**HEPATITIS C DISEASE CLASSIFICATION USING MACHINE LEARNING ALGORITHMS**

**PROJECT REPORT**

***Submitted by***

**NEERAJ B – E0320032**

***In partial fulfilment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

**(Artificial Intelligence and Data Analytics)**

**Sri Ramachandra Faculty of Engineering and Technology**

**Sri Ramachandra Institute of Higher Education and Research, Porur, Chennai -600116**

**MARCH, 2022**

**HEPATITIS C DISEASE CLASSIFICATION USING MACHINE LEARNING ALGORITHMS**

**PROJECT REPORT**

***Submitted by***

**NEERAJ B – E0320032**

***In partial fulfilment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

**(Artificial Intelligence and Data Analytics)**

**Sri Ramachandra Faculty of Engineering and Technology**

**Sri Ramachandra Institute of Higher Education and Research, Porur, Chennai -600116**

**MARCH, 2022**

**BONAFIDE CERTIFICATE**

Certified that this project report **“HEPATITIS C DISEASE CLASSIFICATION USING MACHINE LEARNING ALGORITHMS”** is the bonafide record of work done by **“NEERAJ B – E0320032”** who carried out the internship work under my supervision.

**Signature of the Supervisor Signature of Vice-Principal**

|  |  |
| --- | --- |
| **Prof. Nirmala B**  **Assistant Professor,**  Department of Computer Science and Engineering  Sri Ramachandra Faculty of Engineering and Technology,  SRIHER, Porur, Chennai-600 116. | **Prof. M. Prema**  **Vice-Principal,**  Department of Computer Science and Engineering  Sri Ramachandra Faculty of Engineering and Technology,  SRIHER, Porur, Chennai-600 116. |

**Evaluation Date:**

**ACKNOWLEDGEMENT**

I express my sincere gratitude to our Chancellor, Vice-Chancellor and our sincere gratitude to our Provost **Dr. V.Raju** and our Vice-Principal **Prof. M.Prema** for their support and for providing the required facilities for carrying out this study.

I wish to thank my faculty supervisor, **Prof. Nirmala B**, Department of Computer Science and Engineering, Sri Ramachandra faculty of Engineering and Technology for extending help and encouragement throughout the project. Without her continuous guidance and persistent help, this project would not have been a success for me.

I am grateful to all the members of Sri Ramachandra Faculty of Engineering and Technology, my beloved parents and friends for extending the support, who helped us to overcome obstacles in the study.

|  |  |  |  |
| --- | --- | --- | --- |
| **TABLE OF CONTENTS**  **S. NO TITLE** | | **PAGES** | |
| 1. INTRODUCTION    1. Data Analytics    2. Machine Learning | | 8  8  8 | |
| 1. LITERATURE SURVEY | | 9 | |
| 1. PROPOSED METHODOLOGY   3.1 Introduction  3.2 Problem Statement  3.3 Objective  3.4 Methodology | | 10  10  10  11  11 | |
| 1. TOOLS AND TECHNOLOGIES   4.1 System Requirements  4.2 Technology Used  4.3 Software Requirements | | 13  13  13  14 | |
| 1. RESULTS AND DISCUSSION   5.1 Code  5.2 Sample Output | | 15  15  30 | |
| 1. CONCLUSION AND FUTURE WORK   6.1 Conclusion  6.2 Future Scopes  Reference  Worklog  **LIST OF FIGURES** | | 39  39  39  39  41 | |
| **FIGURE**  **NO** | **TITLE** | | **PAGE NO** | |
| 3.1 | Workflow | | 12 | |
| 5.1 | Analysing whether the gender population is affected more by the disease | | 30 | |
| 5.2 | Analysing the variation of each age group of the population affected maximum by the disease | | 30 | |
| 5.3 | Analysing whether the category belongs to most of the population affected by the disease. | | 31 | |
| 5.4 | Analysing whether the factor Protein is correlated with the factor ALB | | 31 | |
| 5.5 | Analysing number of people affected by each category | | 31 | |
| 5.6 | Analysing the factor Bilirubin correlated with the factor ALB | | 32 | |
| 5.7 | BIL, CHE, CHOL, GGT test levels diagnosing in each age group | | 32 | |
| 5.8 | ALB, ALP, ALT, AST test levels diagnosing in each age group | | 32 | |
| 5.9 | Analysing to find the average of patients through CREA test levels | | 33 | |
| 5.10 | Confusion matrix for Logistic Regression | | 33 | |
| 5.11 | Classification report for Logistic Regression | | 34 | |
| 5.12 | Confusion matrix for Random Forest | | 34 | |
| 5.13 | Classification report for Random Forest | | 35 | |
| 5.14 | Tree plot for Random Forest Classifier | | 35 | |
| 5.15 | Confusion matrix for Ada Boost | | 36 | |
| 5.16 | Classification report for Ada Boost | | 36 | |
| 5.17 | Confusion matrix for XG Boost | | 37 | |
| 5.18 | Classification report for XG Boost | | 37 | |
| 5.19 | Comparing the models | | 38 | |
| 5.20 | Visualization of models | | 38 | |

**ABSTRACT**

Hepatitis C is a short-term sickness for some people, but it becomes a long-term, chronic condition for more than half of those who contract the virus. Cirrhosis and liver cancer are two significant, potentially life-threatening side effects of chronic hepatitis C. Hepatitis C is a liver illness caused by the hepatitis C virus, according to the Centers for Disease Control and Prevention (CDC) (HCV). Hepatitis C is transmitted by blood contact with an infected individual.

The project was completed for International Centre for Emerging Technologies on Hepatitis C Disease Classification Using Machine Learning Algorithms from January to March in Data Science. The dataset was collected. After that, we cleansed the data, tested and trained the data, then developed a classification model. Later, using the accuracy of the interpreted classification of Logistics Regression, Random Forest, Ada Boost, and XG Boost were compared and visualized using visualization tools.

**CHAPTER 1**

**INTRODUCTION**

**1.1 Data Analytics:**

The word 'data' has been popular over the past decades. Be it data from a large industry about their sales or data from every individual about their opinions, and they must be analysed to benefit ourselves. The science of studying raw data to conclude it is known as data analytics.

Data analytics approaches and processes have been turned into mechanical methods and algorithms that operate on raw data for human consumption. Data analytics techniques and processes have been turned into mechanical methods and algorithms that operate on raw data for human consumption.

**1.2 Machine Learning:**

Machine learning is a form of artificial intelligence that lets software programmes to improve their prediction accuracy without being expressly designed to do so. In order to predict new output values, machine learning algorithms use past data as input. Machine learning is significant because it allows businesses to see trends in customer performance and company operating patterns while also assisting in the creation of new goods. Machine learning is at the heart of many of today's most successful businesses, like Twitter, Amazon, and Uber. For many firms, machine learning has become a key competitive distribution.

Nowadays, three common strategies are used to train machine learning algorithms. Machine learning may be divided into three categories: supervised learning, unsupervised learning, and reinforcement learning.

**CHAPTER 2**

**LITERATURE SURVEY**

* **Predicting the cross-immunoreactivity of hepatitis C virus hyper-variable region one peptides using polynomial neural networks (2015):**
* **Link:** https://sci-hub.hkvisa.net/10.1109/iccabs.2015.7344732
* Acute hepatitis infections are usually self-resolving and do not require treatment. However, in some cases, the condition becomes permanent (chronic). Chronic hepatitis C can cause liver damage (cirrhosis) or cancer in the long run.
* **Hepatitis C virus genetic association to rate of liver fibrosis progression (2013):**
* **Link:** https://sci-hub.hkvisa.net/10.1109/iccabs.2013.6629225
* This study was to design a model based on machine learning for diagnosing the Stage of liver fibrosis in the patients. The researchers examined 1240 patient records with chronic viral Hepatitis C. Machine learning models were developed and tested using data from 689 patients grouped by Stage of liver fibrosis. Essential predictors were chosen from nine usual prognostic factors. They achieved the highest accuracy of 80.56%.
* **Machine Learning Model for Diagnosing the Stage of Liver Fibrosis in Patients with Chronic Viral Hepatitis C (2021):**
* **Link:** <https://www.preprints.org/manuscript/202102.0488/v1>
* The goal of the study was to create a machine learning model for determining the stage of liver fibrosis in individuals with chronic viral hepatitis C using data from routine clinical exams. In compared to the "gold standard" of diagnosis (liver biopsy), the established approach for diagnosing the 3–4 stages of liver fibrosis in individuals with chronic viral hepatitis C was 80.56 percent accurate.

**CHAPTER 3**

**PROPOSED METHODOLOGY**

**3.1 Introduction:**

The hepatitis C virus (HCV) is transmitted by infected blood from one person to another. Hepatitis C treatment entails weekly injections and oral medications, which many HCV-infected people cannot endure due to other health conditions or unfavourable side effects. A virus-borne illness that targets the liver and causes inflammation. Contact with tainted blood spreads the virus through sharing needles or using unsterile tattoo equipment.

The majority of persons show no signs or symptoms. Those who experience symptoms may include weariness, nausea, lack of appetite, and skin and eye yellowing. Antiviral medicine is used to treat hepatitis C. Newer treatments can help some patients get rid of the infection.

**3.2 Problem Statement:**

* **PROBLEM:** To classify and analyse Hepatitis C Disease from the dataset based on factors like Albumin Blood Test, Alkaline phosphates, Gamma-Glutamyl Transferase, Proteins and etc.
* **SOLUTION:** To build a convolution ensemble machine learning model to classify the patient test levels into those that diagnosis disease – Hepatitis C or not

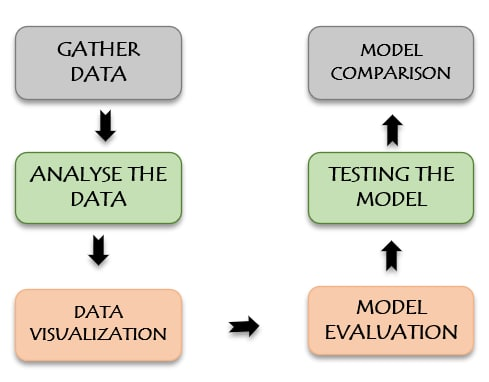
**3.3 Objective:**

The main goal of the project is to analyze & classify the hepatitis c disease for the given data and perform the following:

* To analyse which gender of population is affected more by the disease
* To find which age group of the population is affected at maximum by the disease
* To detect in which category most of the population affected by the disease belong too
* To discover the factor in which Protein is correlated with the factor ALB
* To know the number of people affected by each category
* To find the average of patients through Crea test levels.
* To detect the test levels in ALB, ALP, ALT, AST tests to diagnosis in each age group.
* To know the test levels in BIL, CHE, CHOL, GGT tests to diagnosis in each age group.
* To discover whether the factor Bilirubin is correlated with the factor ALB.

**3.4 Methodology**

* **DATA GATHERING:** The data related to the topic was gathered from Kaggle.
* **DATA ANALYSIS:** The data was examined to uncover hidden patterns and features that reflect pneumonia from the images in the supplied data set.
* **DATA VISUALIZATION:** The data was visualized in Tableauto make identifying patterns, trends, and outliers in massive data sets easier.
* **MODEL EVALUATION:** The model was evaluated to see if it was the best fit for the given problem and data.
* **TESTING THE MODEL:** The model was put to the test to see if it accurately classified or predicted the outcome.
* **MODEL COMPARISON:** The examined and predicted results were compared to see how good the model is at predicting the outcome.



**Fig 3.1 Workflow**

**CHAPTER 4**

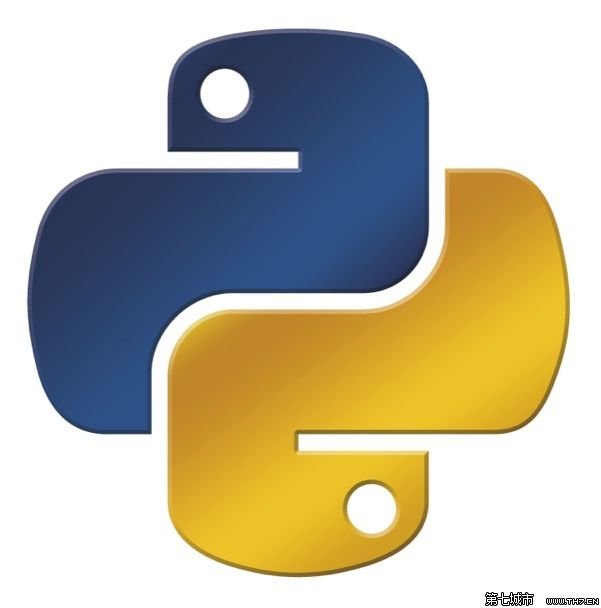
**TOOLS AND TECHNOLOGIES**

**4.1 System Requirements:**

* Operating System: Windows 10 or later (64 bit).
* Disk Space: 1.32 GB (does not include disk space for IDE or tools).
* Tools: Windows Power-Shell 5.0 or newer (it is pre-installed with windows 10)

**4.2 Technology Used:**

**Python:**

Python is a versatile programming language that might be used for various purposes. Python is used in web development, artificial intelligence, operating systems, mobile application development, and computer gaming. A straightforward and natural language just as strong as the significant rivals.

It's open-source so that everyone may contribute to its outcome. Code that is almost as comprehensible as plain English. Appropriate for doing errands frequently, given the short time to improve. Python is widely to use for web and software development, task automation, data analysis, and data visualisation. Python has been used by many non-programmers, such as accountants and scientists, for a range of common activities, such as arranging money, due to its relative ease of learning.

**4.3 Software Requirements:**

**Jupyter Notebook:**

Jupyter Notebook is free, open-source web software that lets you create and share documents with live code, equations, visualizations, and narrative text. It can be used for various activities, including data cleansing and transformation, numerical simulation, statistical modelling, data visualization, and machine learning.

**Tableau:**

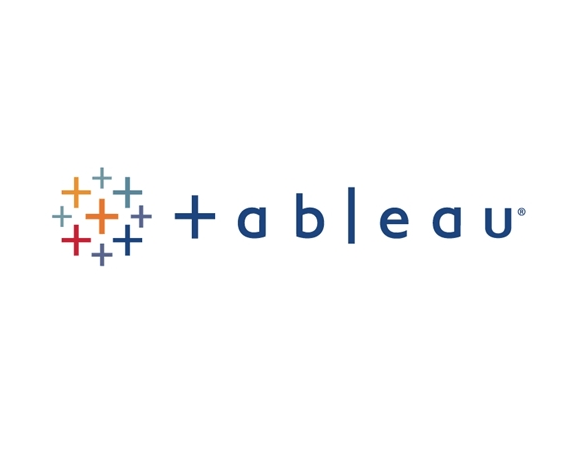
Tableau is a visual analytics technology that is reinventing the way we use teams to analyse issues by enabling individuals and companies to get the most out of their data.In the Business Intelligence Industry, Tableau is a strong and rapidly developing data visualisation application.

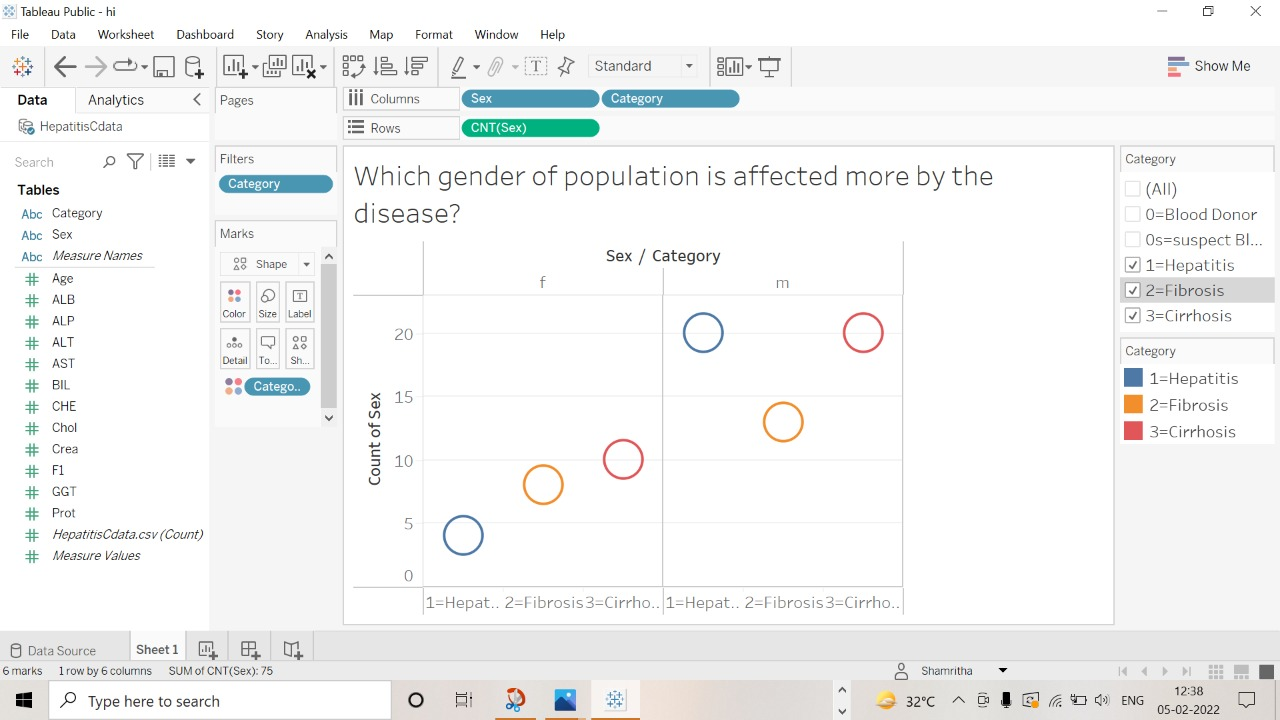
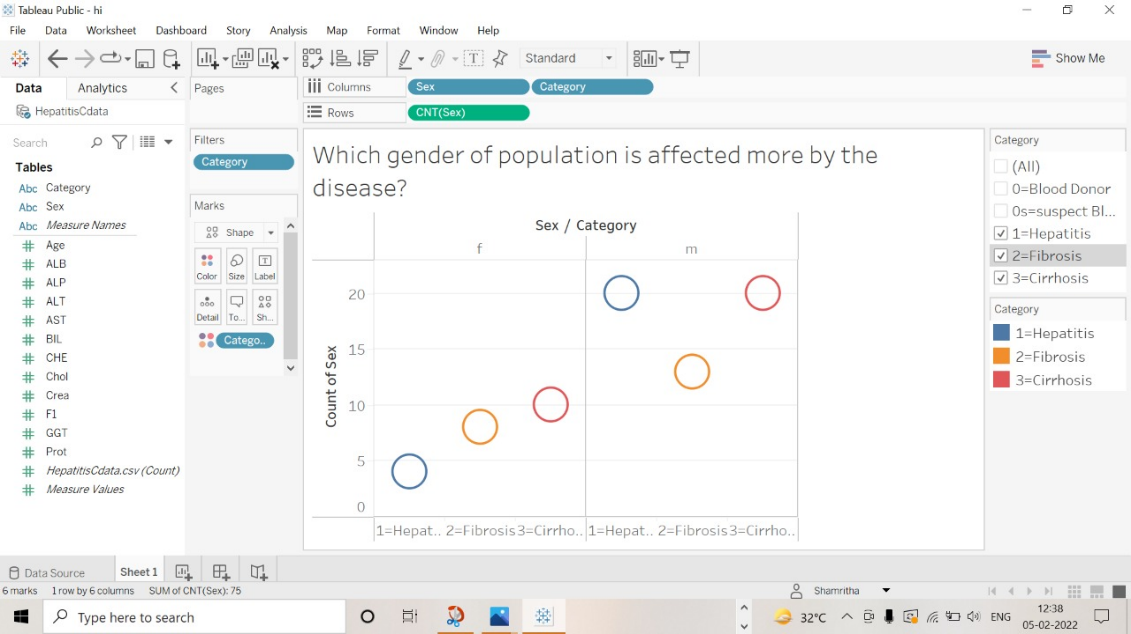
Tableau aids in the generation of data that specialists at all levels of an organisation can grasp. Non-technical people can easily develop customised dashboards.

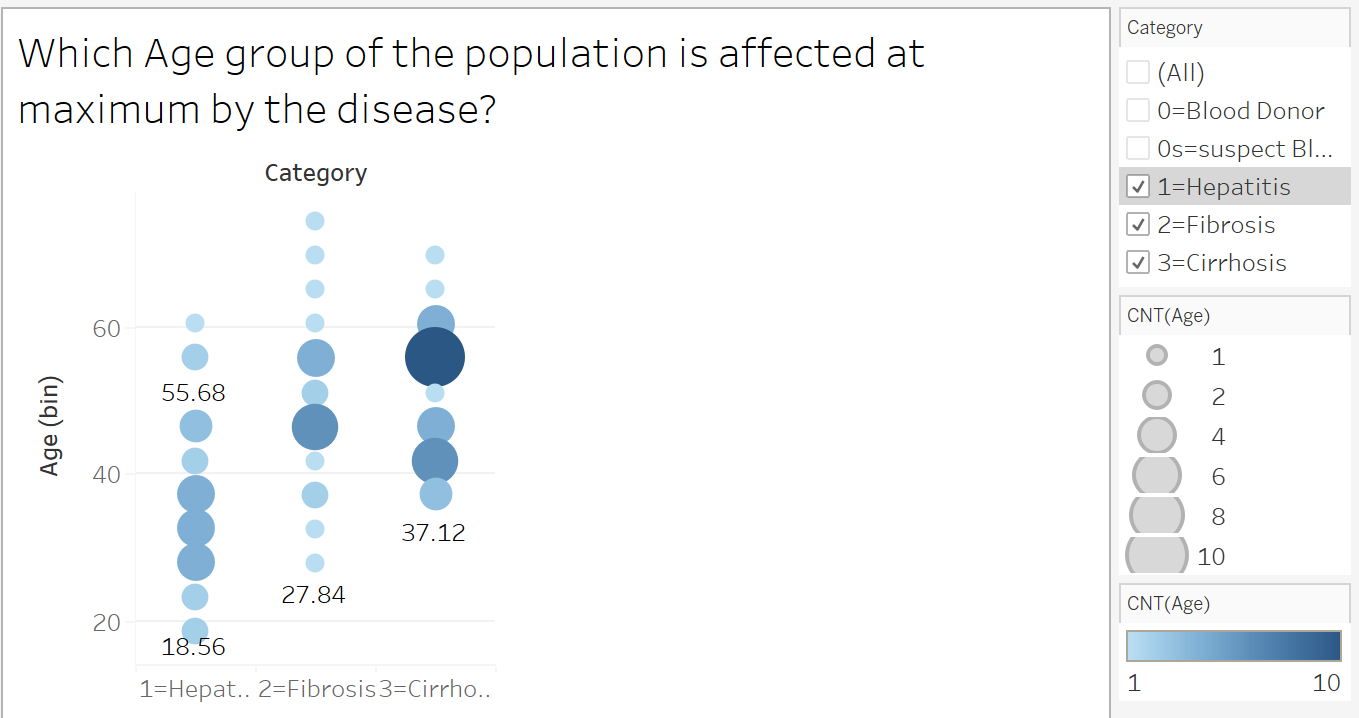
**CHAPTER 5**

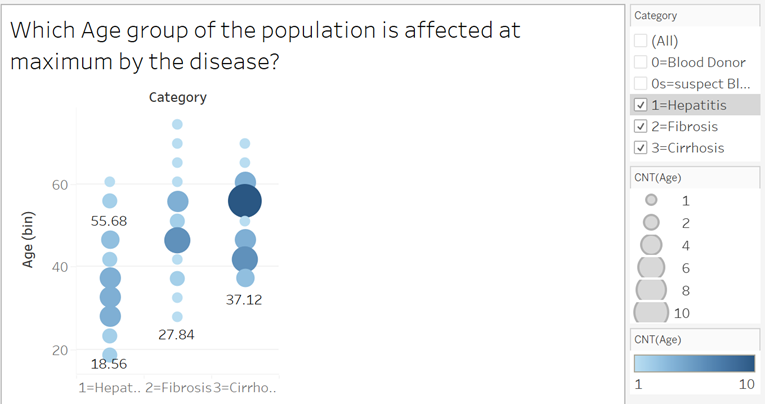
**RESULTS AND DISCUSSION**

**5.1 Code:**

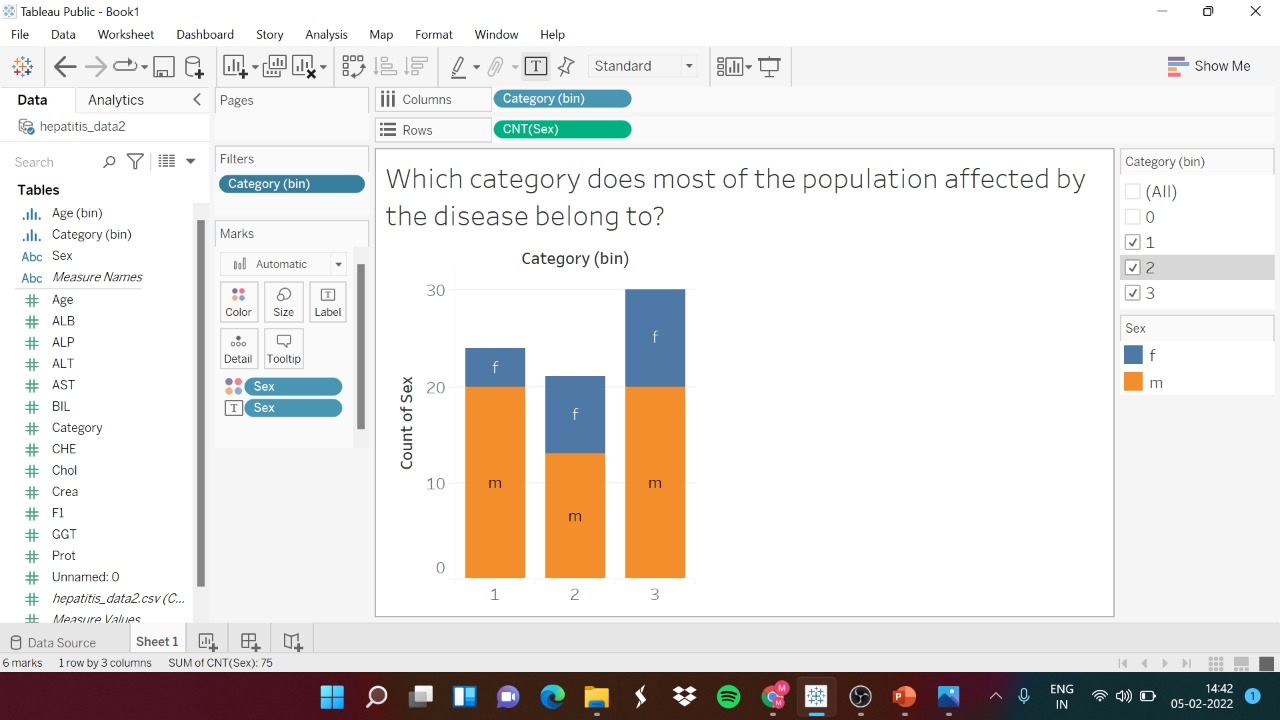
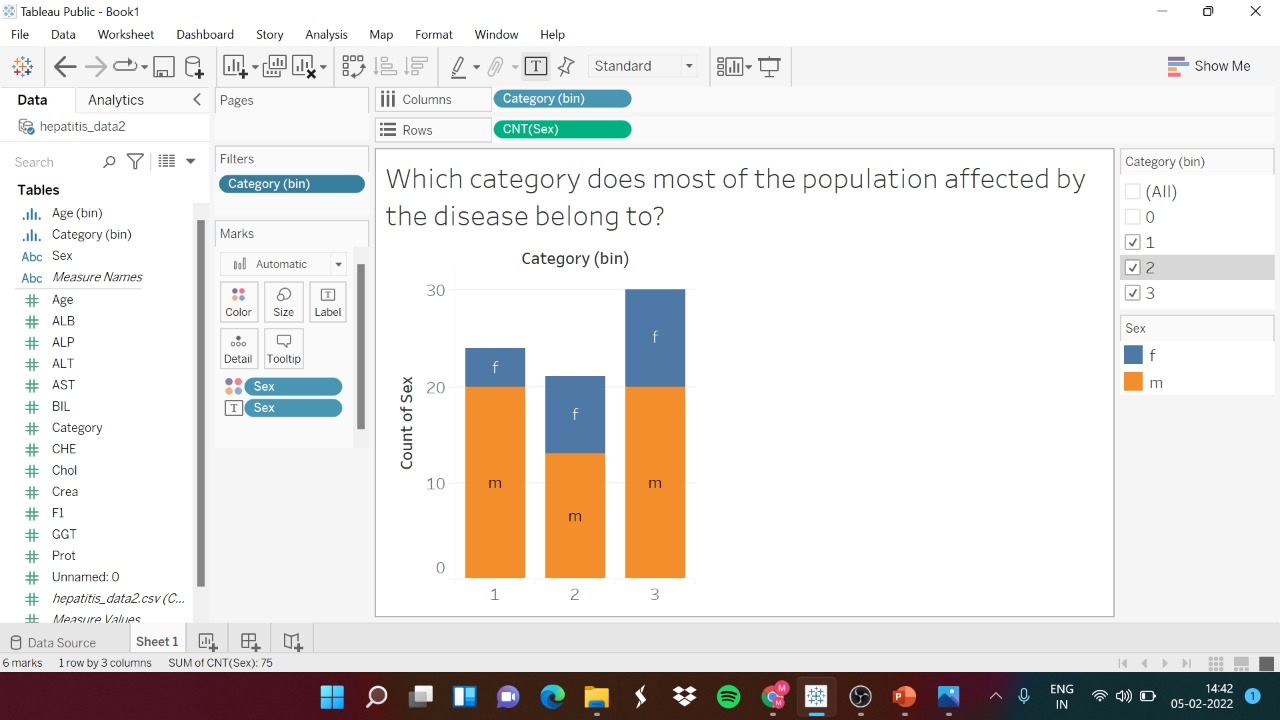
*#!/usr/bin/env python  
# coding: utf-8  
  
# # HEPATITIS DISEASE DIAGNOSIS  
  
# # TEAM DETAILS:  
  
# ### 1) Merin Rose Reji E0320044  
# ### 2) Shamritha.S E0320032  
  
# # PROBLEM STATEMENT:  
  
# To predict and analyze Hepatitis C Disease from the dataset based on factors like Albumin Blood Test,Alkaline phosphates,   
# Gamma-Glutamyl Transferase, Proteins and etc.  
  
# # OBJECTIVE:  
  
# Hepatitis is an inflammation of the liver, most commonly caused by a viral infection.The data set contains laboratory values of   
# blood donors and Hepatitis C patients and demographic values like age. The data was obtained from UCI Machine Learning  
# Repository.  
#   
# - To analyze which gender of population is affected more by the disease  
#   
# - To find which age group of the population is affected at maximum by the disease  
#   
# - To detect in which category most of the population affected by the disease belong to  
#   
# - To discover the factor in which Protein is correlated with the factor ALB  
#   
# - To know the number of people affected by each category  
#   
# - To find the average of patients through Crea test levels.  
#   
# - To detect the test levels in ALB,ALP,ALT,AST tests to diagnosis in each age group .  
#   
# - To know the test levels in BIL,CHE,CHOL,GGT tests to diagnosis in each age group.  
#   
# - To discover whether the factor Bilirubin is correlated with the factor ALB.  
  
# ### Columns Abbreviation  
# ALB-Albumin Blood Test  
#   
# ALP-Alkaline phosphatase  
#   
# ALT-Alanine Transaminase  
#   
# AST-Aspartate Transaminase  
#   
# BIL-Bilirubin  
#   
# CHE-Acetylcholinesterase  
#   
# CHOL-Cholesterol  
#   
# CREA-Creatinine  
#   
# GGT-Gamma-Glutamyl Transferase  
#   
# PROT-Proteins  
  
# # Importing the library  
  
# In[1]:***import** pandas **as** pd  
**import** numpy **as** np  
**import** seaborn **as** sns  
**from** matplotlib **import** pyplot **as** plt  
**import** matplotlib.pyplot **as** plt  
**import** matplotlib **as** mpl  
**import** warnings  
  
  
*# # Data Collection  
  
# In[2]:*data = pd.read\_csv(**"HepatitisCdata.csv"**)  
data  
  
  
*# In[3]:  
  
  
#Renaming "Unnamed: 0" column to "Id"*data.rename(  
 columns=({**'Unnamed: 0'**: **'Patient\_ID'**}),  
 inplace=**True**,  
)  
  
  
*# In[4]:*data.head()  
  
  
*# In[5]:*data.tail()  
  
  
*# In[6]:*data.Category.unique()  
  
  
*# # Data Cleaning  
  
# In[7]:  
  
  
# Finding the shape of the dataset*data.shape  
  
  
*# In[8]:  
  
  
# checking whether NA value are present or not for each column.*isna\_value = data.isna()  
isna\_value  
  
  
*# In[9]:  
  
  
# checking whether NA value are present or not for the whole dataset.*print(data.isnull().values.any())  
  
  
*# In[10]:  
  
  
# Count NA values present in each columns.*print(data.isnull().sum())  
  
  
*# In[11]:  
  
  
# Total count of NA values present in the entire dataset*print(data.isnull().sum().sum())  
  
  
*# In[12]:  
  
  
# Replacing the NA values in mean of 'ALB' column*mean\_ALB = data[**'ALB'**].mean()  
data[**'ALB'**].fillna(mean\_ALB,inplace=**True**)  
  
  
*# In[13]:  
  
  
# Replacing the NA values in mean of 'ALP' column*mean\_ALP = data[**'ALP'**].mean()  
data[**'ALP'**].fillna(mean\_ALP,inplace=**True**)  
  
  
*# In[14]:  
  
  
# Replacing the NA values in mean of 'ALT' column*mean\_ALT = data[**'ALT'**].mean()  
data[**'ALT'**].fillna(mean\_ALT,inplace=**True**)  
  
  
*# In[15]:  
  
  
# Replacing the NA values in mean of 'CHOL' column*mean\_CHOL = data[**'CHOL'**].mean()  
data[**'CHOL'**].fillna(mean\_CHOL,inplace=**True**)  
  
  
*# In[16]:  
  
  
# Replacing the NA values in mean of 'PROT' column*mean\_PROT = data[**'PROT'**].mean()  
data[**'PROT'**].fillna(mean\_PROT,inplace=**True**)  
  
  
*# In[17]:  
  
  
# Count NA values present in each columns from the updated dataset.*print(data.isnull().sum())  
  
  
*# In[18]:  
  
  
# Count NA values present in 'ALP' column from the updated dataset.*data[**"ALP"**].isnull()  
  
  
*# # Data Understanding  
  
# In[19]:  
  
  
# Detailed info about the dataset.*data.info()  
  
  
*# In[20]:  
  
  
# Finding the unique Values in the Category column.*data[**'Category'**].unique()  
  
  
*# In[21]:  
  
  
# Finding the unique Values in the Sex column.*data[**'Sex'**].unique()  
  
  
*# In[22]:  
  
  
# Finding the unique Values in the Age column.*data[**'Age'**].unique()  
  
  
*# In[23]:  
  
  
# Satistical Describtion of the Hepatitis-C data.*data.describe()  
  
  
*# In[24]:  
  
  
# Satistical Describtion for non-numerical data.*data.describe(exclude=**'number'**)  
  
  
*# In[25]:  
  
  
# Datatypes of each column.*data.dtypes  
  
  
*# In[26]:  
  
  
# Total Length of the rows present in the dataset.*len(data)  
  
  
*# In[27]:  
  
  
# Name of the columns present in the dataset.*data.columns  
  
  
*# In[28]:  
  
  
# Total Length of the columns present in the dataset.*len(data.columns)  
  
  
*# In[29]:  
  
  
# Maximum value of each column*data[[**'Age'**,**'ALB'**,**'ALP'**,**'ALT'**,**'AST'**,**'BIL'**,**'CHE'**,**'CHOL'**,**'CREA'**,**'GGT'**,**'PROT'**]].max()  
  
  
*# In[30]:  
  
  
# Minimum value of each column*data[[**'Age'**,**'ALB'**,**'ALP'**,**'ALT'**,**'AST'**,**'BIL'**,**'CHE'**,**'CHOL'**,**'CREA'**,**'GGT'**,**'PROT'**]].min()  
  
  
*# In[31]:  
  
  
# Average value of each column*data[[**'Age'**,**'ALB'**,**'ALP'**,**'ALT'**,**'AST'**,**'BIL'**,**'CHE'**,**'CHOL'**,**'CREA'**,**'GGT'**,**'PROT'**]].mean()  
  
  
*# ## Feature Extraction  
  
# In[32]:  
  
  
# Dropping the 'Patient\_ID' column from the dataset*New\_data = data.drop(**'Patient\_ID'**, axis = 1)  
  
  
*# In[33]:  
  
  
# Replacing the 'Category' values with 0 or 1*New\_data[**'Category'**].loc[New\_data[**'Category'**].isin([**'0=Blood Donor'**, **'0s=suspect Blood Donor'**])] = 0  
New\_data[**'Category'**].loc[New\_data[**'Category'**].isin([**'1=Hepatitis'**, **'2=Fibrosis'**, **'3=Cirrhosis'**])] = 1  
  
*# Replacing 'Sex' column with male as 1 and Female as 2*New\_data[**'Sex'**] = New\_data[**'Sex'**].map({**'m'**: 1, **'f'**: 2})  
  
  
*# In[34]:*New\_data.head()  
  
  
*# In[35]:  
  
  
# Converting the 'category' column as Integer*New\_data[**'Category'**] = New\_data[**'Category'**].astype(int)  
  
  
*# In[36]:*New\_data.head(2)  
  
  
*# ## Extracting Independent and dependent Variable  
  
# In[37]:*X = New\_data.drop([**'Category'**], axis = 1)  
X.head(3)  
  
  
*# In[38]:*y = New\_data[**'Category'**]  
y.head(3)  
  
  
*# ## Splitting The Data into Training and Testing datasets  
  
# - Here, we are trying to predict whether the patient has Hepatitis C or not using the given data.   
# - Hence, the Category will be the y label and rest of the data will be the X or the input data.  
  
# In[39]:***from** sklearn.model\_selection **import** train\_test\_split  
  
  
*# In[40]:  
  
  
# Splitting the Data into 70%*X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, train\_size=0.7, random\_state=42)  
X\_train.shape, X\_test.shape  
  
  
*# In[41]:*len(X\_train), len(X\_test)  
  
  
*# ## Feature Scaling  
  
# In[42]:***from** sklearn.preprocessing **import** StandardScaler  
scaler = StandardScaler()  
X\_train = scaler.fit\_transform(X\_train)  
X\_test = scaler.transform(X\_test)  
  
  
*# # Logistic Regression  
  
# In[43]:  
  
  
#Fitting Logistic Regression to the Training set***from** sklearn.linear\_model **import** LogisticRegression  
lr = LogisticRegression()  
lr.fit(X\_train, y\_train)  
  
  
*# In[44]:*LogisticRegressionScore1 = lr.score(X\_test,y\_test)  
print(**"Accuracy obtained by Logistic Regression model for testing data :"**, LogisticRegressionScore1\*100)  
  
  
*# ## Predicting the Test Result  
  
# In[45]:*y\_pred\_lr = lr.predict(X\_test)   
  
  
*# In[46]:  
  
  
#confusion matrix***import** matplotlib.pyplot **as** plt  
**from** sklearn.datasets **import** make\_classification  
**from** sklearn.metrics **import** plot\_confusion\_matrix  
**from** sklearn.model\_selection **import** train\_test\_split  
**from** sklearn.svm **import** SVC  
  
plot\_confusion\_matrix(lr, X\_test, y\_test,cmap=**"Blues\_r"**)   
plt.show()  
  
  
*# ## Comparing Y-test and Predicted Value  
  
# In[47]:*pd.DataFrame({**'y\_test'**:y\_test,**'y\_pred '**:y\_pred\_lr })  
  
  
*# ## Classification Report For Logistic Regression  
  
# In[48]:  
  
  
# Having a look at the classification report of Logistic Regression***from** sklearn **import** metrics  
print(metrics.classification\_report(y\_test, y\_pred\_lr))  
  
  
*# In[96]:***from** sklearn.metrics **import** precision\_score,recall\_score,f1\_score  
logi\_precision = precision\_score(y\_test,y\_pred\_lr)  
logi\_recall = recall\_score(y\_test,y\_pred\_lr)  
logi\_f1 = f1\_score(y\_test,y\_pred\_lr)  
print(**'Presision Score for Logistic Regression'**,logi\_precision)  
print(**'Recall Score for Logistic Regression'**,logi\_recall)  
print(**'F1 Score for Logistic Regression'**,logi\_f1)  
  
  
*# # Random Forest Model   
  
# - Fitting Random Forest Classifier to the training set   
  
# In[56]:***from** sklearn.ensemble **import** RandomForestClassifier  
  
  
*# In[57]:*classifier\_rf = RandomForestClassifier(random\_state=42, n\_jobs=-1, max\_depth=7,  
 n\_estimators=100, oob\_score=**True**, criterion=**'entropy'**)  
  
  
*# In[58]:*get\_ipython().run\_cell\_magic(**'time'**, **''**, **'classifier\_rf.fit(X\_train, y\_train)'**)  
  
  
*# In[59]:*RandomForestClassifierScore1 = classifier\_rf.score(X\_test,y\_test)  
print(**"Accuracy obtained by Random Forest Classifier for testing data :"**, RandomForestClassifierScore1\*100)  
  
  
*# ## Predicting the Test Set result  
  
# In[60]:*y\_pred\_rfc = classifier\_rf.predict(X\_test)  
  
  
*# ## Ploting The Confusion Matrix of Random Forest Classifier  
  
# In[61]:***import** matplotlib.pyplot **as** plt  
**from** sklearn.datasets **import** make\_classification  
**from** sklearn.metrics **import** confusion\_matrix   
**from** sklearn.metrics **import** plot\_confusion\_matrix  
**from** sklearn.model\_selection **import** train\_test\_split  
**from** sklearn.svm **import** SVC  
  
plot\_confusion\_matrix(classifier\_rf, X\_test, y\_test,cmap=**"BuPu\_r"**)   
plt.show()  
  
  
*# ## Comparing Y-test and Predicted Value  
  
# In[62]:*pd.DataFrame({**'Actual Value'**:y\_test,**'Predicted Value'**:y\_pred\_rfc})  
  
  
*# In[63]:*np.array(y\_test)  
  
  
*# In[64]:*y\_pred\_rfc  
  
  
*# ## Classification Report for Random Forest Classifier   
  
# - Having a look at the classification report of Random Forest Classifier  
  
# In[65]:***from** sklearn **import** metrics  
  
  
*# In[66]:*print(metrics.classification\_report(y\_test, y\_pred\_rfc))  
  
  
*# In[102]:***from** sklearn.metrics **import** precision\_score,recall\_score,f1\_score  
rand\_precision = precision\_score(y\_test,y\_pred\_rfc)  
rand\_recall = recall\_score(y\_test,y\_pred\_rfc)  
rand\_f1 = f1\_score(y\_test,y\_pred\_rfc)  
print(**'Presision Score for Random Forest Classifier'**,rand\_precision)  
print(**'Recall Score for Random Forest Classifier'**,rand\_recall)  
print(**'F1 Score for Random Forest Classifier'**,rand\_f1)  
  
  
*# ## Tree Plot for Random Forest Classifier  
  
# - Visualizing the single Decision Tree from the Random Forest  
  
# In[60]:***from** sklearn.tree **import** plot\_tree  
plt.figure(figsize=(80,40))  
plot\_tree(classifier\_rf.estimators\_[0], feature\_names = X.columns,class\_names=[**'No Hepatitis'**, **"Hepatitis"**],filled=**True**);  
  
  
*# # Ada Boost  
  
# In[69]:  
  
  
# Importing neccesary packages  
# Load libraries***from** sklearn.ensemble **import** AdaBoostClassifier  
**from** sklearn **import** datasets  
**from** sklearn.model\_selection **import** train\_test\_split  
**from** sklearn **import** metrics  
  
  
*# In[70]:  
  
  
# Create adaboost classifer object*AdaModel = AdaBoostClassifier(n\_estimators=100, learning\_rate=1)  
  
  
*# In[71]:  
  
  
# Train Adaboost Classifer*AdaModel.fit(X\_train, y\_train)  
  
  
*# ## Predicting the Test Result  
  
# In[72]:  
  
  
#Predict the response for test dataset*y\_pred\_adBt = AdaModel.predict(X\_test)  
  
  
*# In[73]:  
  
  
# Model Accuracy, how often is the classifier correct?*adaboostscore1 = metrics.accuracy\_score(y\_test, y\_pred\_adBt)  
print(**"Accuracy obtained by Ada Boost model:"**,adaboostscore1\*100.0)  
  
  
*# In[74]:  
  
  
# Import Support Vector Classifier***from** sklearn.svm **import** SVC  
*#Import scikit-learn metrics module for accuracy calculation***from** sklearn **import** metrics  
svc=SVC(probability=**True**, kernel=**'linear'**)  
  
*# Create adaboost classifer object*abc =AdaBoostClassifier(n\_estimators=50, base\_estimator=svc,learning\_rate=1)  
  
  
*# In[75]:  
  
  
# Train Adaboost Classifer*abc.fit(X\_train, y\_train)  
  
*#Predict the response for test dataset*y\_pred\_adBt1 = abc.predict(X\_test)  
  
  
*# In[76]:  
  
  
# Model Accuracy, how often is the classifier correct?*adaboostscore2 = metrics.accuracy\_score(y\_test, y\_pred\_adBt1)  
print(**"Accuracy obtained by Ada Boost model: "**, adaboostscore2\*100.0)  
  
  
*# ## Ploting The Confusion Matrix of Ada Boost  
  
# In[77]:***import** matplotlib.pyplot **as** plt  
**from** sklearn.datasets **import** make\_classification  
**from** sklearn.metrics **import** confusion\_matrix   
**from** sklearn.metrics **import** plot\_confusion\_matrix  
**from** sklearn.model\_selection **import** train\_test\_split  
**from** sklearn.svm **import** SVC  
  
plot\_confusion\_matrix(AdaModel, X\_test, y\_test,cmap=**"pink"**)   
plt.show()  
  
  
*# ## Comparing Y-test and Predicted Value  
  
# In[78]:*pd.DataFrame({**'Actual Value'**:y\_test,**'Predicted Value'**:y\_pred\_adBt})  
  
  
*# In[79]:*np.array(y\_test)  
  
  
*# In[80]:*y\_pred\_adBt  
  
  
*# ## Classification Report for Ada Boost  
# - Having a look at the classification report of Ada Boost  
  
# In[81]:***from** sklearn **import** metrics  
  
  
*# In[82]:*print(metrics.classification\_report(y\_test, y\_pred\_adBt))  
  
  
*# In[103]:***from** sklearn.metrics **import** precision\_score,recall\_score,f1\_score  
  
ada\_precision = precision\_score(y\_test,y\_pred\_adBt)  
ada\_recall = recall\_score(y\_test,y\_pred\_adBt)  
ada\_f1 = f1\_score(y\_test,y\_pred\_adBt)  
print(**'Presision Score for Ada Boost'**,ada\_precision)  
print(**'Recall Score for Ada Boost'**,ada\_recall)  
print(**'F1 Score for Ada Boost'**,ada\_f1)  
  
  
*# # XG Boost  
  
# In[75]:*get\_ipython().system(**' pip install xgboost'**)  
  
  
*# In[84]:***from** numpy **import** loadtxt  
**from** xgboost **import** XGBClassifier  
**from** sklearn.model\_selection **import** train\_test\_split  
**from** sklearn.metrics **import** accuracy\_score  
**import** xgboost **as** xgb  
**from** sklearn **import** preprocessing  
**import** time  
**import** warnings  
warnings.filterwarnings(**"ignore"**)  
  
  
*# In[85]:*start = time.time()  
xgbr = xgb.XGBClassifier(max\_depth = 5,n\_estimators = 100, random\_state=42)  
  
  
*# In[86]:*xgbr.fit(X\_train, y\_train)  
y\_pred\_xgbr = xgbr.predict(X\_test)  
  
  
*# ## Finding the accurancy of the model  
  
# In[87]:*xgboostscore1 = accuracy\_score(y\_test, y\_pred\_xgbr)  
  
  
*# In[88]:*start\_time = time.time()  
print(**"Accuracy obtained by XG Boost model: %.2f%%"** % (xgboostscore1 \* 100.0))  
print(**"%s seconds "** % (time.time() - start\_time))  
end = time.time()  
diff = end - start  
print(**'Execution time:'**, diff)  
  
  
*# ## Ploting The Confusion Matrix of XG Boost  
  
# In[89]:***import** matplotlib.pyplot **as** plt  
**from** sklearn.datasets **import** make\_classification  
**from** sklearn.metrics **import** confusion\_matrix   
**from** sklearn.metrics **import** plot\_confusion\_matrix  
**from** sklearn.model\_selection **import** train\_test\_split  
**from** sklearn.svm **import** SVC  
  
plot\_confusion\_matrix(xgbr, X\_test, y\_test,cmap=**"summer"**)   
plt.show()  
  
  
*# ## Comparing Y-test and Predicted Value  
  
# In[90]:*pd.DataFrame({**'Actual Value'**: y\_test,**'Predicted Value'**: y\_pred\_xgbr})  
  
  
*# In[91]:*np.array(y\_test)  
  
  
*# In[92]:*y\_pred\_xgbr  
  
  
*# ## Classification Report for XG Boost  
# - Having a look at the classification report of XG Boost  
  
# In[93]:***from** sklearn **import** metrics  
  
  
*# In[94]:*print(metrics.classification\_report(y\_test, y\_pred\_xgbr))  
  
  
*# In[106]:***from** sklearn.metrics **import** precision\_score,recall\_score,f1\_score  
  
xg\_precision = precision\_score(y\_test,y\_pred\_xgbr)  
xg\_recall = recall\_score(y\_test,y\_pred\_xgbr)  
xg\_f1 = f1\_score(y\_test,y\_pred\_xgbr)  
print(**'Presision Score for XG Boost'**,xg\_precision)  
print(**'Recall Score for XG Boost'**,xg\_recall)  
print(**'F1 Score for XG Boost'**,xg\_f1)  
  
  
*# ## Comparing the models  
  
# In[117]:*com = []  
com.append((**'Logistic Regression'**,LogisticRegressionScore1\*100,logi\_precision\*100,logi\_recall\*100,logi\_f1\*100))  
com.append((**'Random Forest'**,RandomForestClassifierScore1\*100,rand\_precision\*100,rand\_recall\*100,rand\_f1\*100))  
com.append((**'Ada Boost'**,adaboostscore1\*100,ada\_precision\*100,ada\_recall\*100,ada\_f1\*100))  
com.append((**'XG Boost'**,xgboostscore1\*100,xg\_precision\*100,xg\_recall\*100,xg\_f1\*100))  
  
  
*# In[118]:*df = pd.DataFrame(com,columns=[**'Machine Learning Model'**,**'Score'**,**'Precision'**,**'Recall'**,**'F1 Score'**])  
df  
  
  
*# In[112]:***import** plotly.express **as** px  
  
fig = px.bar(df, x=**'Machine Learning Model'**, y=**'Score'**,  
 color=**'Machine Learning Model'**,  
 color\_discrete\_sequence=px.colors.qualitative.Plotly,  
 height=400, width=800,)  
  
fig.show()  
  
  
*# In[170]:  
  
  
# define figure*fig, ax = plt.subplots(1, figsize=(16, 8))  
  
x = np.arange(0, len(df[**'Machine Learning Model'**].index))  
  
plt.bar(x-0.3, df[**'Score'**], width = 0.2,color = **'#1D2F6F'**)  
plt.bar(x-0.1, df[**'Precision'**], width = 0.2, color = **'#8390FA'**)  
plt.bar(x+0.1, df[**'Recall'**], width = 0.2, color = **'#6EAF46'**)  
plt.bar(x+0.3, df[**'F1 Score'**], width = 0.2, color = **'#FAC748'**)  
  
plt.xticks(x,[**'Logistic Regression'**,**'Random Forest'**,**'Ada Boost'**,**'XG Boost'**])  
  
plt.title(**'Comparison of the models'**, loc =**'center'**)  
plt.legend([**'Score'**,**'Precision'**,**'Recall'**,**'F1 score'**], loc=**'upper right'**, ncol = 4)  
plt.show()

**5.2 Sample Output:**

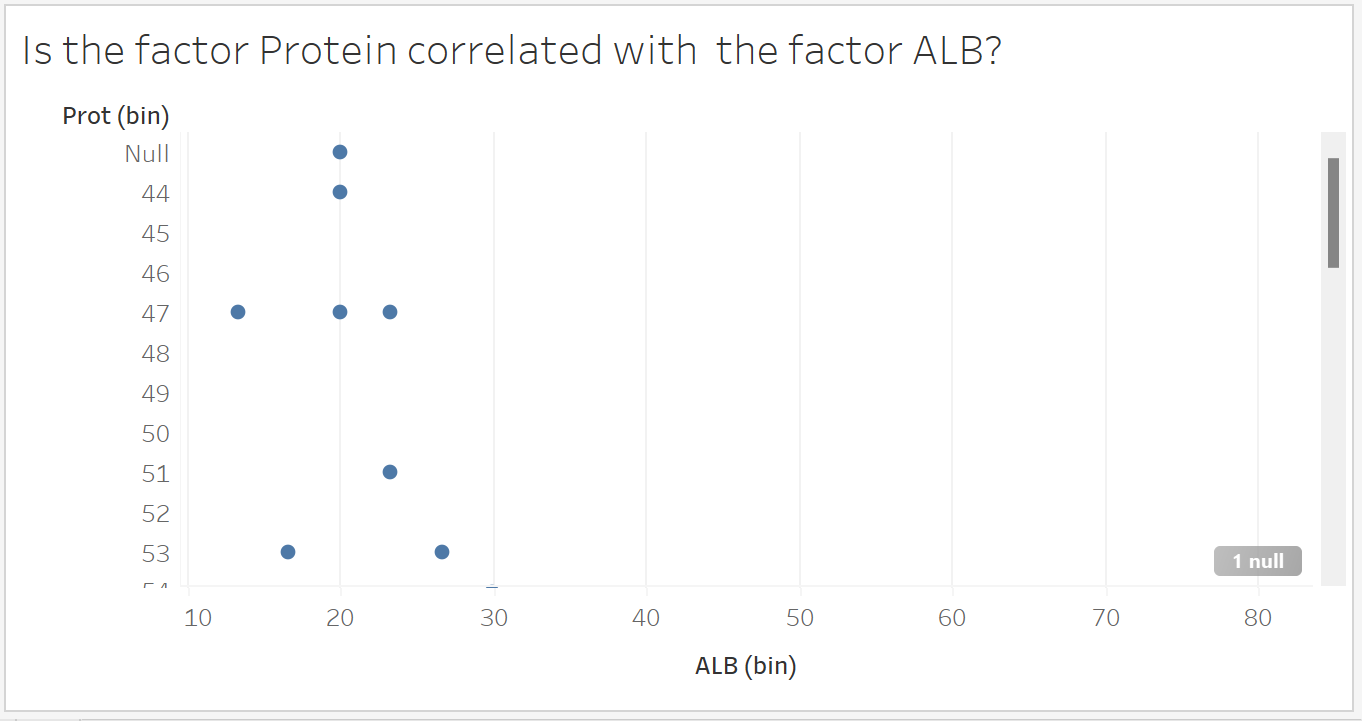
**Fig 5.1 Analyzing whether the gender population is affected more by the disease**

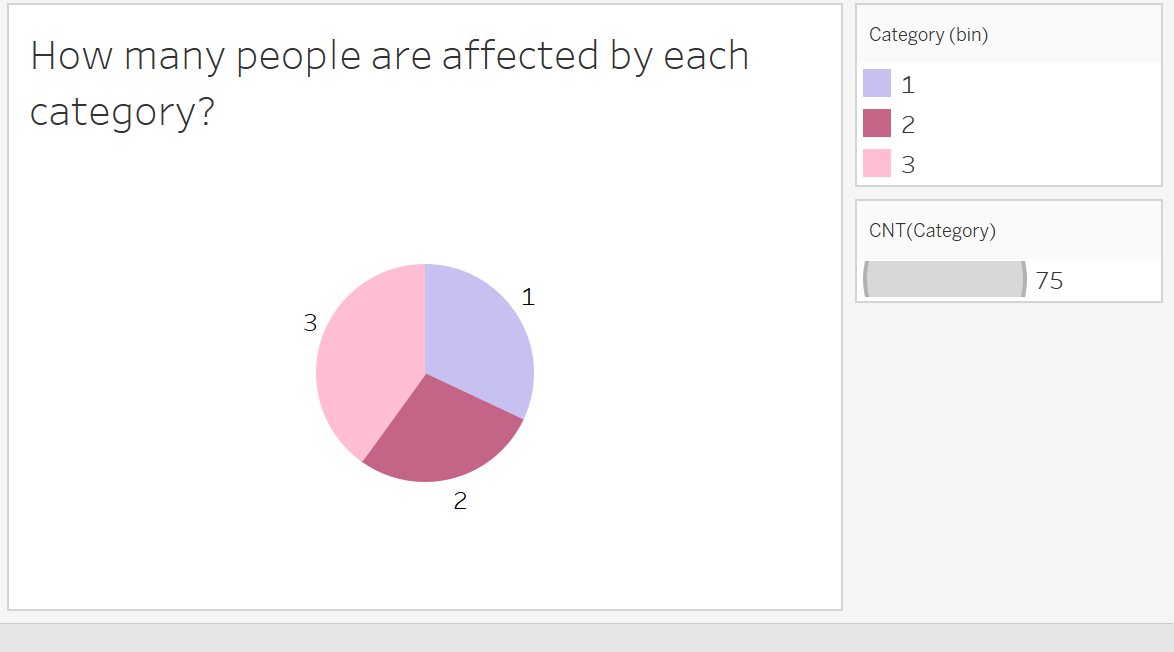
****

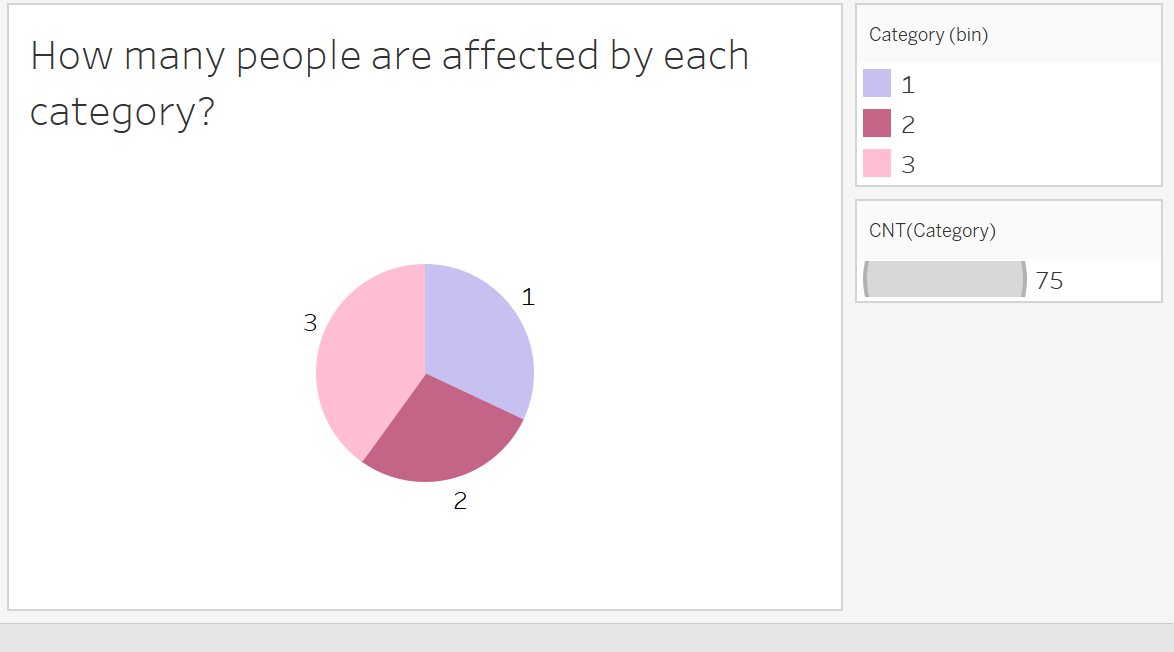
**Fig 5.2 Analyzing the variation of each age group of the population affected maximum by the disease**

****

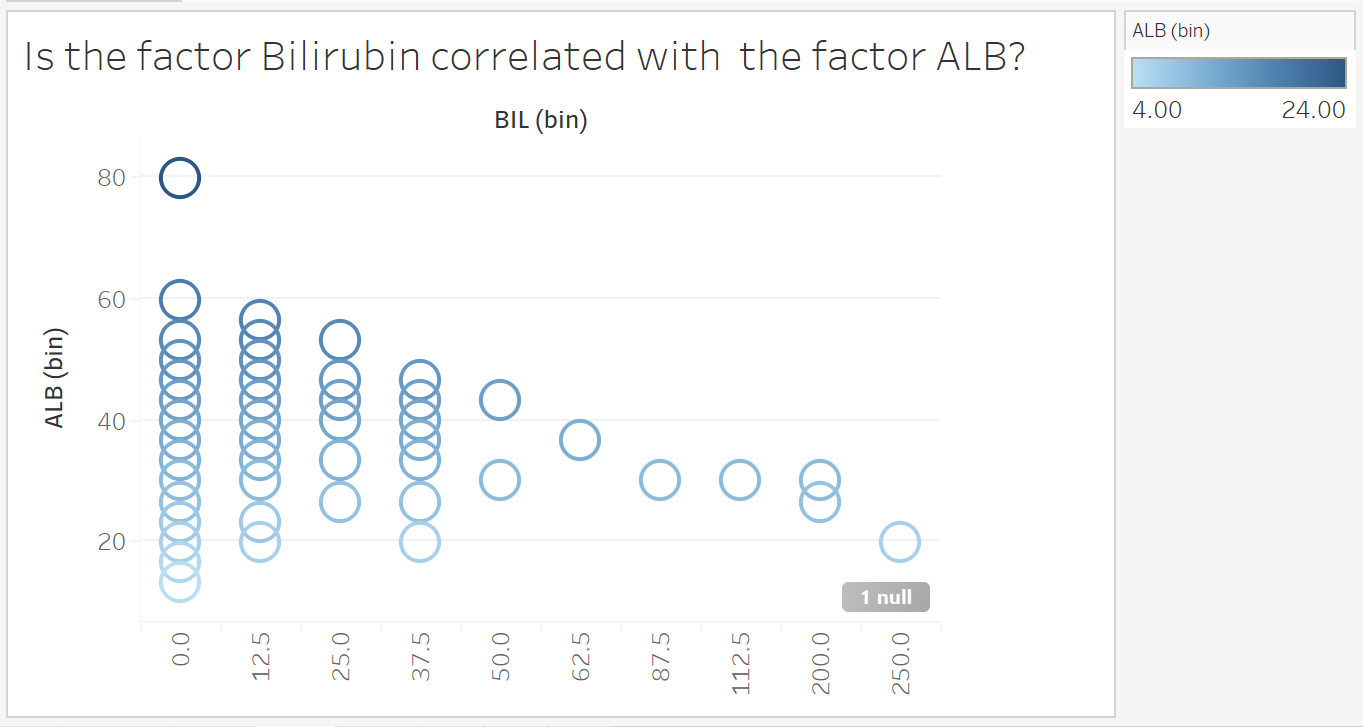
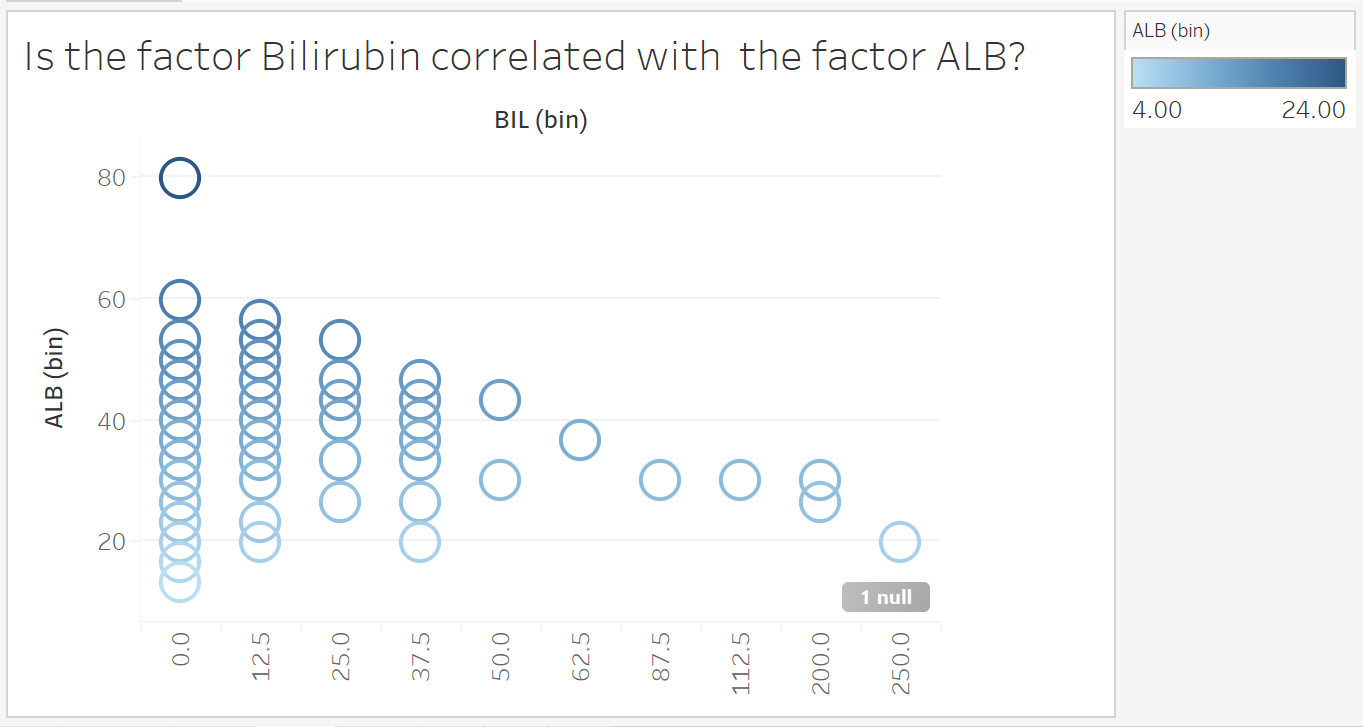
**Fig 5.3 Analyzing whether the category belongs to most of the population affected by the disease.**

****

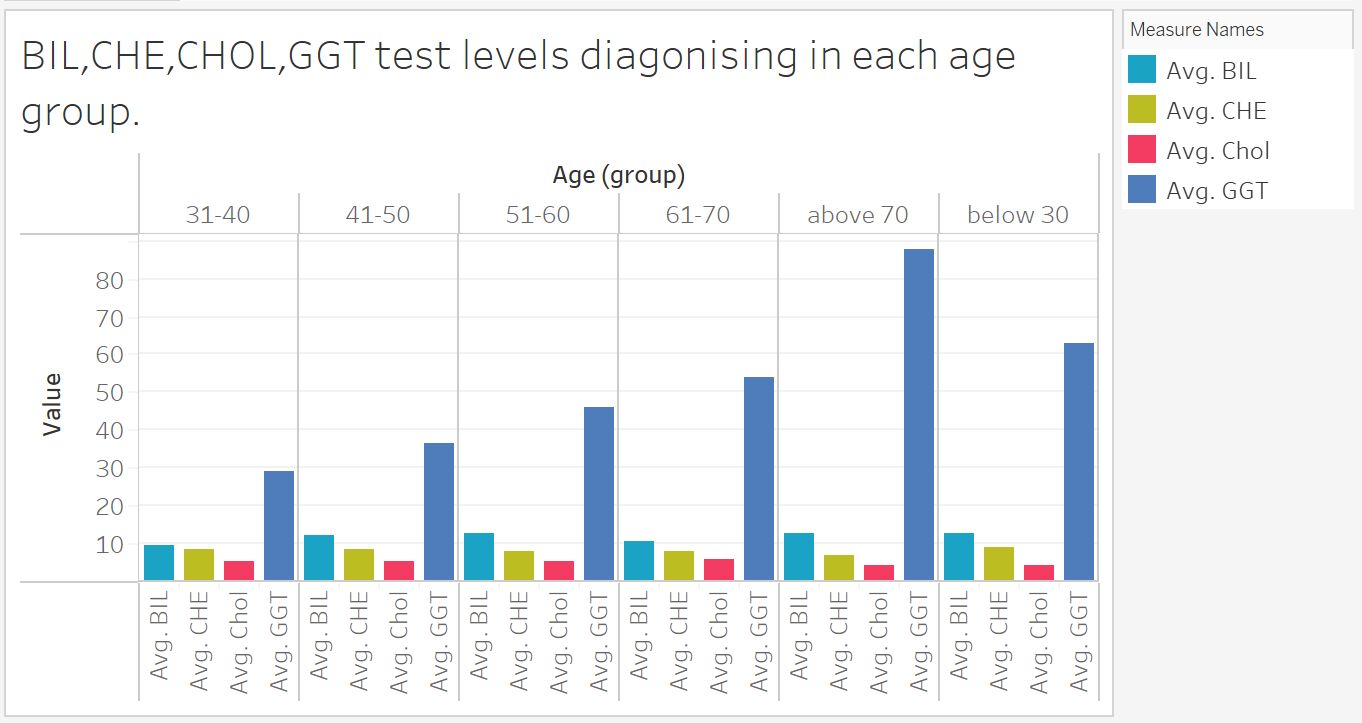
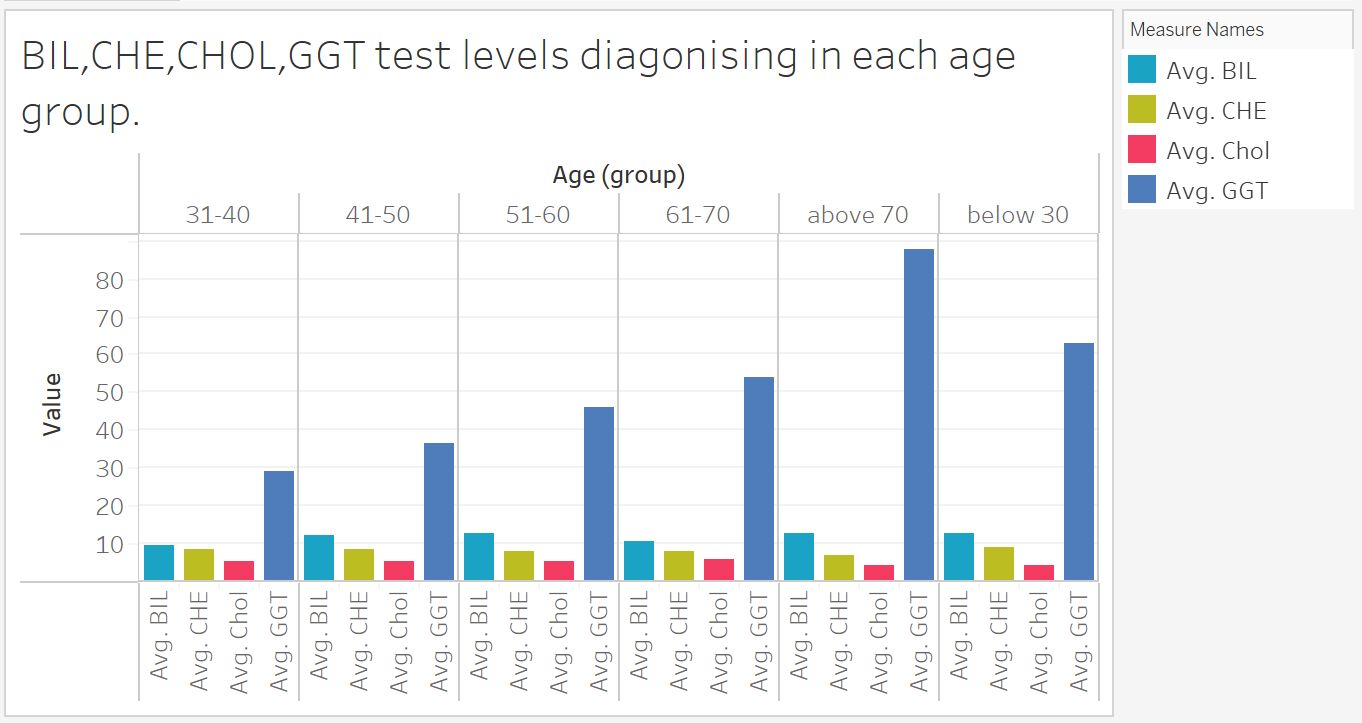
**Fig 5.4 Analyzing whether the factor Protein is correlated with the factor ALB**

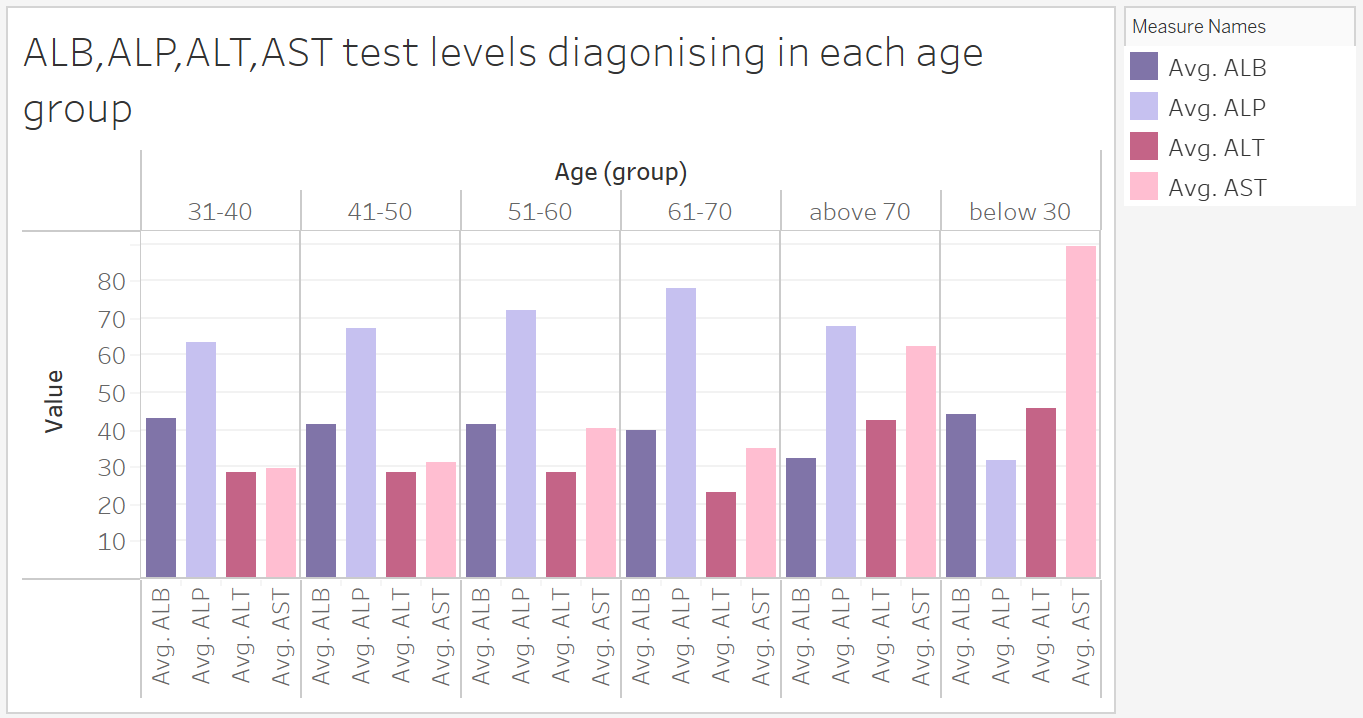
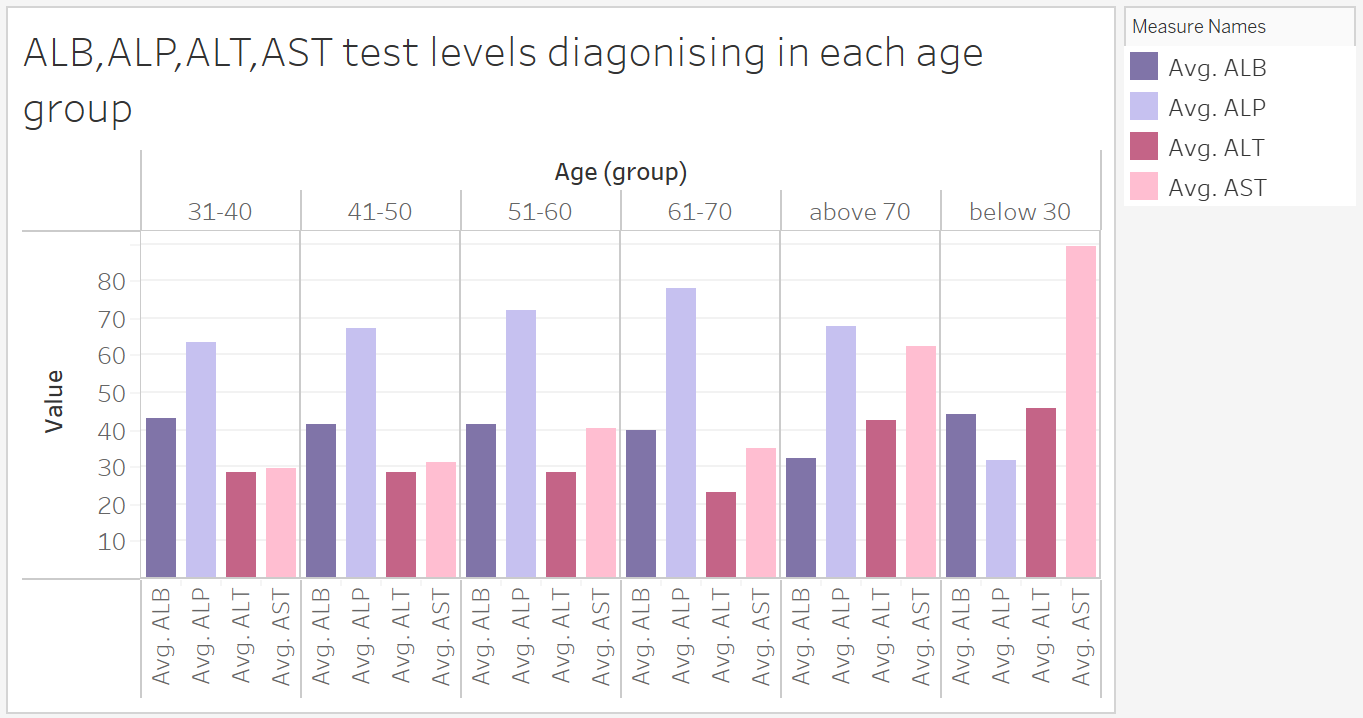
****

**Fig 5.5 Analyzing number of people affected by each category**

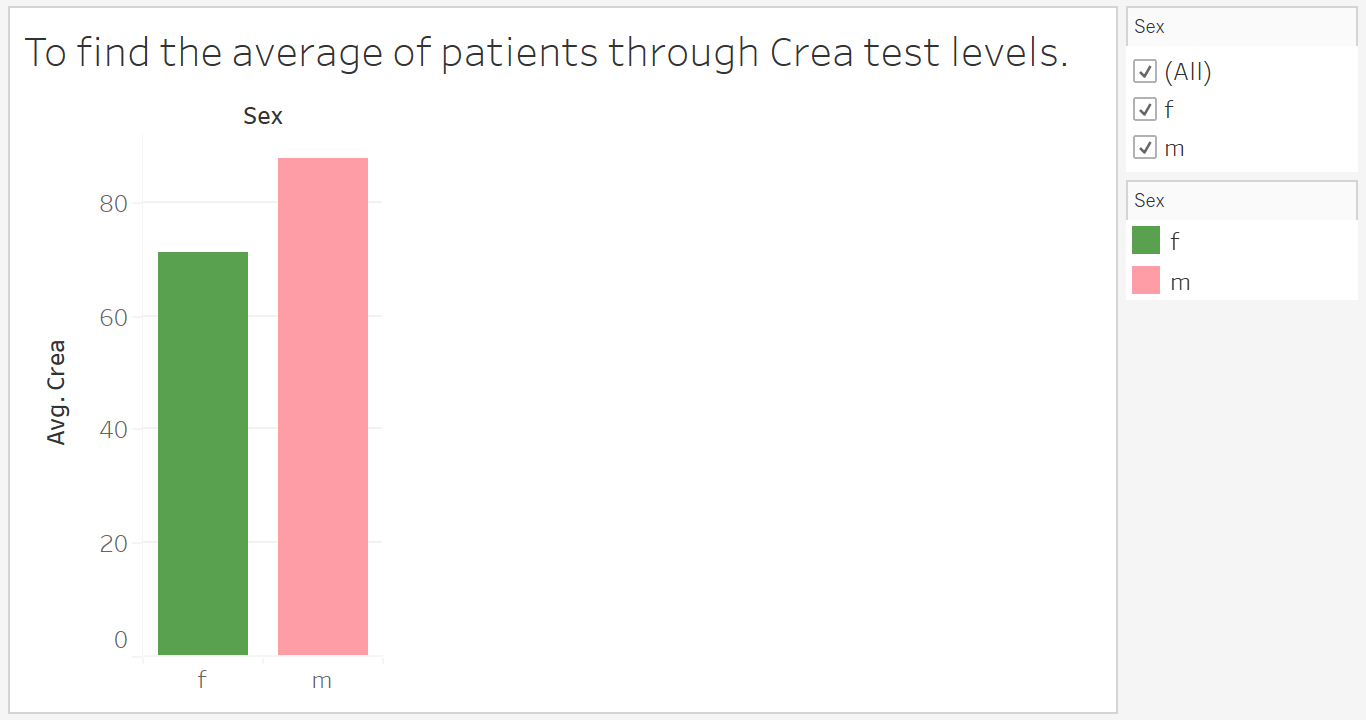
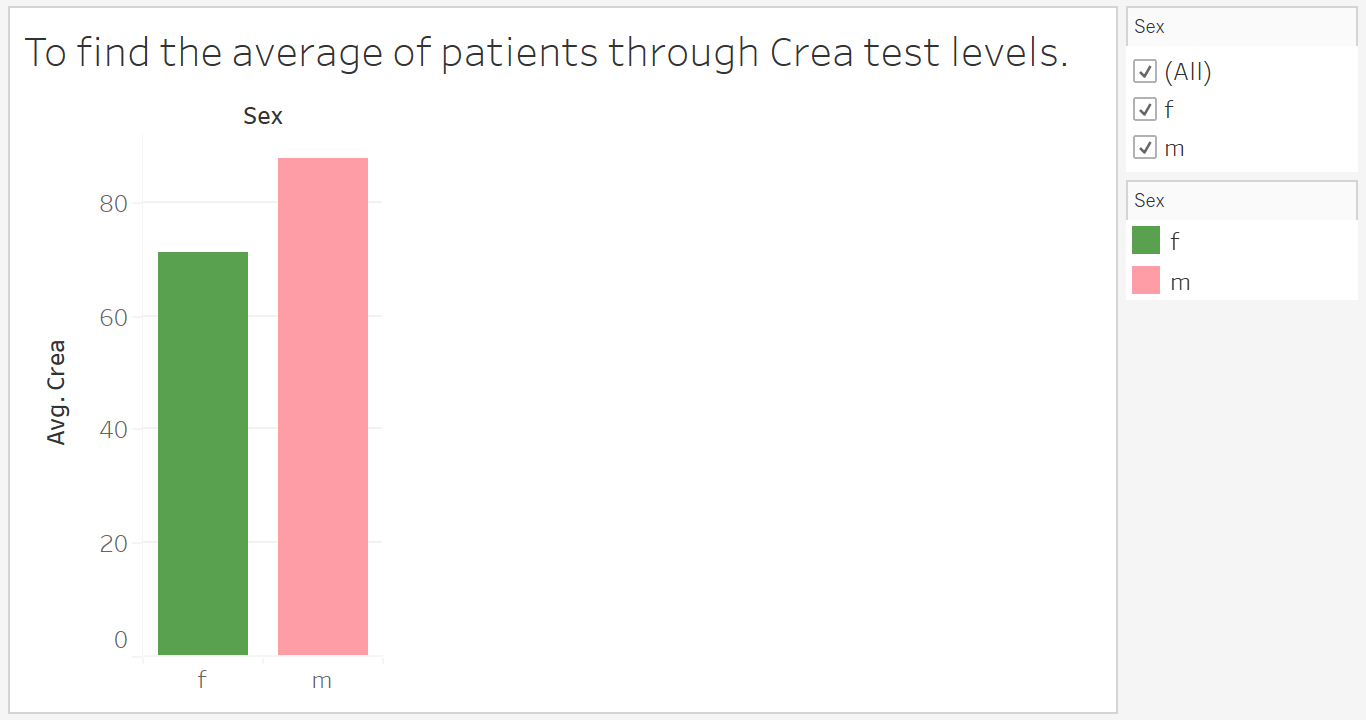
****

**Fig 5.6 Analyzing the factor Bilirubin correlated with the factor ALB**

****

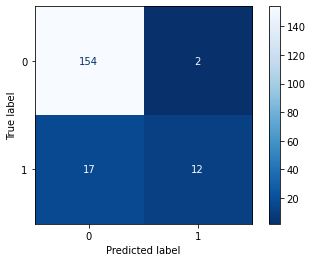
**Fig 5.7 BIL,CHE,CHOL,GGT test levels diagnosing in each age group**

**Fig 5.8 ALB,ALP,ALT,AST test levels diagnosing in each age group**

****

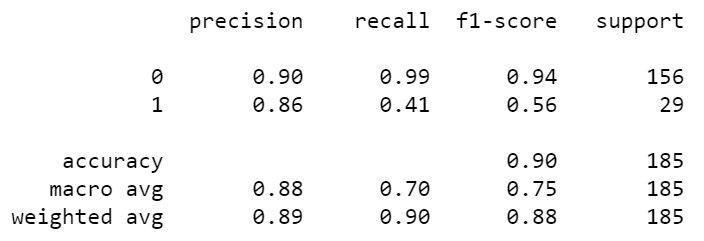
**Fig 5.9 Analyzing to find the average of patients through CREA test levels**

**LOGISTIC REGRESSION:**

* **Confusion Matrix:**

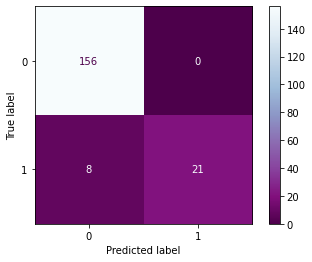
**Fig 5.10 Confusion matrix for Logistic Regression**

The following are the different Confusion matrix values analysed from Fig 5.10:

* The model properly categorized 154 positive class data items.
* The model properly identified 12 negative class data items.
* The model mistakenly identified two negative class data items as belonging to the positive class.
* The model wrongly identified 17 positive class data items as belonging to the negative class.
* **Classification Report:**

**Fig 5.11 Classification report for Logistic Regression**

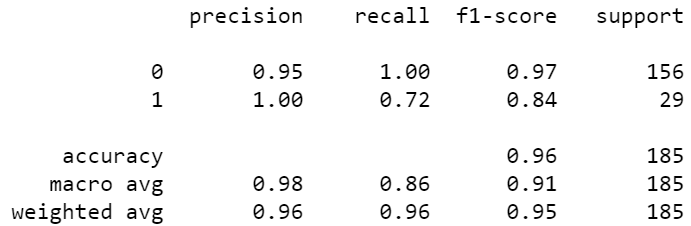
**RANDOM FOREST:**

* **Confusion Matrix:**

**Fig 5.12 Confusion matrix for Random Forest**

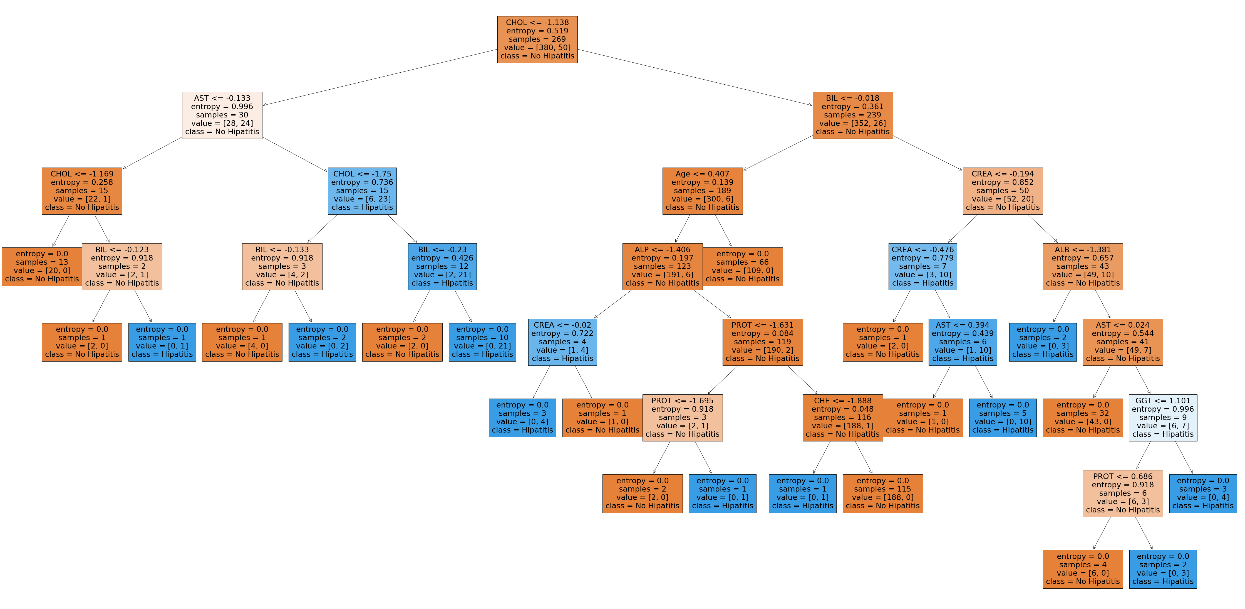
The following are the different Confusion matrix values analysed from Fig 5.12:

* The model properly categorized 156 positive class data items.
* The model properly identified 21 negative class data items.
* The model mistakenly identified zero negative class data items as belonging to the positive class.
* The model wrongly identified 8 positive class data items as belonging to the negative class.
* **Classification Report:**

****

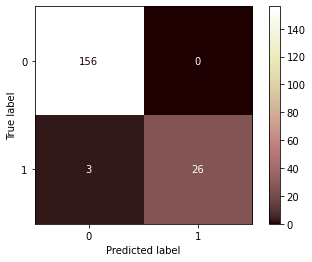
**Fig 5.13 Classification report for Random Forest**

* **Tree Plot for Random Forest Classifier**

****

**Fig 5.14 Tree plot for Random Forest Classifier**

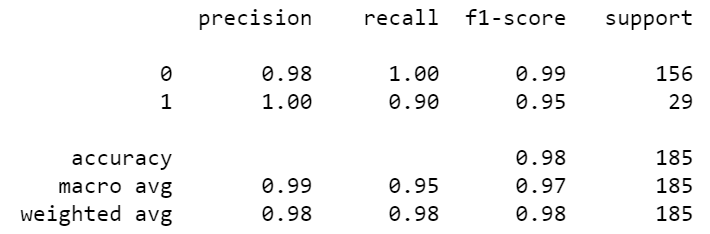
**ADA BOOST:**

* **Confusion Matrix:**

**Fig 5.15 Confusion matrix for Ada Boost**

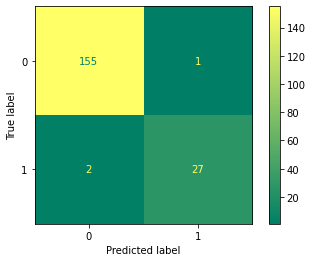
The following are the different Confusion matrix values analysed from Fig 5.15:

* The model properly categorized 156 positive class data items.
* The model properly identified 26 negative class data items.
* The model mistakenly identified zero negative class data items as belonging to the positive class.
* The model wrongly identified 3 positive class data items as belonging to the negative class.
* **Classification Report:**

****

**Fig 5.16 Classification report for Ada Boost**

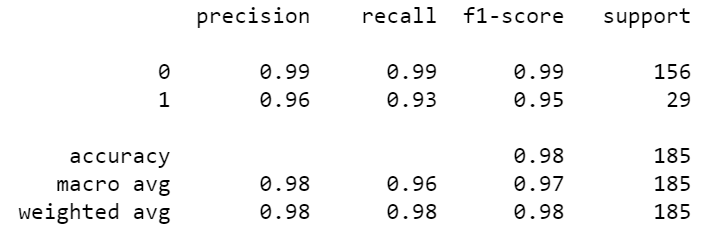
**XG BOOST:**

* **Confusion Matrix:**

**Fig 5.17 Confusion matrix for XG Boost**

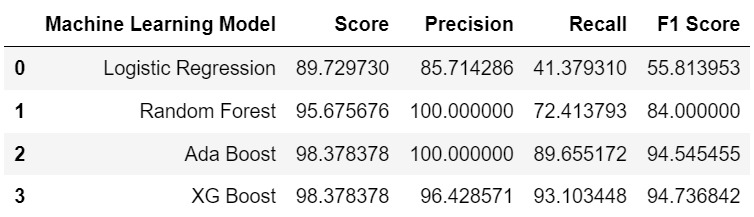
The following are the different Confusion matrix values analysed from Fig 5.17:

* The model properly categorized 155 positive class data items.
* The model properly identified 27 negative class data items.
* The model mistakenly identified one negative class data items as belonging to the positive class.
* The model wrongly identified 2 positive class data items as belonging to the negative class.
* **Classification Report:**

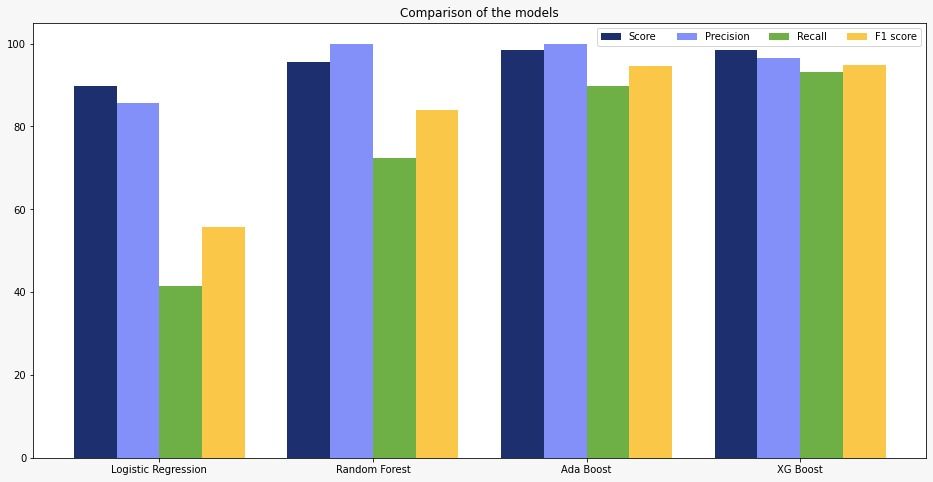
****

**Fig 5.18 Classification report for XG Boost**

**COMPARING THE MODELS:**

****

**Fig 5.19 Comparing the models**

****

**Fig 5.20 Visualization of models**

**CHAPTER 6**

**CONCLUSION AND FUTURE WORK**

**6.1 Conclusion:**

Multiple machine learning algorithms for predicting clinical outcomes in hepatitis C patients were compared in this study. "Logistic Regression," "Random Forest Classifier," "XG Boost," and "Ada Boost" were among the machine learning algorithms employed in this investigation. Several assessment criteria, including "F-measures," Precision, Accuracy, and Recall, are used to evaluate the performance of these algorithms. The accuracy metric for test data showed that the XG Boost Algorithm performed effectively.

**6.2 Future Scope:**

The following are the initiatives that we plan to pursue in the future:

* Explore various machine learning algorithms and implement the same in our project.
* Inquire more knowledge about Tableau and implement the different paths of it in the data visualization part.

**References:**

1. T. ValeriyT. IvanL. Dmitriy, “Machine Learning Model For Diagnosing The Stage Of Liver Fibrosis In Patients With Chronic Viral Hepatitis c,” *Preprints*, vol. 2021, 2021.
2. Khan, M. (2017). Drug side-effect prediction using machine learning methods.
3. Sara Omer Hussien, Sara Sir Elkhatem, Nisreen Osman, Ashraf Osman Ibrahim, “A Review of Data Mining Techniques for Diagnosing Hepatitis”, IEEE 2017, pg. no. 1-6
4. Yu, X. (2014). HIV drug resistant prediction and featured mutants selection using machine learning approaches.
5. A. Akella and S. Akella, “Applying machine learning to evaluate for fibrosis in chronic hepatitis c,” medRxiv, vol. 2020, 2020.
6. Huang, R., Rao, H., Yang, M., Gao, Y., Wang, J., Jin, Q., ... & Wei, L. (2020). Noninvasive measurements predict liver fibrosis well in hepatitis C virus patients after direct-acting antiviral therapy. Digestive diseases and sciences, 65(5), 1491-1500.
7. ElHefnawi, M., Abdalla, M., Ahmed, S., Elakel, W., Esmat, G., Elraziky, M., ... & Hassan, M. (2012, August). Accurate prediction of response to Interferon-based therapy in Egyptian patients with Chronic Hepatitis C using machine-learning approaches. In 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (pp. 771-778). IEEE.
8. Pawlotsky, J. M., Negro, F., Aghemo, A., Berenguer, M., Dalgard, O., Dusheiko, G., ... & European Association for the Study of the Liver. (2020). EASL recommendations on treatment of hepatitis C: final update of the series☆. Journal of Hepatology, 73(5), 1170-1218.
9. Matloob, F., Aftab, S., & Iqbal, A. (2019). A Framework for Software Defect Prediction Using Feature Selection and Ensemble Learning Techniques. International Journal of Modern Education & Computer Science, 11(12).
10. Ali, U., Aftab, S., Iqbal, A., Nawaz, Z., Bashir, M. S., & Saeed, M. A. (2020). Software Defect Prediction Using Variant based Ensemble Learning and Feature Selection Techniques. International Journal of Modern Education & Computer Science, 12(5).

**Worklog**

|  |  |  |
| --- | --- | --- |
| **Day** | **Date** | **Task Done** |
| Day 1 | 15-Jan-2022 | Gathered dataset from Kaggle and referred various websites and learnt about Hepatitis C. |
| Day 2 | 17-Jan-2022 | Understanding and Analysis of data from Hepatitis C dataset. |
| Day 3 | 18-Jan-2022 | Studied and leant about the columns (Test Levels) which affects the disease Hepatitis from the dataset |
| Day 4 | 21-Jan-2022 | Held a meeting with Internal Project Guide. |
| Day 5 | 22-Jan-2022 | Planed about the problem statement which has been implemented on the data visualization part |
| Day 6 | 25-Jan-2022 | Performed Data collection and Data cleaning where we replaced the NA values with mean of each column |
| Day 7 | 27-Jan-2022 | Started understanding the data and came to know about the features of each column. |
| Day 8 | 30-Jan-2022 | Started with the course ‘Tableau for Beginners’ in Udemy |
| Day 9 | 31-Jan-2022 | Started with the course ‘Tableau for Beginners’ in Udemy |
| Day 10 | 1-Feb-2022 | Held a meeting with Internal Project Guide. |
| Day 11 | 2-Feb-2022 | Visualized few objectives for the problem statement in Tableau and implemented the concepts which we have learnt from Tableau course. |
| Day 12 | 4-Feb-2022 | Visualized few objectives for the problem statement in Tableau and implemented the concepts which we have learnt from Tableau course. |
| Day 13 | 5-Feb-2022 | Preparing the required documents for ICET first review presentation. |
| Day 14 | 10-Feb-2022 | Implemented the knowledge acquired to choose the machine learning algorithms for classification models. |
| Day 15 | 11-Feb-2022 | Learnt about the mathematical background of logistic regression |
| Day 16 | 12-Feb-2022 | Performed the feature extraction and splitted the data into Dependent and Independent variables. |
| Day 17 | 13-Feb-2022 | Held a meeting with Internal Project Guide. |
| Day 18 | 14-Feb-2022 | Splitted the data into training and testing data to implement the classification algorithms. |
| Day 19 | 15-Feb-2022 | Fitted the Logistic Regression to a Training set. Predicted the test result for logistic regression and performed the confusion matrix. |
| Day 20 | 16-Feb-2022 | Compared the Y-Test and predicted value and created the classification report for the same. |
| Day 21 | 17-Feb-2022 | Visualized few objectives for the problem statement in Tableau and implemented the concepts which we have learnt from Tableau course. |
| Day 22 | 18-Feb-2022 | Learnt about the mathematical background of Random Forest. |
| Day 23 | 19-Feb-2022 | Fitted the Random Forest to a Training set. Predicted the test result for random forest and performed the confusion matrix. |
| Day 24 | 21-Feb-2022 | Compared the Y-Test and predicted value and created the classification report for the same. |
| Day 25 | 22-Feb-2022 | Visualized few objectives for the problem statement in Tableau and implemented the concepts which we have learnt from Tableau course. |
| Day 26 | 23-Feb-2022 | Learnt about the mathematical background of Ada Boost. |
| Day 27 | 24-Feb-2022 | Fitted the Ada Boost to a Training set. Predicted the test result for Ada Boost and performed the confusion matrix. |
| Day 28 | 28-Feb-2022 | Held a meeting with Internal Project Guide. |
| Day 29 | 1-Mar-2022 | Compared the Y-Test and predicted value and created the classification report for the same. |
| Day 30 | 2-Mar-2022 | Visualized few objectives for the problem statement in Tableau and implemented the concepts which we have learnt from Tableau course. |
| Day 31 | 3-Mar-2022 | Learnt about the mathematical background of XG Boost. |
| Day 32 | 4-Mar-2022 | Fitted the XG Boost to a Training set. Predicted the test result for XG Boost and performed the confusion matrix. |
| Day 33 | 5-Mar-2022 | Compared the Y-Test and predicted value and created the classification report for the same. |
| Day 34 | 6-Mar-2022 | Visualized few objectives for the problem statement in Tableau and implemented the concepts which we have learnt from Tableau course. |
| Day 35 | 7-Mar-2022 | Started journal for Hepatitis C disease Classification Algorithms |
| Day 36 | 10-Feb-2022 | Held a meeting with Internal Project Guide. |
| Day 37 | 14-Mar-2022 | Surveyed few journals for acquiring knowledge for journal purposes |
| Day 38 | 16-Mar-2022 | Completed Journal Paper |
| Day 39 | 21-Mar-2022 | Compared all the accuracy for each machine learning algorithm implemented and visualised it using plotly |
| Day 40 | 22-Mar-2022 | Project Completed |
| Day 41 | 23-Mar-2022 | Held a meeting with Internal Project Guide. |
| Day 42 | 24-Mar-2021 | Started preparing report for Hepatitis C Disease Classification Using Machine Learning Algorithms |
| Day 43 | 25-Mar-2021 | Working on preparing report for Hepatitis C Disease Classification Using Machine Learning Algorithms |
| Day 44 | 26-Mar-2021 | Started preparing final review presentation for Hepatitis C Disease Classification Using Machine Learning Algorithms |
| Day 45 | 27-Mar-2021 | Completed preparing final review presentation and report for Hepatitis C Disease Classification Using Machine Learning Algorithms |