AI&ML,DS)

**Prof. K. Selvani Deepthi**

Head of The Department

Department of CSE(AI&ML,DS)

ANITS

**ACKNOWLEDGEMENT**

The success of our long-term endeavor is attributed to the advice and support of numerous well-wishers. At this moment, we would like to convey our deep gratitude and appreciation to all those who contributed. Our sincere acknowledgments go to Prof. K. S. Deepthi, who served as the Head of the Department for Computer Science & Engineering (AI&ML,DS) at ANITS, for the invaluable support and guidance provided during our internship.

I wish to express MY sincere thanks and gratitude to our YHILLS Academy for smooth onboarding and analyzing the work associated with internship and for guiding me throughout the internenship. I express my warm and sincere thanks for the encouragement, untiring guidance, and confidence they had shown in me.

I also thank all the staff members of the Computer Science & Engineering (AI&ML,DS) department for their valuable advices. We also thank supporting staff for providing resources as and when required.

P TEJESWARA RAO

A21126551040

Table of Contents:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **About the Course Data Science** |  |  |  |  |  |  |  | 5 |
| What You Will Learn |  |  |  |  |  |  |  |  |
| Requirements |  |  |  |  |  |  |  |  |
| Description |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **Who is this course for?** |  |  |  |  |  |  |  | 5 |
| Data Science |  |  |  |  |  |  |  |  |
| Course Syllabus Course |  |  |  |  |  |  |  |  |
| Basic Concepts in Python |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **Data Analysis (Manipulation with NumPy and Pandas)** |  |  |  |  |  |  |  | 7 |
| NumPy |  |  |  |  |  |  |  |  |
| Pandas |  |  |  |  |  |  |  |  |
| Exploratory Data Visualization in Python with Matplotlib and Seaborn |  |  |  |  |  |  |  |  |
| Matplotlib |  |  |  |  |  |  |  |  |
| Seaborn |  |  |  |  |  |  |  |  |
| Statistical Thinking in Python |  |  |  |  |  |  |  |  |
| Measures of Dispersion Interquartile Range (IQR) |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **Supervised and Unsupervised Learning** |  |  |  |  |  |  |  | 15 |
| Supervised Learning |  |  |  |  |  |  |  |  |
| Unsupervised Learning |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **Machine Learning Models** |  |  |  |  |  |  |  | 16 |
| Linear Regression |  |  |  |  |  |  |  |  |
| Logistic Regression |  |  |  |  |  |  |  |  |
| Decision Trees Random Forest |  |  |  |  |  |  |  |  |
| Support Vector Machines (SVM) |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| Project |  |  |  |  |  |  |  | 17 |
|  |  |  |  |  |  |  |  |  |
| Certificate |  |  |  |  |  |  |  | 36 |

# ABOUT THE COURSE DATA SCIENCE:

**TITLE: DATA SCIENCE**

## **WHAT YOU WILL LEARN:**

• You'll receive a complete toolkit through this course, enabling your transformation into a data scientist.

• Enhance your resume with sought-after data science skills, including statistical analysis, Python programming using NumPy, pandas, matplotlib, and Seaborn, advanced statistical examination, Tableau, machine learning with statistical models and scikit-learn, and deep learning with TensorFlow.

• Impress surveys by demonstrating your grasp of the field of data science.

• Learn the art of data preprocessing.

• Gain a profound understanding of the science underpinning Machine Learning (an

essential aspect that many other courses omit!).

## **REQUIREMENTS:**

* No related knowledge is required. We will begin from the very essentials.
* You'll have to introduce Anaconda. We will tell you the best way to do that bit by bit.

## **Description:**

One of the most promising professions to thrive in this century is that of a data scientist. It is characterized by its focus on programming, innovation, and intelligence. Hence, it comes as no surprise that there has been a surge in demand for data scientists in the job market.

## **Who is this course for?:**

• If you're interested in becoming a Data Scientist or wish to explore the field, consider taking this course.

• If you're seeking a promising career, this course is tailored for you.

• This course is well-suited for beginners since it commences with the fundamentals and steadily enhances your skills.

## **Data Science Course Syllabus:**

**MODULE 1: BASICS OF PYTHON**

🡪datatypes

🡪variables

🡪data structures (list, tuple, set, dictionary)

**MODULE 2: DATA ANALYSIS**

🡪numpy

🡪pandas

**MODULE 3: EXPLORATORY DATA VISULIZATION IN PYTHON**

🡪 Matplotlib

🡪seaborn

**MODULE 4: STATISTICAL THINKING IN PYTHON**

🡪Mesure of central tendency

🡪Mesure of Dispersion

🡪 Interquartile Range (IQR)

**MODULE 5: SUPERVISED AND UNSUPERVISED LEARNING**

🡪Supervised learning

🡪unsupervised learning

**MODULE 6:MACHINE LEARNING**

🡪 Linear Regression

🡪Logestic regression

🡪SVM

🡪Random forest

🡪DescisionTree

🡪K-NN

🡪adabooster

**ASSESMENTS:**

After completion of course we submitted 2 projects.

**CERTIFICATION**:

1.By recognizing your commitment and effort in the program, we are pleased to present you with a course completion certificate from YHills.

2.Upon successfully completing the internship, we offer certificates from both YHills and prestigious universities.

3.As a reward, we furnish you with an industry certificate from well-established and esteemed companies.

# Course:

## BASIC CONCEPTS IN PYTHON:

* + - In Python, code blocks are defined through indentation, typically using four spaces. The start of an indented code block is indicated by colons (:), commonly seen in loops and functions.
  + Data storage is facilitated by variables, which you can create by assigning a value to a name, as demonstrated with x = 5. Python's dynamic typing eliminates the need to explicitly specify a variable's type.
  + Python incorporates various built-in data types, including integers (e.g., 5) and floats (e.g., 3.14). Additionally, it supports strings (e.g., "Hello"), lists (e.g., [1, 2, 3]), tuples (e.g., (1, 2, 3)), and dictionaries (e.g., {'name': 'Alice', 'age': 25}).
  + To display output in Python, you can use the print() function. For instance, you can execute print("Hello, World!").
  + Python's comment system involves the use of the # symbol. These comments are not processed by the Python interpreter and are primarily utilized for code documentation.
  + For decision-making within your code, employ if, elif, and else. For example:
  + python
  + Copy code
  + if x > 5:
  + print("x is greater than 5")
  + elif x == 5:
  + print("x is equal to 5")
  + else:
  + print("x is less than 5")
  + Sequences like lists can be iterated over using for loops, while while loops allow for ongoing execution as long as a condition remains true.
  + Functions in Python enable the encapsulation of code blocks for reusability. You define a function using the def keyword. Here's an example:
  + python
  + Copy code
  + def greet(name):
  + print(f"Hello, {name}!")
  + Lists serve as ordered collections of items, with elements accessible by their index, starting from 0.
  + Dictionaries, which store key-value pairs, allow access to values using their corresponding keys. They are commonly used for managing related data.
  + Python's extensive standard library includes various modules that can be imported to access pre-built functions and features. For instance, you can use import math to access mathematical functions.
  + File handling in Python entails reading from and writing to files using the open() function. Remember to close the file when you have finished working with it.
  + Exception handling is crucial for managing errors gracefully and preventing program crashes due to exceptions. You can achieve this using try, except, and finally blocks.
  + Python supports object-oriented programming (OOP) concepts, such as classes and objects, which assist in organizing and structuring code in a more modular manner.

## **DATA ANALYSIS(MANIPULATION WITH NUMPY AND PANDAS):**

### NUMPY:

NumPy is a fundamental package for scientific computing in Python. It provides support for arrays, matrices, and a wide range of mathematical functions to work with them.

1. In Python, code blocks are defined through indentation, typically using four spaces. The start of an indented code block is indicated by colons (:), commonly seen in loops and functions.

3.Data storage is facilitated by variables, which you can create by assigning a value to a name, as demonstrated with x = 5. Python's dynamic typing eliminates the need to explicitly specify a variable's type.

4.Python incorporates various built-in data types, including integers (e.g., 5) and floats (e.g., 3.14). Additionally, it supports strings (e.g., "Hello"), lists (e.g., [1, 2, 3]), tuples (e.g., (1, 2, 3)), and dictionaries (e.g., {'name': 'Alice', 'age': 25}).

5.To display output in Python, you can use the print() function. For instance, you can execute print("Hello, World!").

6.Python's comment system involves the use of the # symbol. These comments are not processed by the Python interpreter and are primarily utilized for code documentation.

7.For decision-making within your code, employ if, elif, and else. For example:

* + - Lists serve as ordered collections of items, with elements accessible by their index, starting from 0.
    - Dictionaries, which store key-value pairs, allow access to values using their corresponding keys. They are commonly used for managing related data.
    - Python's extensive standard library includes various modules that can be imported to access pre-built functions and features. For instance, you can use import math to access mathematical functions.
    - File handling in Python entails reading from and writing to files using the open() function. Remember to close the file when you have finished working with it.
    - Exception handling is crucial for managing errors gracefully and preventing program crashes due to exceptions. You can achieve this using try, except, and finally blocks.
    - Python supports object-oriented programming (OOP) concepts, such as classes and objects, which assist in organizing and structuring code in a more modular manner.

8.NumPy offers a variety of functions for array manipulation, including reshaping, stacking, splitting, and transposing, enabling users to work with arrays in different ways.

9.When it comes to Linear Algebra, NumPy provides built-in support for a range of operations such as matrix multiplication, determinant calculation, eigenvalue computation, and solving linear equations.

10.NumPy's random number generation module (numpy.random) allows users to generate random data or sample from various distributions, adding versatility to data generation tasks.

11.In Python, NumPy seamlessly integrates with other scientific and data analysis libraries like SciPy, Matplotlib, and Pandas, which makes it a critical component in the field of data science and scientific computing.

12.NumPy excels in memory management by efficiently handling memory and supporting data types that are optimized for numerical operations, thus reducing memory overhead.

13.The flexibility in data types provided by NumPy is essential for handling a wide range of data, including integers, floating-point numbers, complex numbers, and custom data types.

14.Being an open-source project, NumPy is both freely available and actively maintained by a dedicated community of developers, ensuring accessibility and continuous improvement

### PANDAS:

Pandas, a widely used open-source library for Python, is known for its strength in data manipulation and analysis. It offers user-friendly data structures and functions tailored for structured data, primarily through its two core data structures: DataFrames and Series.

**Data Structures**:

◦ A DataFrame is a structured data format with labeled rows and columns, resembling a spreadsheet or SQL table. Each column can accommodate varying data types.

◦ A Series, on the other hand, is akin to a DataFrame's column, possessing an associated label or index and structured as a one-dimensional array-like data format.

**Data Import/Export**:

* + Pandas can both import data from a variety of sources, such as CSV, Excel, SQL databases, and more, and also export data to these formats.

**Data Cleaning and Transformation**:

* + Pandas offers robust capabilities for data cleaning and preprocessing, which encompass tasks such as managing missing values, data filtering, and reshaping. Additionally, it enables data transformation and manipulation through functions like merging, joining, and pivoting.

**Indexing and Selection**:

* + You can select and filter data based on labels or conditions using methods like `.loc` and `.iloc`.
* Pandas enables the grouping of data based on specific criteria and executing aggregation functions on the grouped data.
* In the realm of time series data, Pandas boasts robust support. It provides functions for resampling, time-based indexing, and calculating moving window statistics.
* While Pandas is not a dedicated visualization library, it can effortlessly collaborate with visualization libraries like Matplotlib and Seaborn for data plotting.
* As it's built on top of NumPy, Pandas can seamlessly integrate with NumPy arrays to streamline data processing.
* Pandas offers performance optimization functions, including vectorized operations, which deliver significantly faster results compared to traditional for-loops.
* For identifying and managing missing or NaN values in data, Pandas provides a range of methods.
* It can handle reading and writing data in various file formats, simplifying the process of working with data from diverse sources.
* Thanks to its flexibility and user-friendliness, Pandas has gained wide adoption in the fields of data analysis, data science, and machine learning.
* Pandas boasts a thriving and extensive community, and numerous other data-related libraries and tools have been designed to seamlessly integrate with it.

## **EXPLORATORY DATA VISULIZATION IN PYTHON WITH MATPLOTLIB AND SEABORN**:

### Matplotlib:

Matplotlib, a versatile and powerful Python library for data visualization, offers a low-level interface to create a wide range of plots and charts. Let's delve deeper into Matplotlib for a more comprehensive understanding:

**Line Plot:**

* + A line plot is created using `plt.plot()`.
  + It's useful for visualizing trends or changes in data over time or continuous variables.
  + You can customize the line style, color, and markers.

**Scatter Plot**:

* + Scatter plots are used to display individual data points on a two-dimensional space.
  + They are great for showing the relationships and distributions of data points.
  + You can use `plt.scatter()` to create scatter plots.

**Bar Plot:**

* + Bar plots are useful for showing comparisons between different categories or groups.
  + They can be created using `plt.bar()` or `plt.barh()` for horizontal bar plots.

**Histogram:**

* + Histograms show the distribution of a single variable.
  + They are created with `plt.hist()` and are useful for visualizing data frequency.

**Box Plot:**

* + Box plots (box-and-whisker plots) display the distribution of a dataset's summary statistics.
  + They are created using `plt.boxplot()` and are helpful for identifying outliers and assessing data distributions.

### Seaborn:

Seaborn is built on top of Matplotlib and provides a high-level interface for creating aesthetically pleasing statistical visualizations. Here's a more detailed explanation:

**Heatmap:**

* + Heatmaps are used to visualize the relationships between two variables in a matrix format.
  + Seaborn's `sns.heatmap()` makes it easy to create heatmaps with color-coded values.

**Pair Plot**:

* + Pair plots, created using `sns.pairplot()`, are a grid of scatterplots for visualizing relationships between multiple variables in a dataset.
  + They are especially useful for understanding pairwise correlations.

**Violin Plot:**

* + Violin plots are a combination of box plots and kernel density estimation plots.
  + They show the distribution of data across different categories or groups.
  + `sns.violinplot()` is used to create them.

**Count Plot**:

* + Count plots are a type of bar plot that shows the count of observations in each category.
  + They can be generated with `sns.countplot()` and are great for categorical data.

**Facet Grid**:

* + Seaborn's `FacetGrid` allows you to create a grid of plots based on the structure of your dataset.
  + You can use it to visualize relationships within subgroups of your data.

In summary, Matplotlib is a low-level library that gives you full control over the customization of your plots, while Seaborn is a high-level library designed for quick and informative statistical visualization. The choice between them depends on your specific visualization needs, level of customization, and the type of plots you want to create.

## 

## **STATISTICAL THINKING IN PYTHON:**

Python's statistical thinking encompasses a range of concepts and techniques used to analyze and summarize data. The three fundamental concepts include the measures of central tendency, measures of dispersion, and the inter quartile range (IQR).

**Measures of Central Tendency:**

1. **Mean (Average):**
   * The most commonly used measure of central tendency is the mean, which is obtained by summing up all the data points and then dividing the sum by the total number of data points. To calculate the mean in Python, you can utilize the numpy library with the following code:

import numpy as np

mean = np.mean(data)

**2.Median:**

* + The median, when the data is sorted in ascending order, represents the middle value of a dataset. When there's an even number of data points, it is the average of the two middle values.
  + To compute the median with `numpy`:

python

median = np.median(data)

**3.Mode**:

* + The mode is the value that appears most frequently in the dataset, and a dataset can be either uni-modal with only one mode or multi modal with more than one mode, or it may have no mode at all.
  + You can calculate the mode using libraries like `statistics` or `scipy.stats`: python

from statistics import mode from scipy import stats

mode\_value = mode(data)

mode\_value = stats.mode(data)

### Measures of Dispersion:

**1.Range:**

* + The range measures the spread of data by finding the difference between the maximum and minimum values in the dataset.
  + To calculate the range:

python

data\_range = max(data) - min(data)

**2.Variance and Standard Deviation:**

* + Variance quantifies how data points deviate from the mean. A higher variance indicates greater spread.
  + Standard deviation is the square root of the variance and is a more interpretable measure.
  + You can compute variance and standard deviation using `numpy`: python

variance = np.var(data) std\_deviation = np.std(data)

### Interquartile Range (IQR):

### 

### -The IQR measures the spread of data within the middle 50% of the dataset, making it a robust measure that is not affected by extreme outliers.

* + To compute the IQR using `numpy`:

python

Q1 = np.percentile(data, 25) Q3 = np.percentile(data, 75) IQR = Q3 - Q1

* The IQR is particularly useful for identifying potential outliers and creating box plots. It defines the box

(representing the middle 50% of data, from Q1 to Q3) and the whiskers (commonly 1.5 times the IQR length).

* Values that fall below Q1 - 1.5 \* IQR or above Q3 + 1.5 \* IQR are often considered potential outliers.

In summary, measures of central tendency help you understand the typical or central values in your data, while measures of dispersion provide insights into how the data is spread out. The IQR, as a measure of dispersion, focuses on the middle 50% of the data and is useful for outlier detection. Python libraries like `numpy`,

`statistics`, and `scipy` make it easy to calculate these statistical measures for your datasets.

## SUPERVISED AND UNSUPERVISED LEARNING:

### Supervised:

Supervised learning is a type of machine learning where the algorithm learns from a labeled dataset. In a supervised learning problem, the dataset used for training consists of input-output pairs, where the output (target) is known. The goal is to learn a mapping from inputs to outputs so that the algorithm can make predictions or classifications on new, unseen data.

1. **Training Data**: In supervised learning, you start with a training dataset that includes input features and corresponding target values. For example, if you're building a spam email classifier, the input features could be the email's content, and the target values would be whether the email is spam or not (binary classification).
2. **Model Training**: You choose a machine learning model (e.g., linear regression, decision trees, neural networks) and train it on the training data. The model learns to make predictions based on the patterns it discovers in the input-output pairs.
3. **Prediction**: After the model is trained, you can use it to make predictions or classify new, unseen data. For example, if you have a new email, the model can predict whether it's spam or not based on the learned patterns.
4. **Evaluation**: You assess the model's performance by comparing its predictions to the true target values in a separate evaluation dataset. Common metrics for evaluation include accuracy, precision, recall, and F1-score, depending on the type of problem (classification or regression).
5. **Use Cases**: Supervised learning is used for various tasks, such as image classification, sentiment analysis, recommendation systems, and more. It's well-suited for problems where you have labeled data and want to make predictions or decisions.

**Unsupervised:**

Unsupervised learning, on the other hand, deals with datasets that have no labeled outputs. The primary goal of unsupervised learning is to find underlying patterns, structures, or relationships in the data without

guidance on what the output should be. It's about discovering hidden insights or grouping similar data points together.

1. **Data Clustering**: A common task in unsupervised learning is data clustering. Clustering algorithms group similar data points together based on their similarities or distances, creating clusters or groups. The most famous clustering algorithm is K-Means.
2. **Dimensionality Reduction**: Unsupervised learning also includes dimensionality reduction techniques, such as Principal Component Analysis (PCA) and t-SNE. These methods help reduce the complexity of data by capturing its essential information in a lower-dimensional space.
3. **Anomaly Detection**: Another application is anomaly detection, where the algorithm identifies unusual data points or outliers in the dataset. This is helpful for fraud detection or quality control.
4. **Generative Modeling**: Unsupervised learning can be used to create generative models, such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), which can generate new data points that resemble the training data.
5. **Use Cases**: Unsupervised learning is employed in various domains, including customer segmentation, data compression, and exploratory data analysis. It's particularly useful when you have large, unlabeled datasets and want to discover patterns or reduce data complexity.

supervised learning involves learning from labeled data to make predictions or classifications, while unsupervised learning is about discovering hidden patterns or structures in unlabeled data. Both paradigms play crucial roles in machine learning and have diverse applications across various domains.

## MACHINE LEARNING MODELS:

### Linear Regression:

* + Linear regression is used for regression tasks, where the goal is to predict a continuous numerical output based on input features.
  + The model assumes a linear relationship between the input features and the target variable. It finds the best-fit line that minimizes the sum of squared differences between predicted and actual values.

### Logistic Regression:

* + Logistic regression is primarily used for binary classification problems, where the output is a binary (two- class) label (e.g., 0 or 1).
  + It uses the logistic function to model the probability of the input belonging to one of the classes.

### Decision Trees:

* + Decision trees are versatile for both classification and regression. They create a tree-like structure where each node represents a decision based on input features.

### 

### 4.Random Forest:

* + Random Forest is an ensemble method that combines multiple decision trees to improve predictive accuracy and reduce overfitting.
  + It builds multiple decision trees and aggregates their predictions, often using a majority vote (for classification) or averaging (for regression).

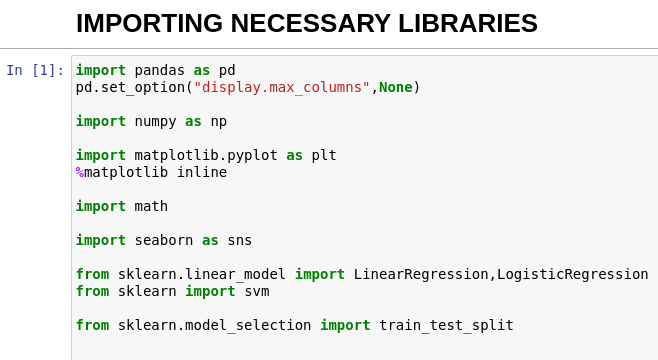
### Support Vector Machines (SVM):

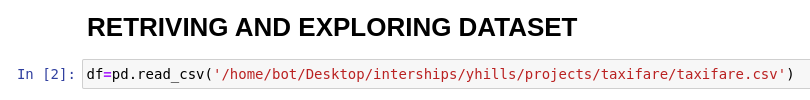
* + SVM is used for both classification and regression. It aims to find a hyperplane that best separates classes or predicts continuous values.
  + SVM seeks the optimal hyperplane by maximizing the margin between data points from different classes. It can use various kernel functions to handle non-linear data.

1. **K-Nearest Neighbors (K-NN):**
   * K-NN is a simple yet effective algorithm for both classification and regression. It makes predictions based on the majority class or average of the k-nearest data points in the feature space.
   * The choice of 'k' (number of neighbors) impacts the algorithm's performance and sensitivity to noise.
2. **Naive Bayes**:
   * Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. It is often used in text classification and spam filtering.
   * It assumes that features are conditionally independent given the class, simplifying the computation of probabilities.
3. **AdaBoost**:
   * AdaBoost is an ensemble learning method that combines multiple weak classifiers to create a strong classifier.
   * It assigns weights to data points and repeatedly trains models to correct the errors made by the previous models. The final model is a weighted sum of the weak models.

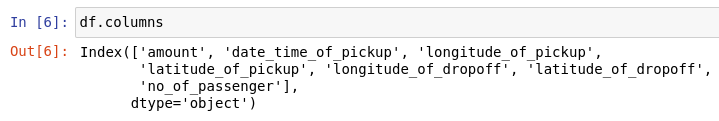
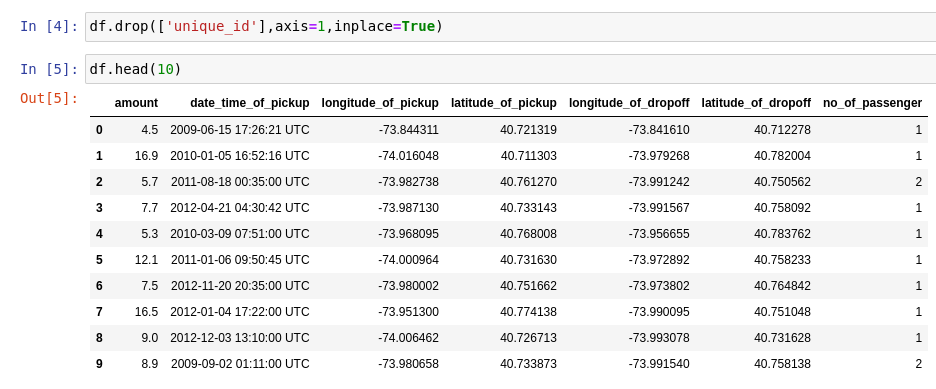
# PROJECT:

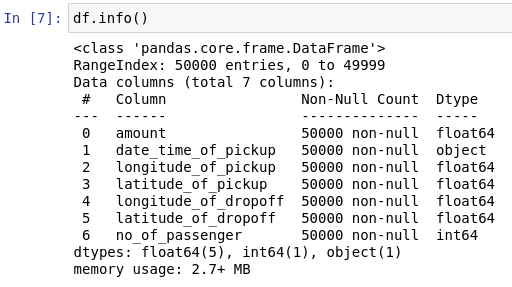
**TITLE : TAXI\_FARE**

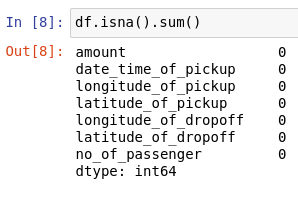
****

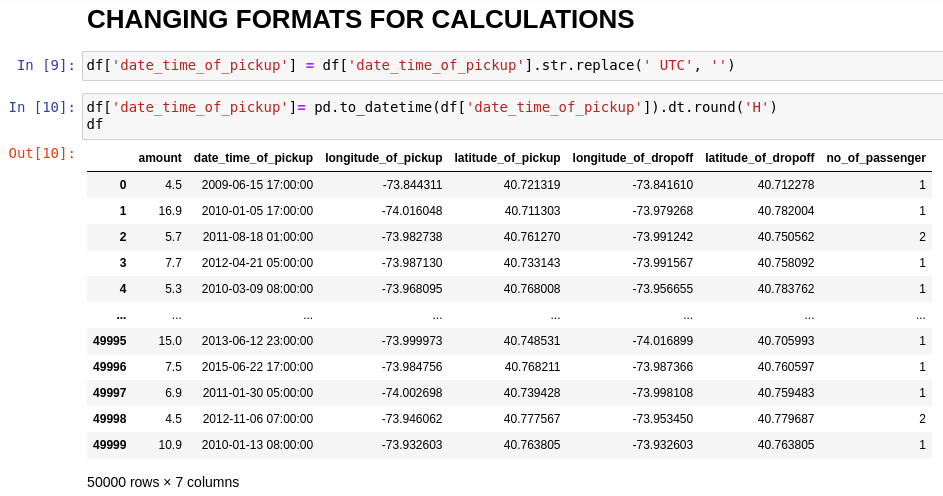
****

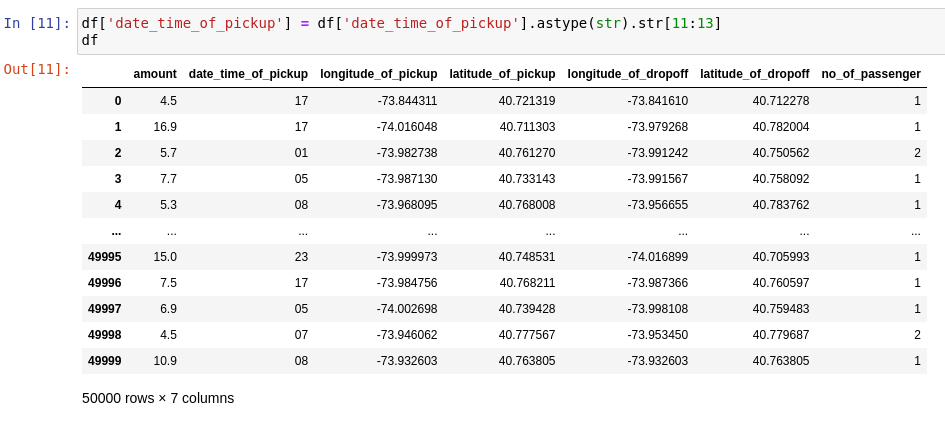
****

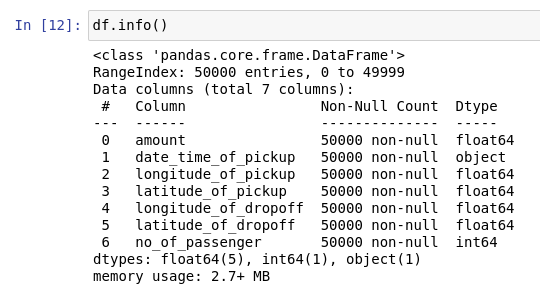
****

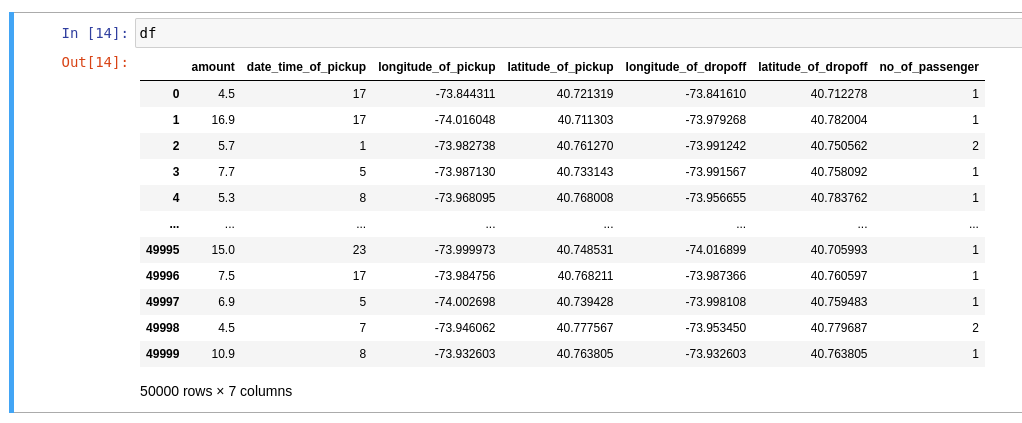
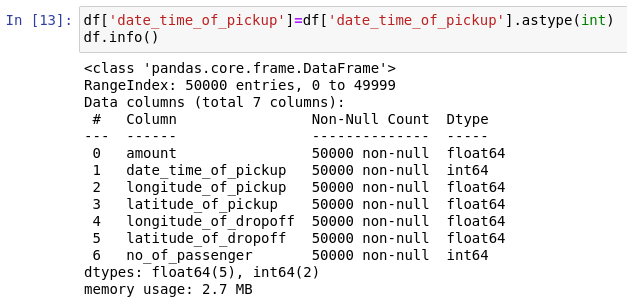
****

****



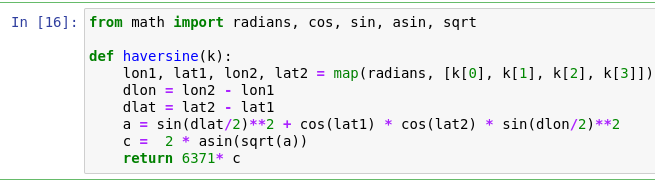






**METHODS TO CHANGE TO THOSE GIVEN LATITUDE AND LONGITUDES**

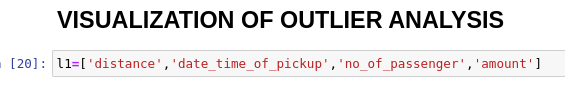
**TO DISTANCE:**

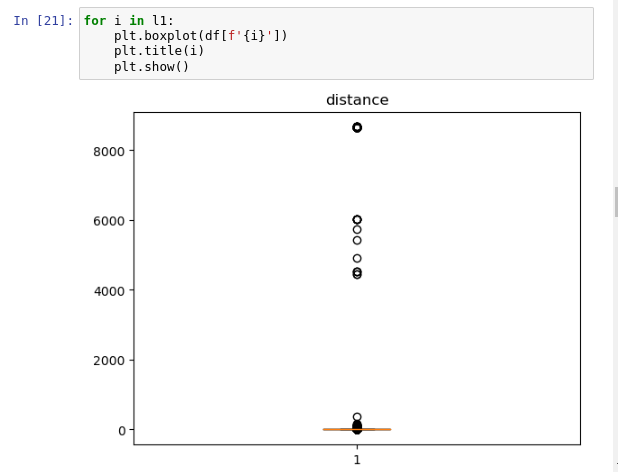


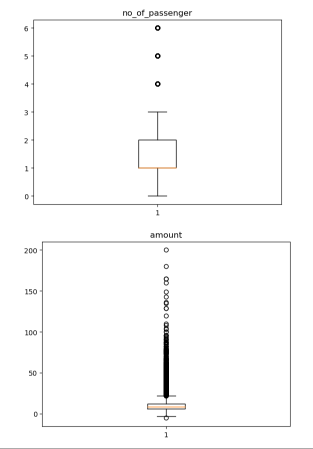


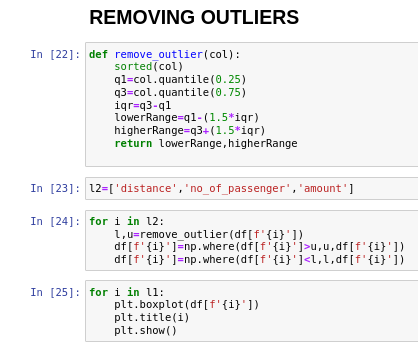


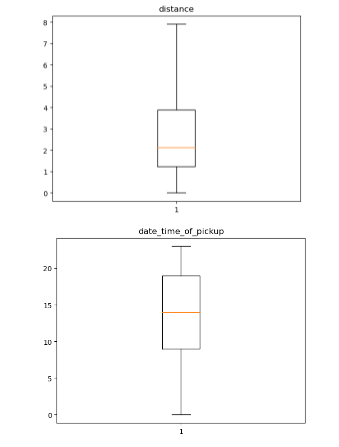
* DROP THE COORDINATES AS WE GOT DISTANCE

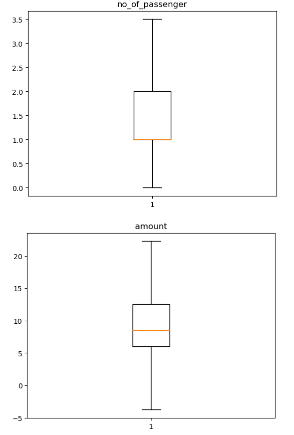


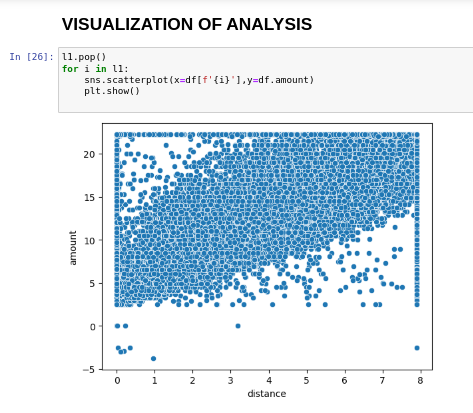


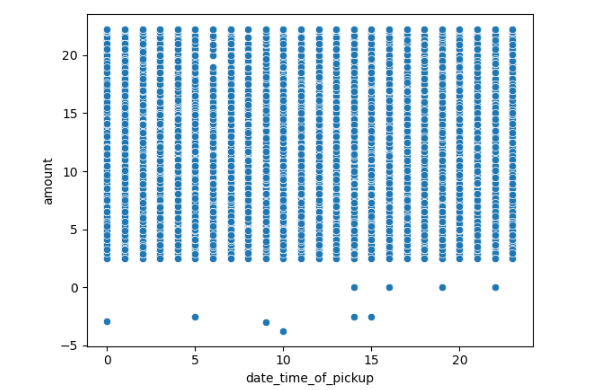


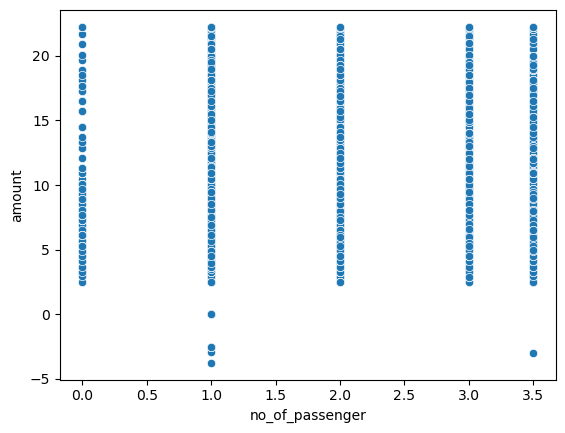


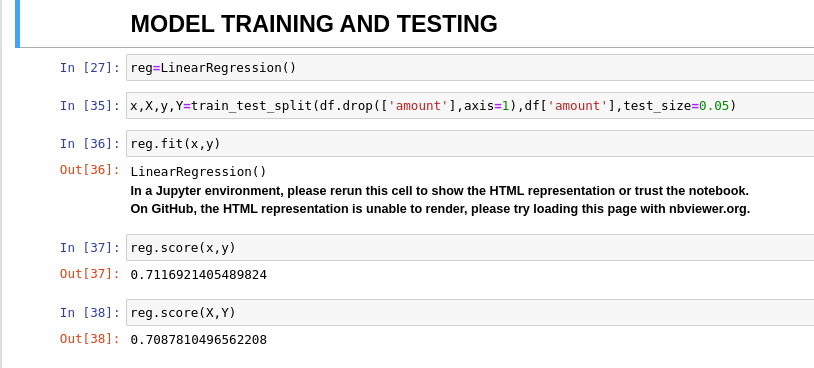


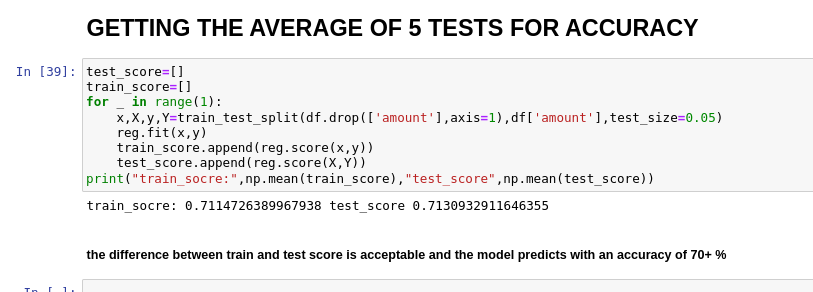
****

****

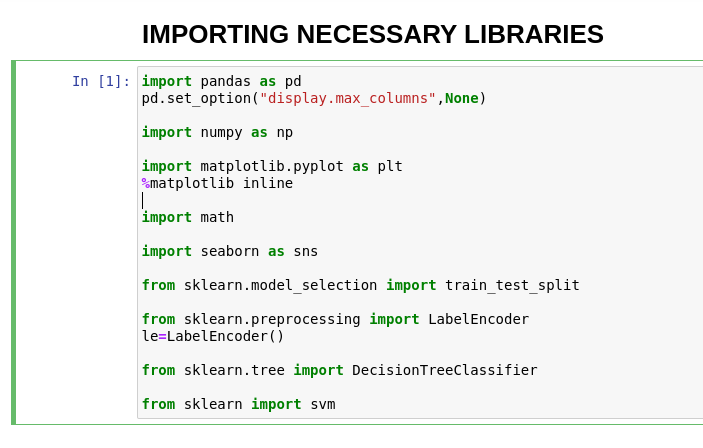
****

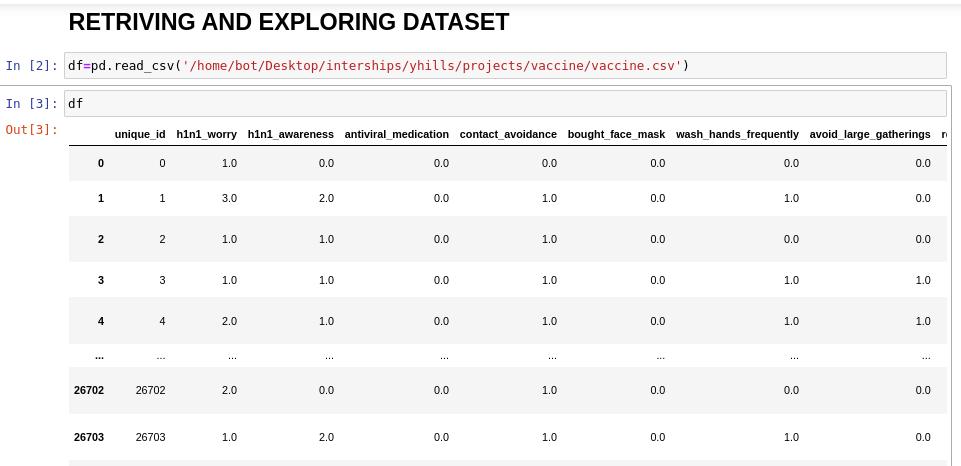
****

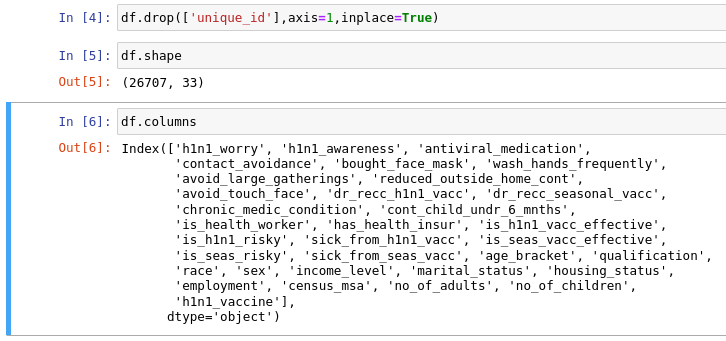
****

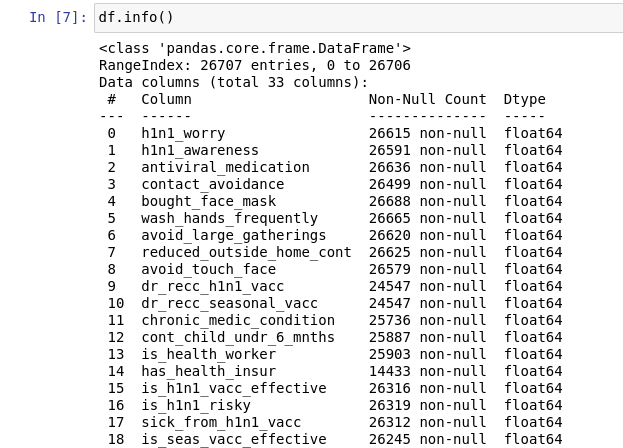
****

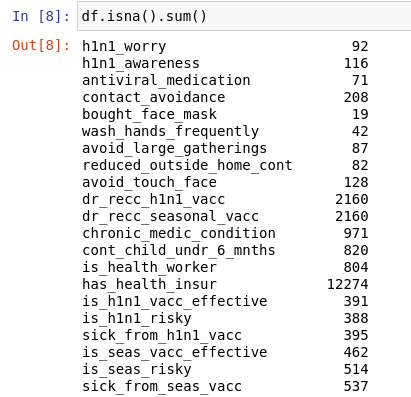
**TITLE: H1N1\_VACCINATION PREDICTION :**





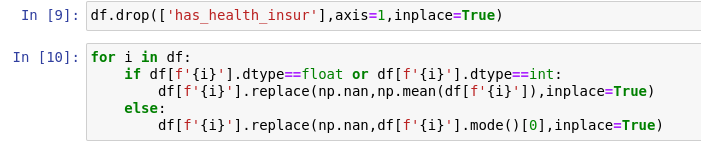






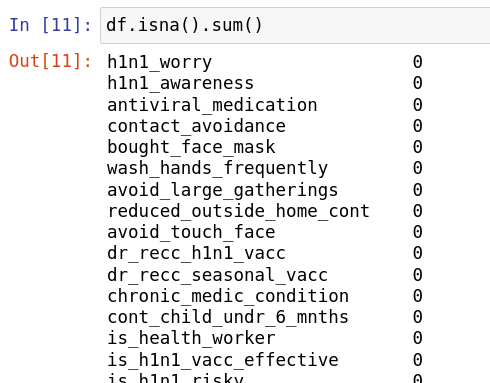
* dropping has\_health\_insur because it have a high amount of null

values.

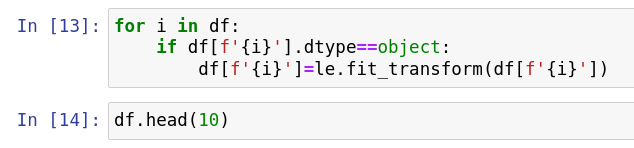
* By using the below method int and float variable having null values

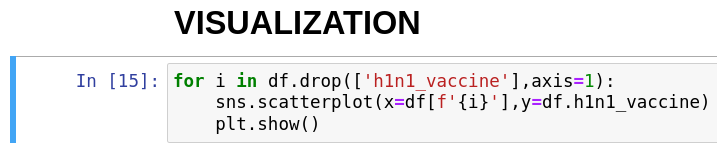
are replaced by median and object type are replaced by mode.

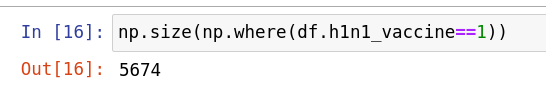
* Null values are removed.



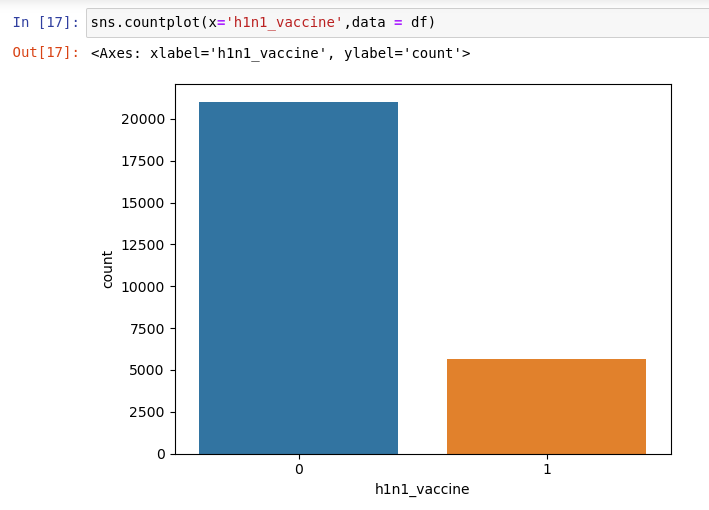
* Changing object type variable to integer using label encoder.

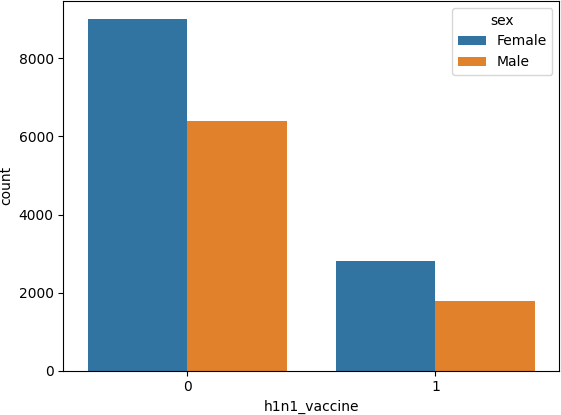


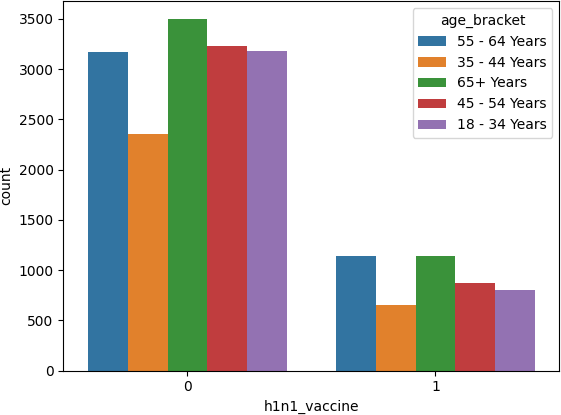


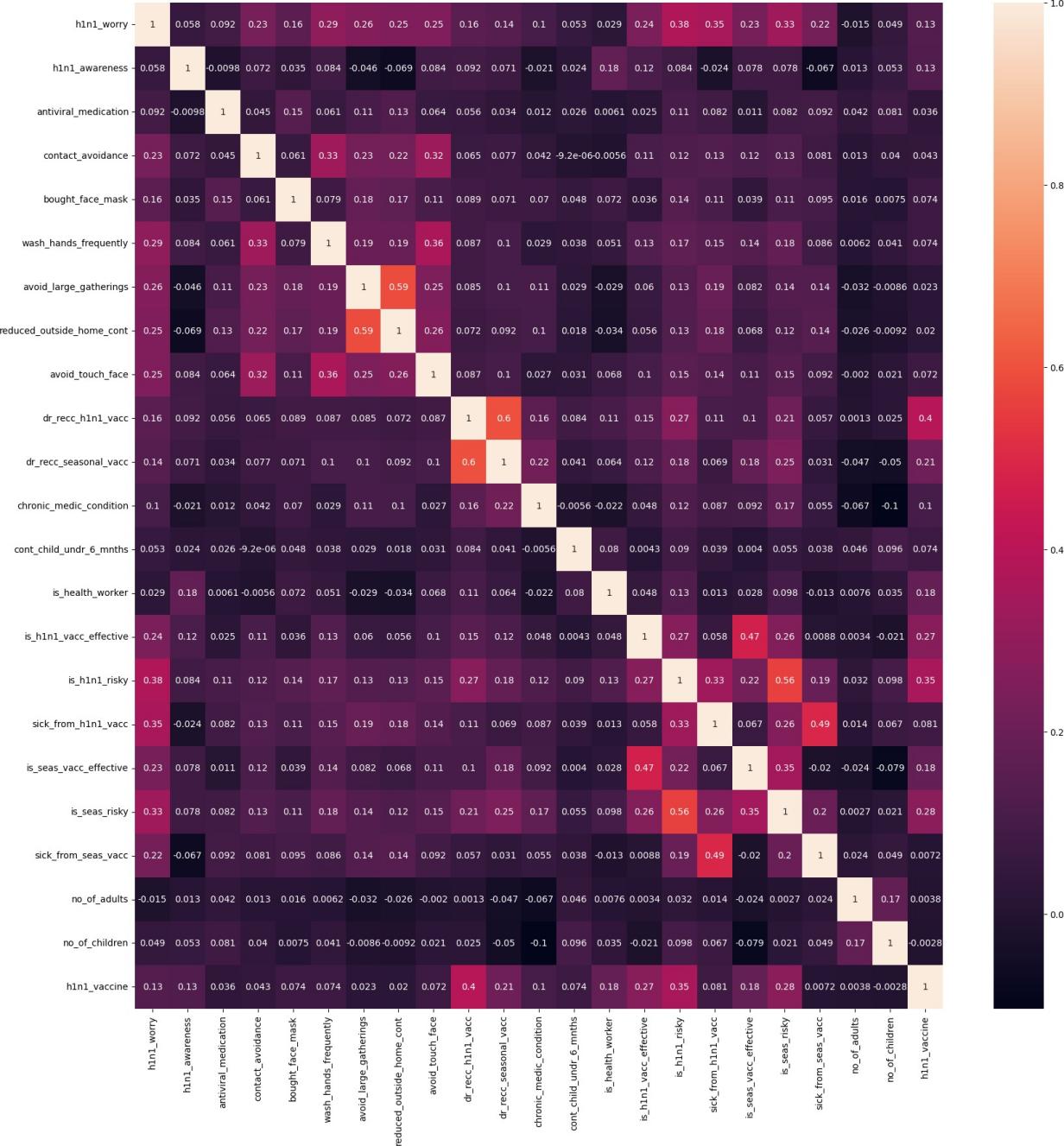


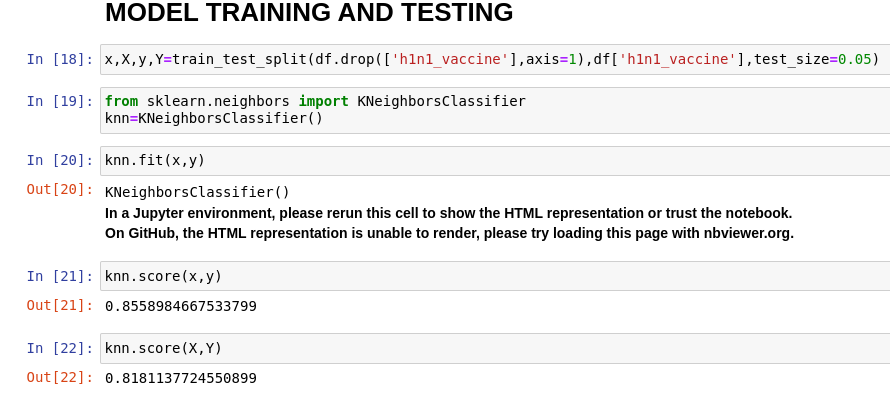
Countplot on number of people got vaccine.



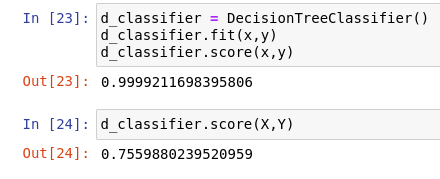




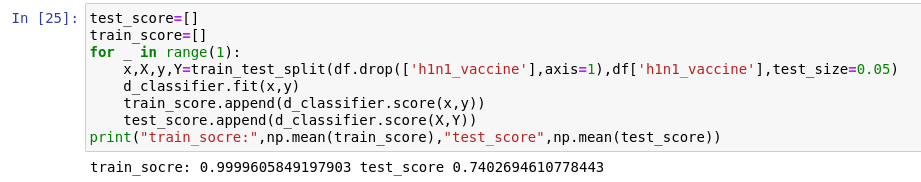


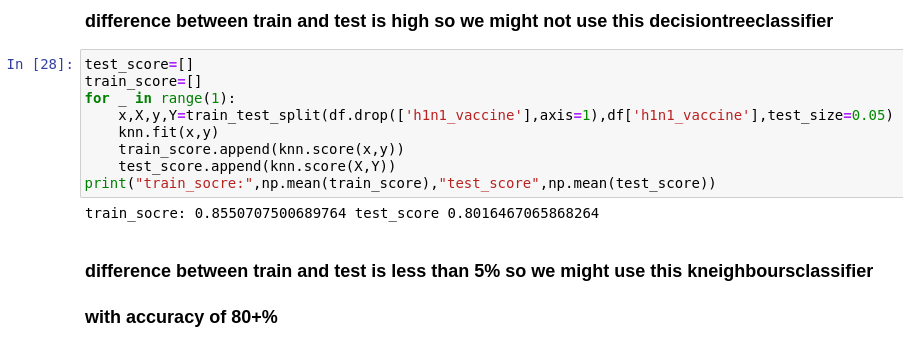


* Decision Tree Classifier



* scores:



* 

**CERTIFICATE:**

****

****