Project Report

HEPATITIS C VIRUS FOR EGYPTIAN PATIENTS

INT254CA1 Project

by

HEMCHAND CHANDRAVANSHI

Section – KM098

Roll Number – RKM098B48

&

VIVEK SINGH

Section – KM098

Roll Number – RKM098B56

Department of Intelligent Systems

School of Computer Science Engineering

Lovely Professional University, Jalandhar

November – 2022

Student Declaration

This is to declare that this report has been written by us. No part of the report is copied from other sources. All information included from other sources have been duly acknowledged. We aver that if any part of the report is found to be copied, we are shall take full responsibility for it.

Hemchand

<<Hemchand Chandravanshi >>
<<Roll number: RKM098B48>>

<<Vivek Singh >>
<<Roll number: RKM098B56>>

Place: Lovely Professional University, Phagwara, Punjab

Date: 08 November 2022

TABLE OF CONTENTS

TITLE PAGE NO.

BONAFIDE CERTIFICATE

Certified that this project report “ HEPATITIS C VIRUS FOR EGYPTIAN PATIENTS ” is the bonafide work of “ HEMCHAND CHANDRAVANSHI and VIVEK SINGH” who carried out the project work under my supervision.

<<Signature of the Supervisor>>
<<Dr. Dhanpratap Singh >>
<<Lovely Professional University>>
<<ID : 25706>>
<< Computer Science and Engineering>>

Background and objectives of project assigned:

Background:

Hepatitis C being as a prevalent disease in the world especially in countries like Egypt. It is estimated that 3-4 million new cases every year, indicating as a public health problem and should be addressed with identification and treatment policies. In the initial stage, it is asymptomatic however when infection progress it leads to chronic conditions such as liver cirrhosis and hepatocellular carcinoma. Some of the various non-invasive serum biochemical markers are used to identify this disease. To construct a model to diagnose HCV and identify patients who have been infected with the virus, ML methods such as classification approaches can be used.

Motivation:

Sharps injuries, needles, and scalpels are a high-risk population for HCWs when performing their health-care tasks. As a result of caring for patients afflicted with HCV, HCWs are at risk of infection. This has prompted academics and researchers to develop a methodology for predicting HCV illness in HCWs at an early stage. The National Liver Institute (NLI), based at Menoufiya University in Menoufiya, Egypt, provided the HCV dataset for HCWs, which was used in the proposed study to construct the ML Model. Further the University of California, School of Information and Computer Science proposed the HCV-Egy-data.csv dataset to perform the discretization along with discretization criteria data set(Discretization_criteria.csv).

Objective:

The main objective of this project is to perform the Discretization on the dataset **(HCV-Egy-data.csv)** obtained from University of California for Egyptian patients who underwent treatment dosages for HCV about 18 months. Discretization should be applied based on expert recommendations; there is an attached file **(Discretization-criteria.csv)** for Discretization criteria also which shows how the discretization should be performed.

Discretization is the process through which we can transform continuous variables, models or functions into a discrete form. We do this by creating a set of contiguous intervals (or bins) that go across the range of our desired variable/model/function.

Whenever there is dataset with continuous variable it becomes harder to train the ML model so discretization is recommended for such variable. After performing the Discretization the dataset works well with the predictive models such as classification model.

Discretization-Criteria:

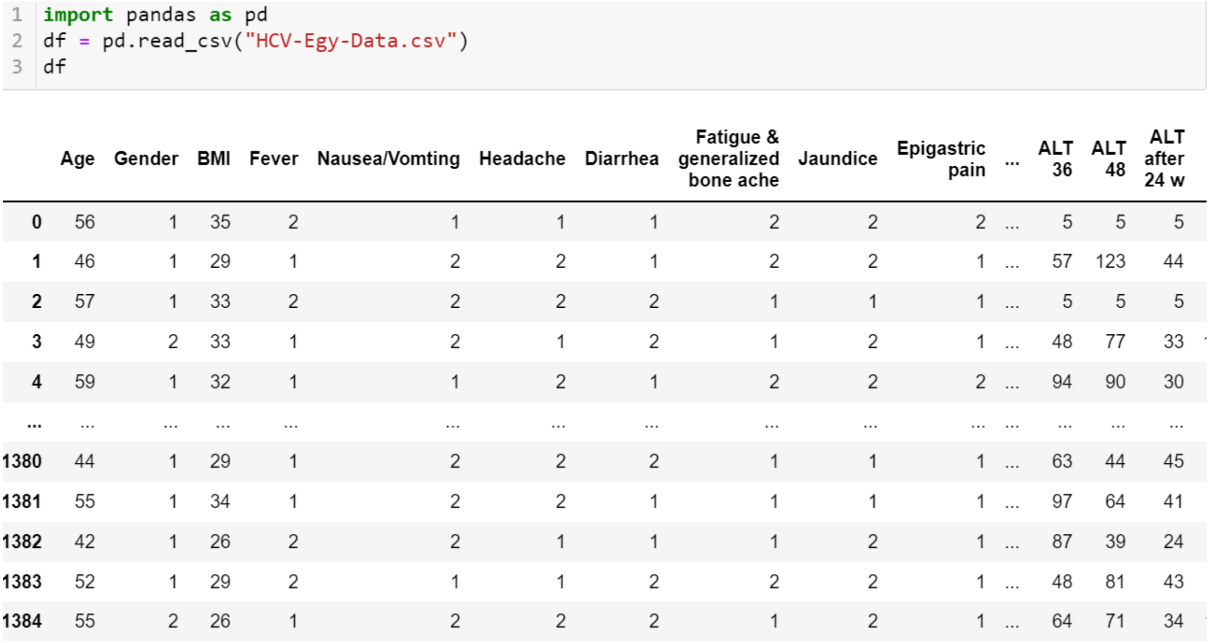
Feature Names	Feature Values	Discretization (Items)
Age	32:61	[0; 32],]32; 37],]37; 42],]42; 47],]47; 52],]52; 57],]57; 62]
Gender	Male,Female	[Male], [Female]
BMI(Body Mass Index)	22:35	[0; 18:5[[18:5; 25[, [25; 30[, [30; 35[, [35; 40[
Fever	Absent, Present	[Absent], [Present] -
Nausea/Vomiting	Absent, Present	[Absent], [Present] -
Headache	Absent, Present	[Absent], [Present] -
Diarrhea	Absent, Present	[Absent], [Present] -
Fatigue	Absent, Present	[Absent], [Present] -
Bone ache	Absent, Present	[Absent], [Present] -
Jaundice	Absent, Present	[Absent], [Present] -
Epigastria pain	Absent, Present	[Absent], [Present] -
WBC	2991:12101	[0; 4000[, [4000; 11000[, [11000; 12101]
RBC	3816422:5018451	[0;3000000[,[3000000;5000000[,[5000000;5018451]
HGB	02:20	If(Gender==[Male]):[2;14[,[14;17:5],]17:5;20] If(Gender==[Female]):[2;12:3[, [12:3; 15:3],]15:3; 20]
Plat(Platelet)	93013:226464	[93013; 100000[, [100000; 255000[,[255000; 226465[
AST1(1 week)	0.088888889	[0; 20[, [20; 40],]40; 128]
ALT1(1 week)	0.088888889	[0; 20[, [20; 40],]40; 128]
ALT4(4 weeks)	0.088888889	[0; 20[, [20; 40],]40; 128]
ALT12(12 weeks)	0.088888889	[0; 20[, [20; 40],]40; 128]
ALT24(24 weeks)	0.088888889	[0; 20[, [20; 40],]40; 128]
ALT36(36 weeks)	0.088888889	[0; 20[, [20; 40],]40; 128]
ALT48(48 weeks)	0.088888889	[0; 20[, [20; 40],]40; 128]
RNA Base	0:1201086	[0; 5],]5; 1201086]
RNA 4	0:1201715	[0; 5],]5; 1201715]
RNA 12	0:3731527	[0; 5],]5; 3731527]

RNA EOT	0:808450	[0; 5],]5; 808450]
RNA EF(Elongation Factor)	0:808450	[0; 5],]5; 808450]
Baseline Histological Grading	01:16	[1]; [2]; [3]; :::[16]
Baseline Histological staging	F0:F4	[No Fibrosis], [Portal Fibrosis], [Few Septa], [Many Septa], [Cirrhosis]

Outcomes:

We performed the Discretization using python various library, and we successfully got the Discretized data set as per expert recommendation and after this we have also build the ML model for predicting the Stage of Hepatitis c virus (**Baseline Histological staging**).

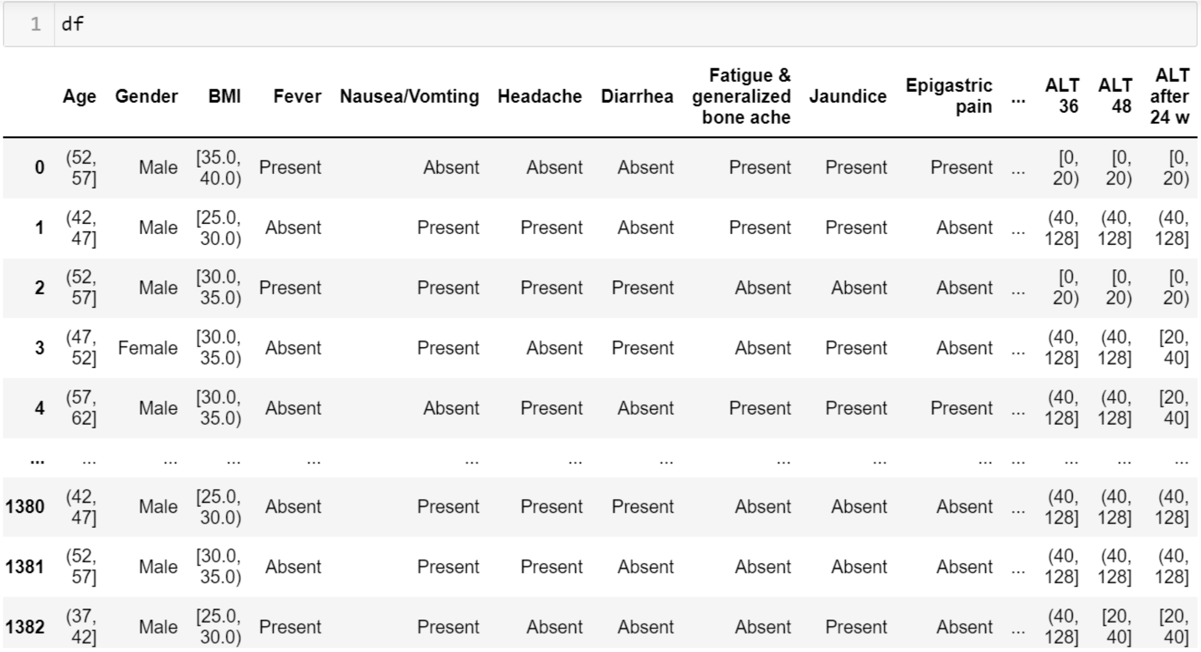
Dataset Before Discretization:



```python
import pandas as pd
df = pd.read_csv("HCV-Egy-Data.csv")
df
```

	Age	Gender	BMI	Fever	Nausea/Vomting	Headache	Diarrhea	Fatigue & generalized bone ache	Jaundice	Epigastric pain	...	ALT 36	ALT 48	ALT after 24 w
0	56	1	35	2	1	1	1	2	2	2	...	5	5	5
1	46	1	29	1	2	2	1	2	2	1	...	57	123	44
2	57	1	33	2	2	2	2	1	1	1	...	5	5	5
3	49	2	33	1	2	1	2	1	2	1	...	48	77	33
4	59	1	32	1	1	2	1	2	2	2	...	94	90	30
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1380	44	1	29	1	2	2	2	1	1	1	...	63	44	45
1381	55	1	34	1	2	2	1	1	1	1	...	97	64	41
1382	42	1	26	2	2	1	1	1	2	1	...	87	39	24
1383	52	1	29	2	1	1	2	2	2	1	...	48	81	43
1384	55	2	26	1	2	2	2	1	2	1	...	64	71	34

Dataset After Discretization:



```
df
```

	Age	Gender	BMI	Fever	Nausea/Vomting	Headache	Diarrhea	Fatigue & generalized bone ache	Jaundice	Epigastric pain	...	ALT 36	ALT 48	ALT after 24 w
0	(52, 57]	Male	[35.0, 40.0)	Present	Absent	Absent	Absent	Present	Present	Present	...	[0, 20)	[0, 20)	[0, 20)
1	(42, 47]	Male	[25.0, 30.0)	Absent	Present	Present	Absent	Present	Present	Absent	...	(40, 128]	(40, 128]	(40, 128]
2	(52, 57]	Male	[30.0, 35.0)	Present	Present	Present	Present	Absent	Absent	Absent	...	[0, 20)	[0, 20)	[0, 20)
3	(47, 52]	Female	[30.0, 35.0)	Absent	Present	Absent	Present	Absent	Present	Absent	...	(40, 128]	(40, 128]	[20, 40]
4	(57, 62]	Male	[30.0, 35.0)	Absent	Absent	Present	Absent	Present	Present	Present	...	(40, 128]	(40, 128]	[20, 40]
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1380	(42, 47]	Male	[25.0, 30.0)	Absent	Present	Present	Present	Absent	Absent	Absent	...	(40, 128]	(40, 128]	(40, 128]
1381	(52, 57]	Male	[30.0, 35.0)	Absent	Present	Present	Absent	Absent	Absent	Absent	...	(40, 128]	(40, 128]	(40, 128]
1382	(37, 42]	Male	[25.0, 30.0)	Present	Present	Absent	Absent	Absent	Present	Absent	...	(40, 128]	[20, 40]	[20, 40]

Description of Project

Libraries and Modules Used in Project:

We have used the Python Programming to build the project. Some python libraries used in project

1. **Pandas**:
   Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library. Pandas is fast and it has high performance & productivity for users. We have used pandas cut(), Interval() and many function to discrete the dataset into different interval.

2. **Matplotlib:**
   Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack.

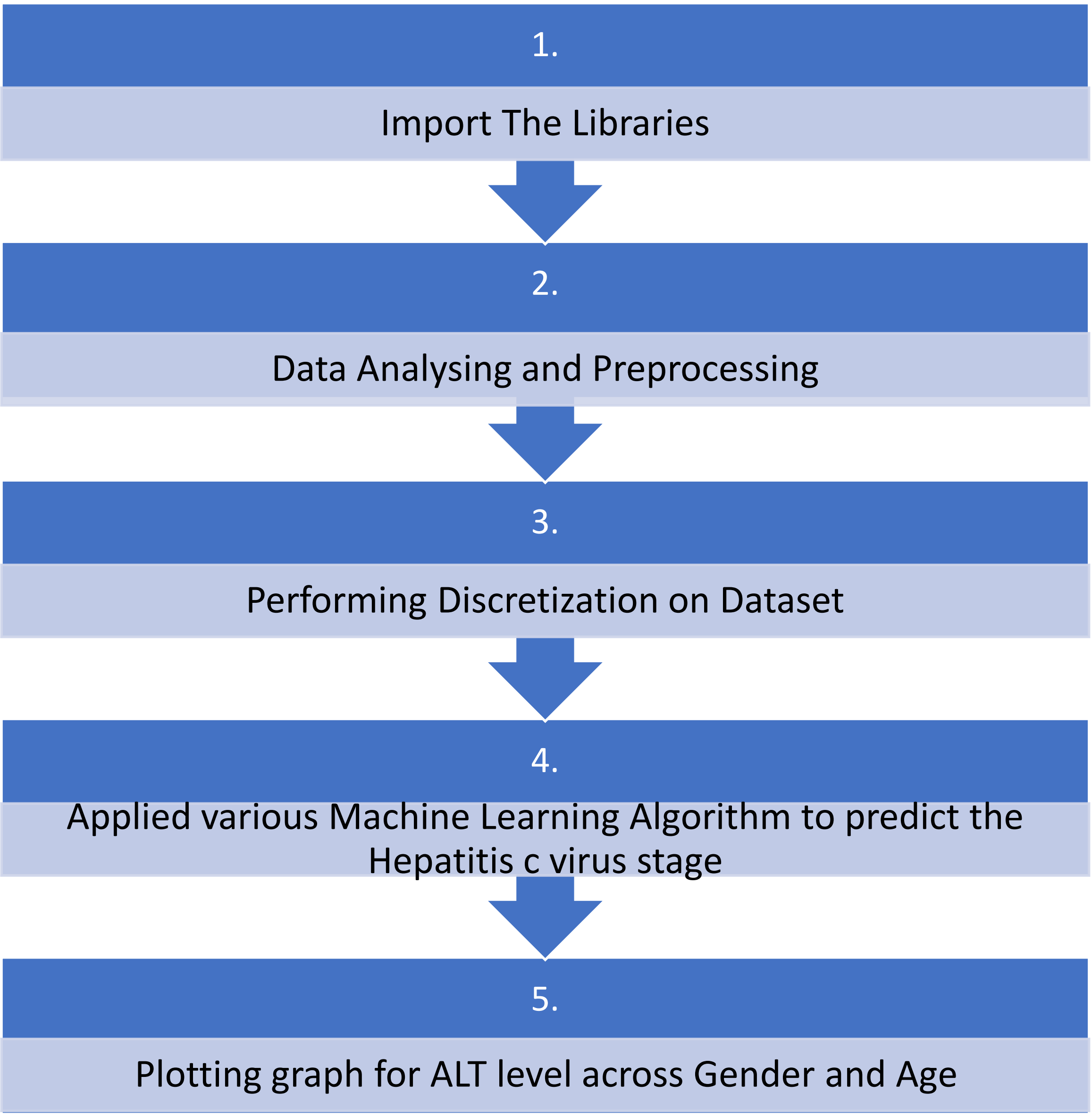
   One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc. we have used matplotlib to show the ALT level across gender and age.

3. **Seaborn:**
   Seaborn is an open-source Python library built on top of matplotlib. It is used for data visualization and exploratory data analysis. Seaborn works easily with dataframes and the Pandas library. The graphs created can also be customized easily. Below are a few benefits of Data Visualization.

4. **Scipy:**
   SciPy is a scientific computation library that uses NumPy underneath. SciPy stands for Scientific Python. It provides more utility functions for optimization, stats and signal processing. We have used scipy to find Pearson correlation coefficient.

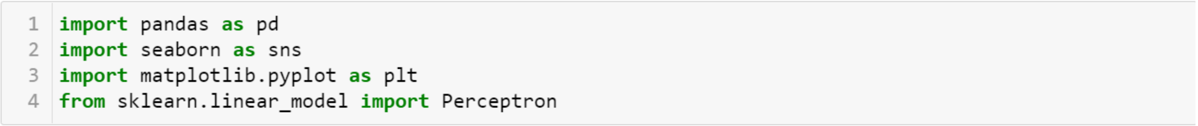
5. **Scikit-learn:**
   Scikit-learn is a machine learning library for Python. It features several regression, classification and clustering algorithms including SVMs, gradient boosting, k-means, random forests and DBSCAN. It is designed to work with Python Numpy and SciPy. The scikit-learn project kicked off as a Google Summer of Code (also known as GSoC) project by David Cournapeau as scikits.learn. It gets its name from “Scikit”, a separate third-party extension to SciPy.

Flow Chart

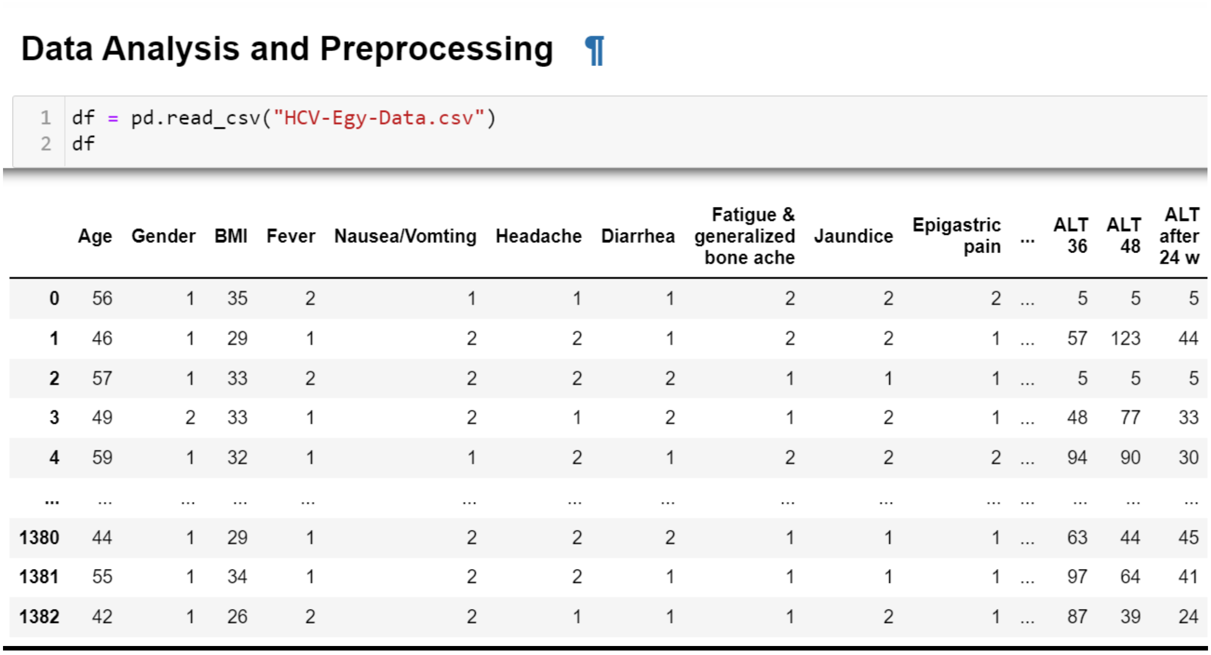

1.
Import The Libraries
2.
Data Analysing and Preprocessing
3.
Performing Discretization on Dataset
4.
Applied various Machine Learning Algorithm to predict the Hepatitis c virus stage
5.
Plotting graph for ALT level across Gender and Age

Implementation:

Impot The Libraries



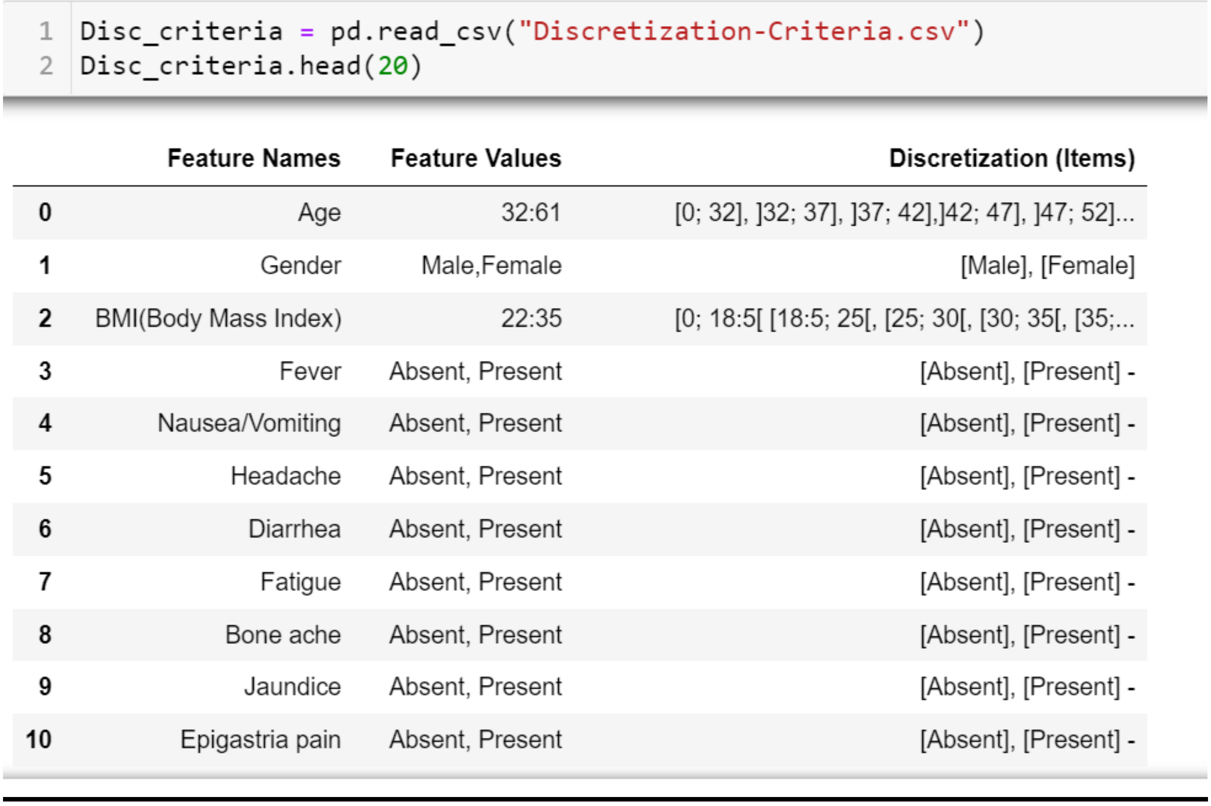
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import Perceptron
```



Data Analysis and Preprocessing ¶

```
df = pd.read_csv("HCV-Egy-Data.csv")
df
```

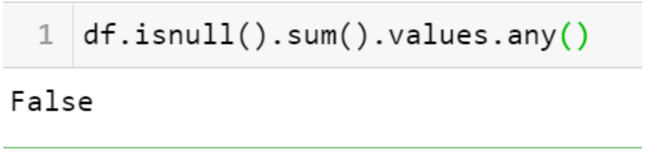
	Age	Gender	BMI	Fever	Nausea/Vomting	Headache	Diarrhea	Fatigue & generalized bone ache	Jaundice	Epigastric pain	...	ALT 36	ALT 48	ALT after 24 w
0	56	1	35	2	1	1	1	2	2	2	...	5	5	5
1	46	1	29	1	2	2	1	2	2	1	...	57	123	44
2	57	1	33	2	2	2	2	1	1	1	...	5	5	5
3	49	2	33	1	2	1	2	1	2	1	...	48	77	33
4	59	1	32	1	1	2	1	2	2	2	...	94	90	30
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1380	44	1	29	1	2	2	2	1	1	1	...	63	44	45
1381	55	1	34	1	2	2	1	1	1	1	...	97	64	41
1382	42	1	26	2	2	1	1	1	2	1	...	87	39	24



```
Disc_criteria = pd.read_csv("Discretization-Criteria.csv")
Disc_criteria.head(20)
```

	Feature Names	Feature Values	Discretization (Items)
0	Age	32:61	[0; 32],]32; 37],]37; 42],]42; 47],]47; 52]...
1	Gender	Male,Female	[Male], [Female]
2	BMI(Body Mass Index)	22:35	[0; 18:5[[18:5; 25[, [25; 30[, [30; 35[, [35;...
3	Fever	Absent, Present	[Absent], [Present] -
4	Nausea/Vomiting	Absent, Present	[Absent], [Present] -
5	Headache	Absent, Present	[Absent], [Present] -
6	Diarrhea	Absent, Present	[Absent], [Present] -
7	Fatigue	Absent, Present	[Absent], [Present] -
8	Bone ache	Absent, Present	[Absent], [Present] -
9	Jaundice	Absent, Present	[Absent], [Present] -
10	Epigastria pain	Absent, Present	[Absent], [Present] -

Checking For null values whether any columns contains the null values or not If it contains null values then remove the null values using various null removal techniques like – using mod, median, mean etc, removal of null is necessary or else the various predictive algorithm will not compute on time.

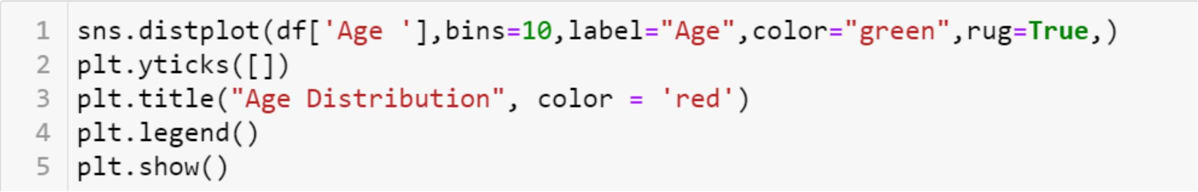


```
df.isnull().sum().values.any()
```

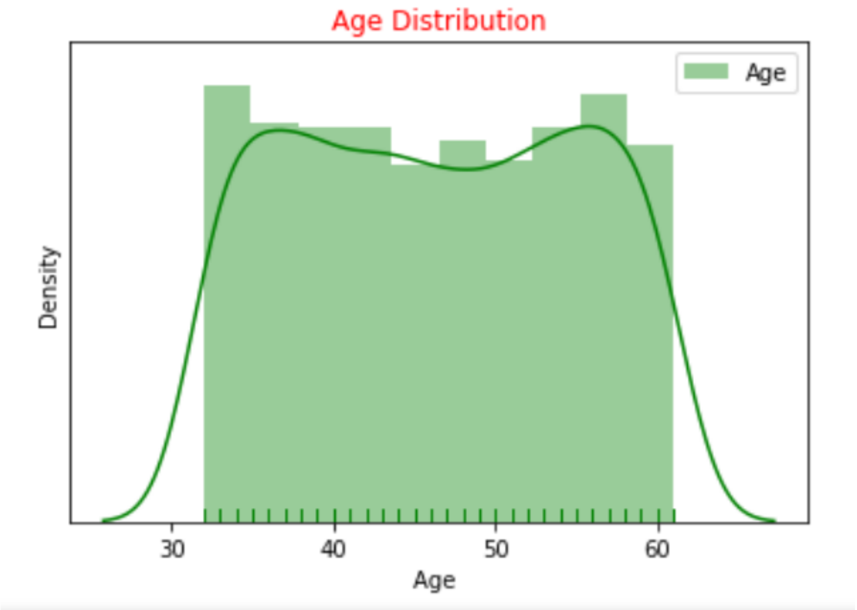
```
False
```

Above representation shows that our dataset does not contains any null values.

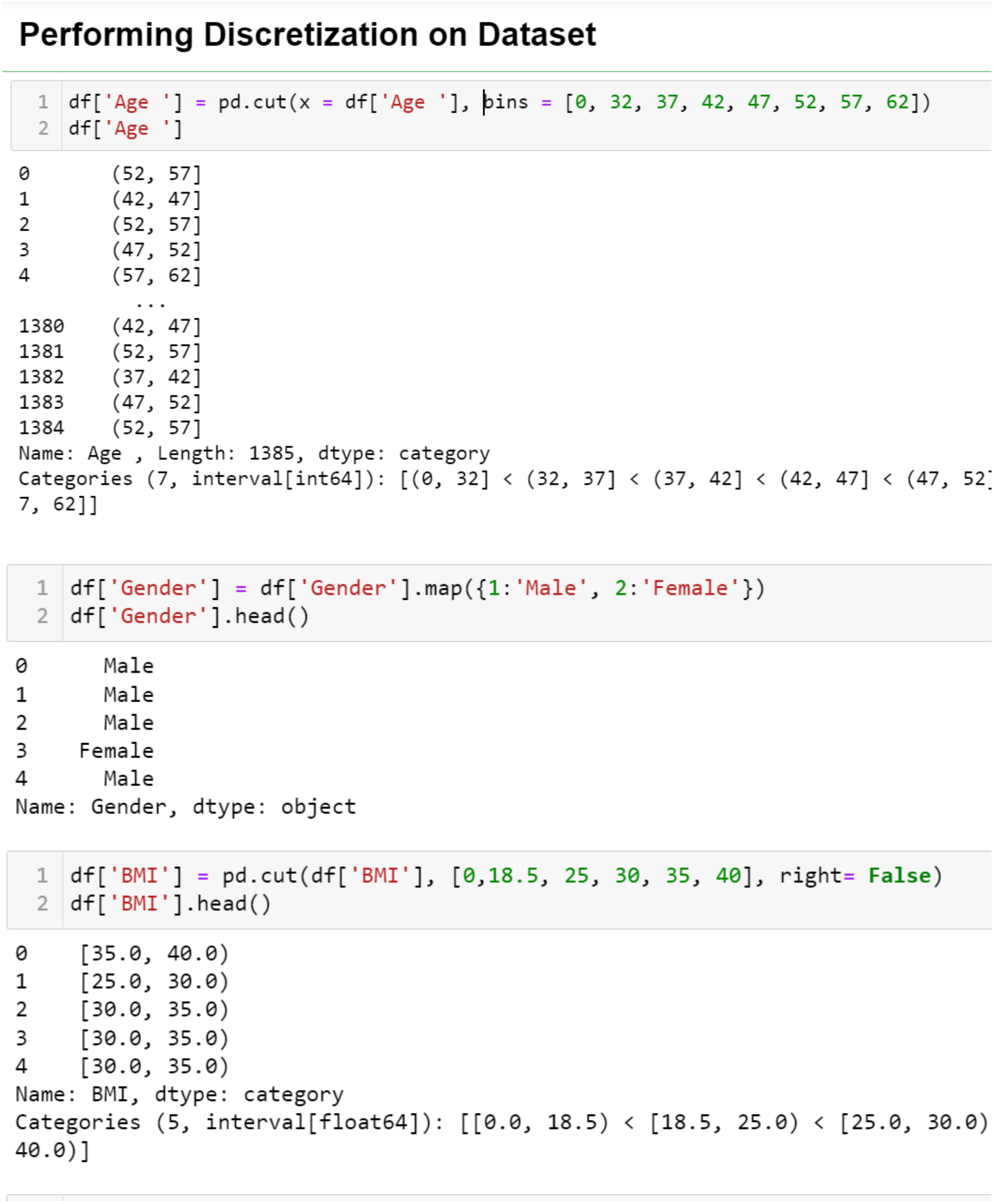
Distribution of Age



```
sns.distplot(df['Age '],bins=10,label="Age",color="green",rug=True,)
plt.yticks([])
plt.title("Age Distribution", color = 'red')
plt.legend()
plt.show()
```



The above outcome is only for one column that is Age, we can see that the values of column Age is almost continuous, this kind of data is not good for classification model, and here all the columns of this dataset is almost continuous only so we need to discretize the data set. And to discretize it we have used the pandas cut() and Interval() modules.



Performing Discretization on Dataset

```
df['Age '] = pd.cut(x = df['Age '], bins = [0, 32, 37, 42, 47, 52, 57, 62])
df['Age ']
```

```
0       (52, 57]
1       (42, 47]
2       (52, 57]
3       (47, 52]
4       (57, 62]
          ...   
1380    (42, 47]
1381    (52, 57]
1382    (37, 42]
1383    (47, 52]
1384    (52, 57]
Name: Age , Length: 1385, dtype: category
Categories (7, interval[int64]): [(0, 32] < (32, 37] < (37, 42] < (42, 47] < (47, 52
7, 62]]
```

```
df['Gender'] = df['Gender'].map({1:'Male', 2:'Female'})
df['Gender'].head()
```

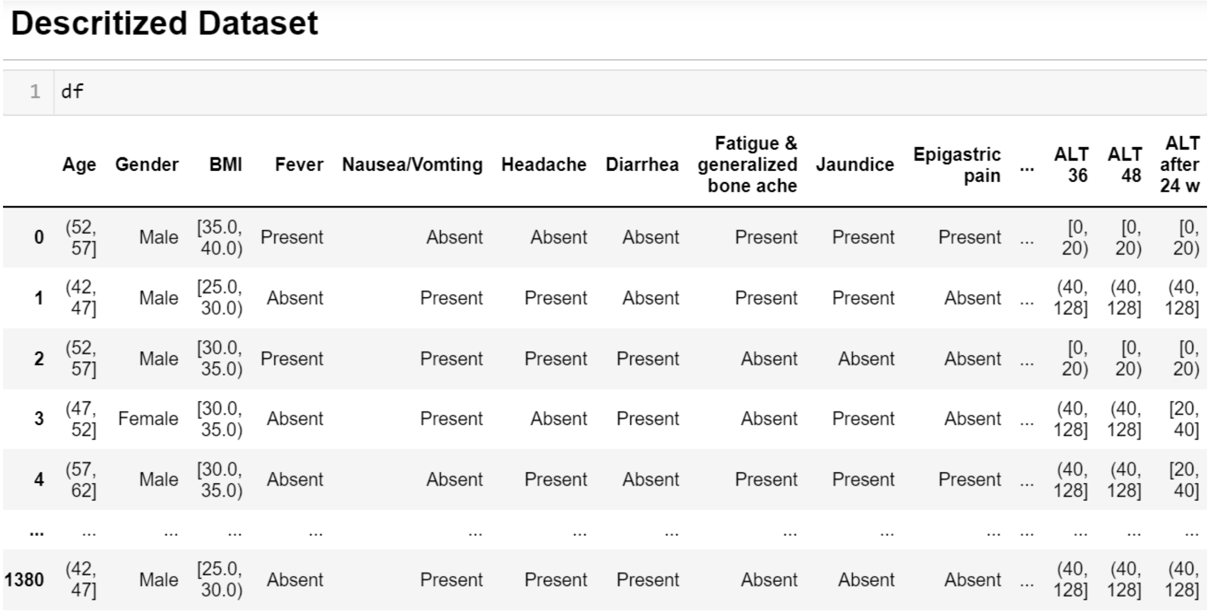
```
0      Male
1      Male
2      Male
3    Female
4      Male
Name: Gender, dtype: object
```

```
df['BMI'] = pd.cut(df['BMI'], [0,18.5, 25, 30, 35, 40], right= False)
df['BMI'].head()
```

```
0    [35.0, 40.0)
1    [25.0, 30.0)
2    [30.0, 35.0)
3    [30.0, 35.0)
4    [30.0, 35.0)
Name: BMI, dtype: category
Categories (5, interval[float64]): [[0.0, 18.5) < [18.5, 25.0) < [25.0, 30.0)
40.0)]
```

Here in this report we have shown the discretization for some of the columns like ‘Age’, ‘Gender’, and ‘BMI’, similar operation is performed in rest of the columns in its original code file, some of the columns needs manual discretization, we performed that also and now you can see that the data is discretized in various interval range and it will act as discretised data set.

Final Dataset after Discretization



Descritized Dataset

```
df
```

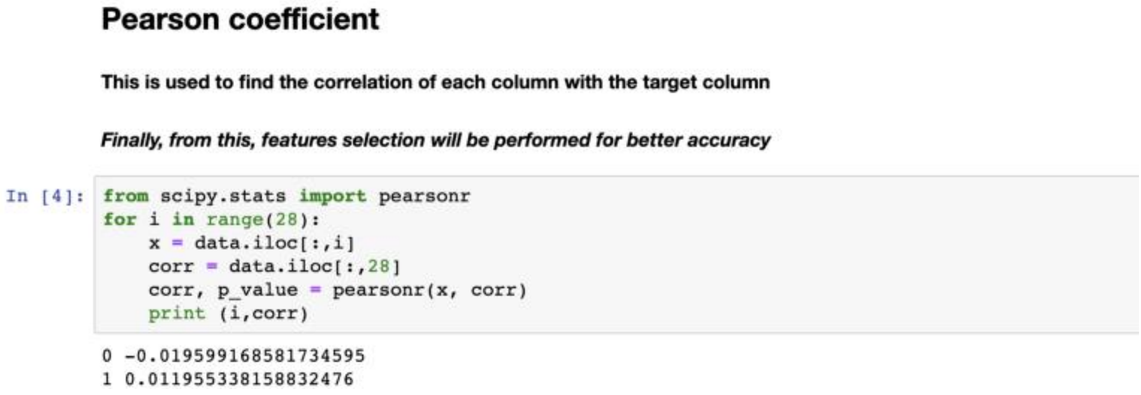
	Age	Gender	BMI	Fever	Nausea/Vomting	Headache	Diarrhea	Fatigue & generalized bone ache	Jaundice	Epigastric pain	...	ALT 36	ALT 48	ALT after 24 w
0	(52, 57]	Male	[35.0, 40.0)	Present	Absent	Absent	Absent	Present	Present	Present	...	[0, 20)	[0, 20)	[0, 20)
1	(42, 47]	Male	[25.0, 30.0)	Absent	Present	Present	Absent	Present	Present	Absent	...	(40, 128]	(40, 128]	(40, 128]
2	(52, 57]	Male	[30.0, 35.0)	Present	Present	Present	Present	Absent	Absent	Absent	...	[0, 20)	[0, 20)	[0, 20)
3	(47, 52]	Female	[30.0, 35.0)	Absent	Present	Absent	Present	Absent	Present	Absent	...	(40, 128]	(40, 128]	[20, 40]
4	(57, 62]	Male	[30.0, 35.0)	Absent	Absent	Present	Absent	Present	Present	Present	...	(40, 128]	(40, 128]	[20, 40]
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1380	(42, 47]	Male	[25.0, 30.0)	Absent	Present	Present	Present	Absent	Absent	Absent	...	(40, 128]	(40, 128]	(40, 128]

Prediction using various Machine Learning Algorithms:

Data Preprocessing

We will perform feature selection for better accuracy. Although, since the dataset is very small as compared to its values and requirements, which gives less accuracy if feature selection is not performed, then accuracy is kind of similar (but less).

We are going to use the **Pearson coefficient** for finding the correlation between the target column and other corresponding columns.



Pearson coefficient

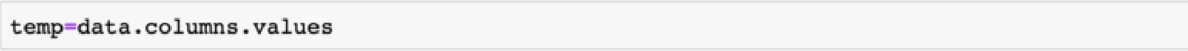
This is used to find the correlation of each column with the target column

Finally, from this, features selection will be performed for better accuracy

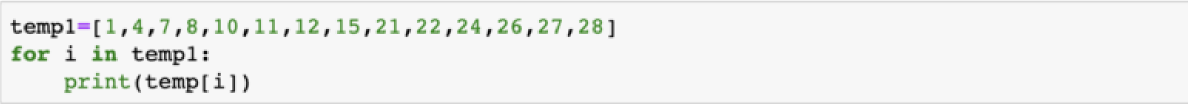
```
In [4]: from scipy.stats import pearsonr
        for i in range(28):
            x = data.iloc[:,i]
            corr = data.iloc[:,28]
            corr, p_value = pearsonr(x, corr)
            print (i,corr)
```

```
0 -0.019599168581734595
1 0.011955338158832476
```

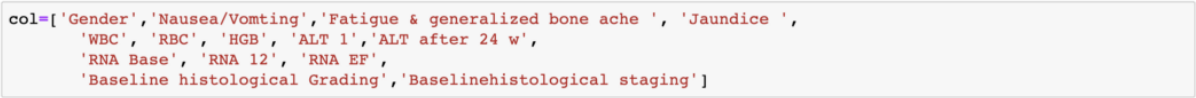
This gives us a list of correlation values ranging between -1 and 1. Through this we will make a new data frame having a positive correlation, hence improving the accuracy further.



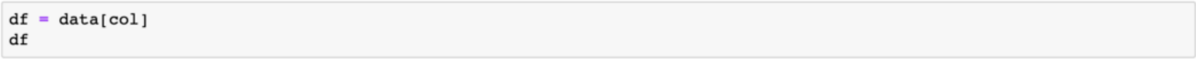
```
temp=data.columns.values
```



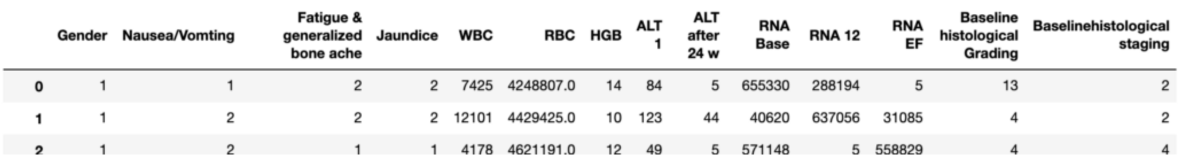
```
temp1=[1,4,7,8,10,11,12,15,21,22,24,26,27,28]
for i in temp1:
    print(temp[i])
```



```
col=['Gender','Nausea/Vomting','Fatigue & generalized bone ache ', 'Jaundice ',
       'WBC', 'RBC', 'HGB', 'ALT 1','ALT after 24 w',
       'RNA Base', 'RNA 12', 'RNA EF',
       'Baseline histological Grading','Baselinehistological staging']
```



```
df = data[col]
df
```

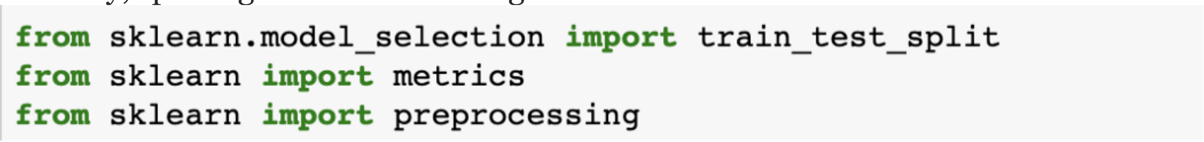


	Gender	Nausea/Vomting	Fatigue & generalized bone ache	Jaundice	WBC	RBC	HGB	ALT 1	ALT after 24 w	RNA Base	RNA 12	RNA EF	Baseline histological Grading	Baselinehistological staging
0	1	1	2	2	7425	4248807.0	14	84	5	655330	288194	5	13	2
1	1	2	2	2	12101	4429425.0	10	123	44	40620	637056	31085	4	2
2	1	2	1	1	4178	4621191.0	12	49	5	571148	5	558829	4	4

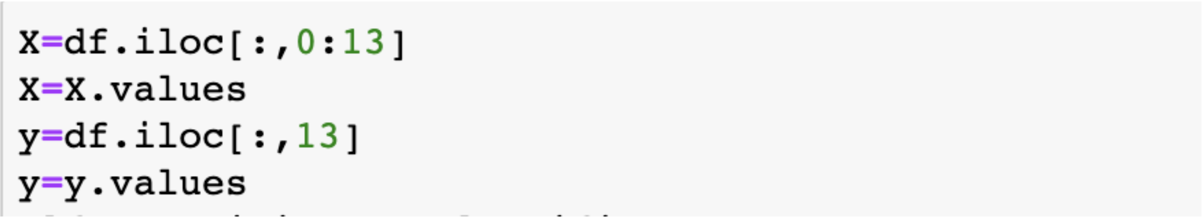
After making a new data frame with relevant columns and target columns (Baselinehistological staging), its time for splitting the dataset into a training dataset and testing dataset.

Splitting dataset into train and test set

This is done using the sklearn package. Also, we will import a few more libraries for accuracy, splitting the dataset and algorithms.



```
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn import preprocessing
```

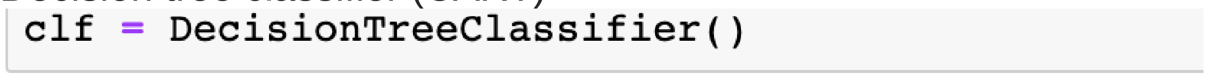


```
X=df.iloc[:,0:13]
X=X.values
y=df.iloc[:,13]
y=y.values
```

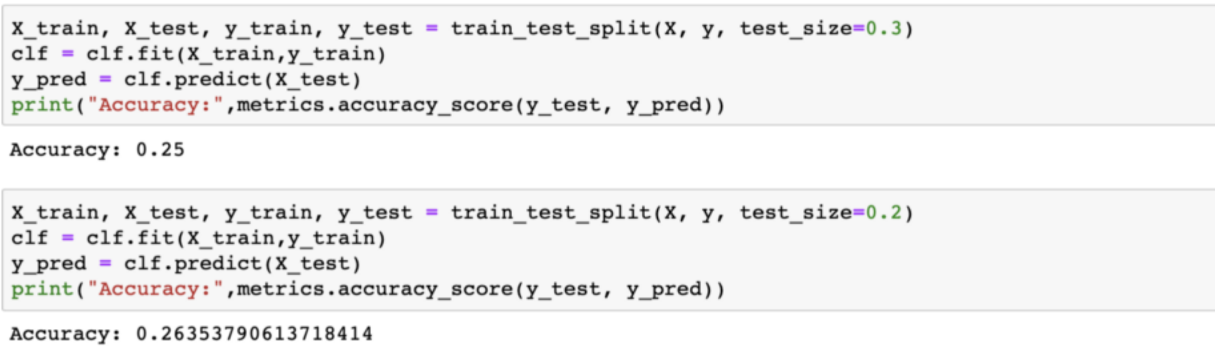
Implementing Machine Learning Algorithms (models)

We have experimented with different sizes of testing and training sets. These are represented in fractions(0.3 represents 30% of the dataset is testing dataset and 70% dataset is testing dataset and so on).

Decision tree classifier (CART)



```
clf = DecisionTreeClassifier()
```

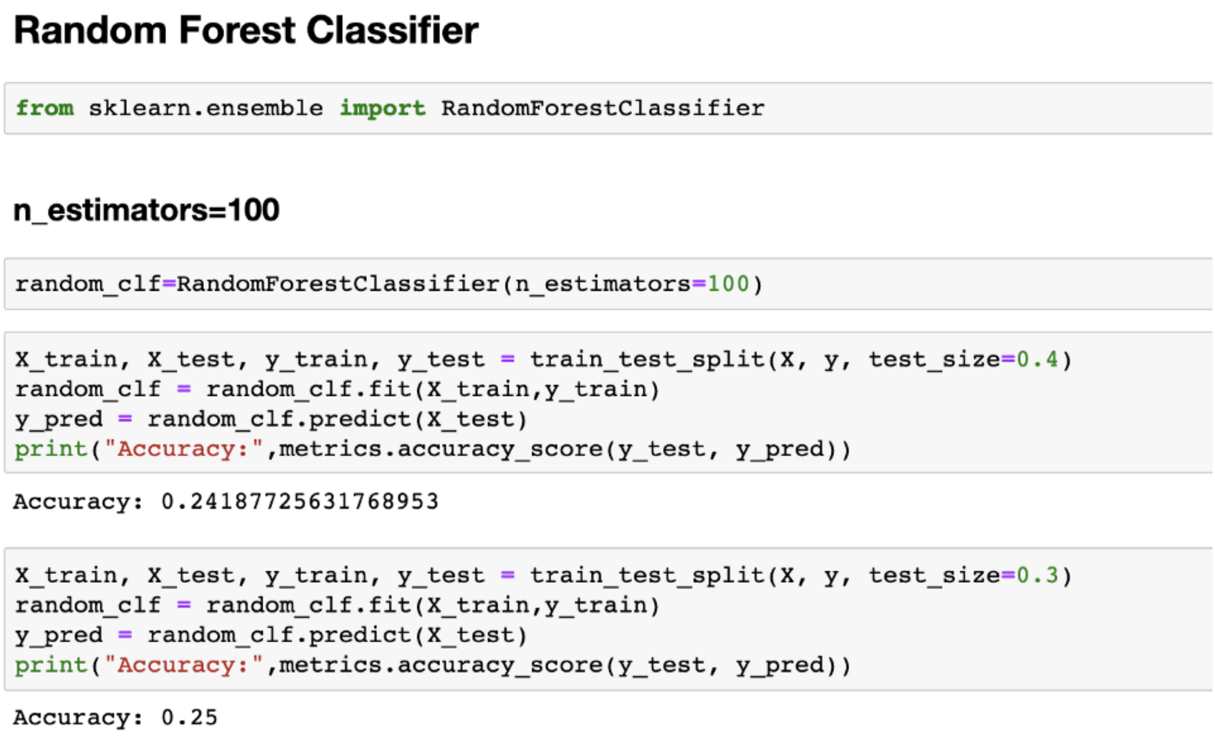


```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
clf = clf.fit(X_train,y_train)
y_pred = clf.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.25
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
clf = clf.fit(X_train,y_train)
y_pred = clf.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.26353790613718414
```



Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier
```

n_estimators=100

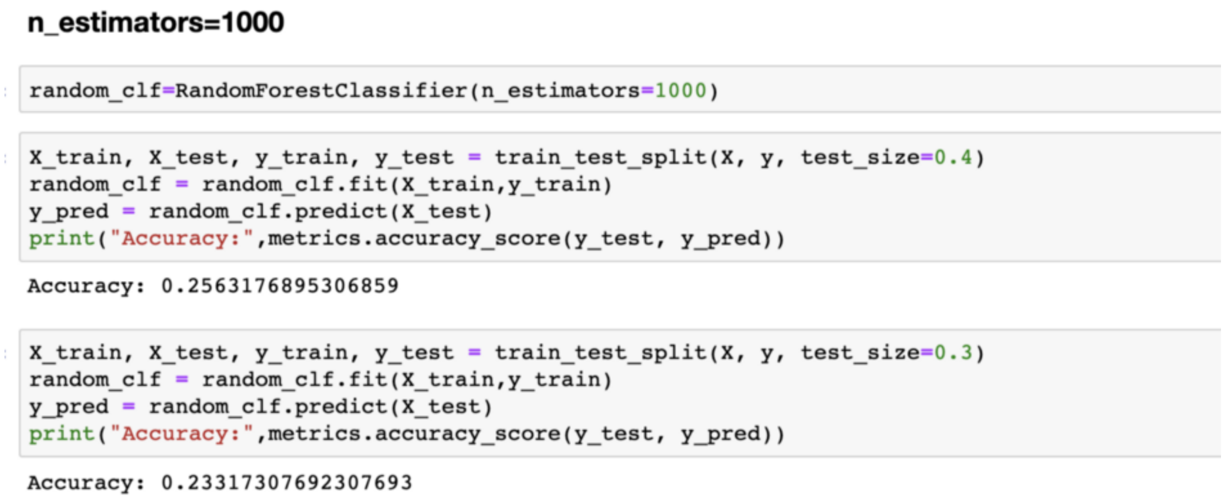
```
random_clf=RandomForestClassifier(n_estimators=100)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4)
random_clf = random_clf.fit(X_train,y_train)
y_pred = random_clf.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.24187725631768953
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
random_clf = random_clf.fit(X_train,y_train)
y_pred = random_clf.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.25
```



n_estimators=1000

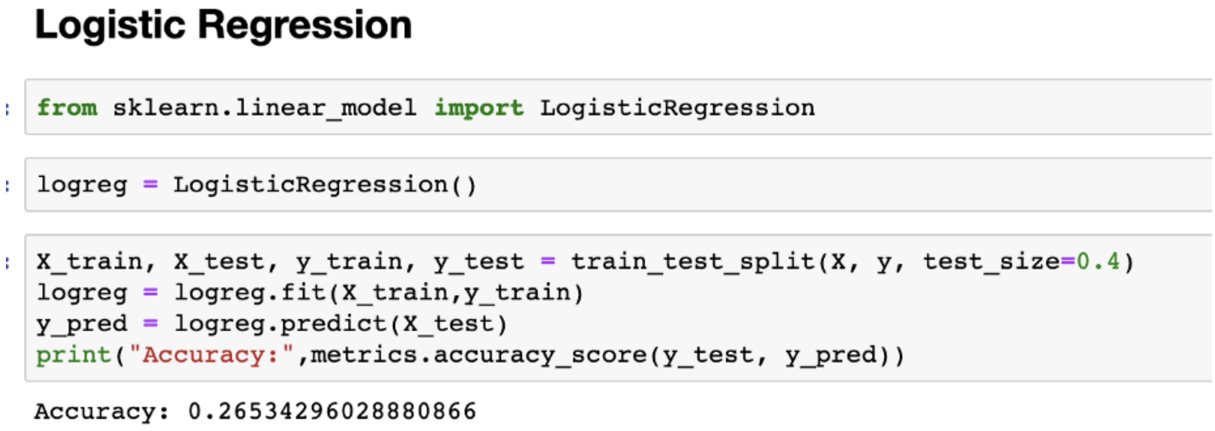
```
random_clf=RandomForestClassifier(n_estimators=1000)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4)
random_clf = random_clf.fit(X_train,y_train)
y_pred = random_clf.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.2563176895306859
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
random_clf = random_clf.fit(X_train,y_train)
y_pred = random_clf.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.23317307692307693
```



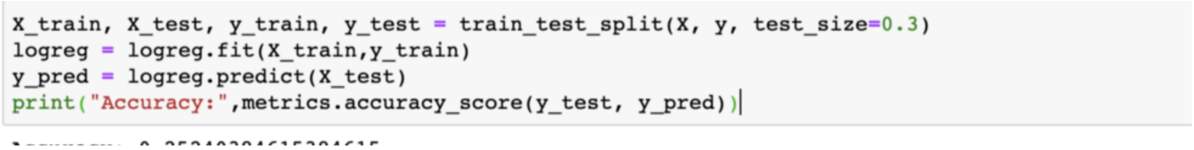
Logistic Regression

```
from sklearn.linear_model import LogisticRegression
```

```
logreg = LogisticRegression()
```

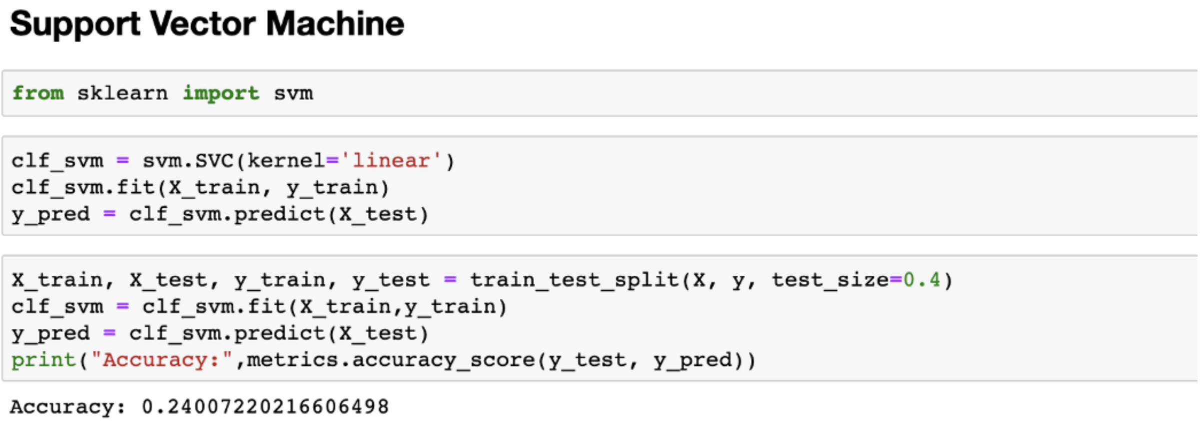
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4)
logreg = logreg.fit(X_train,y_train)
y_pred = logreg.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.26534296028880866
```



```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
logreg = logreg.fit(X_train,y_train)
y_pred = logreg.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.25240384615384615
```



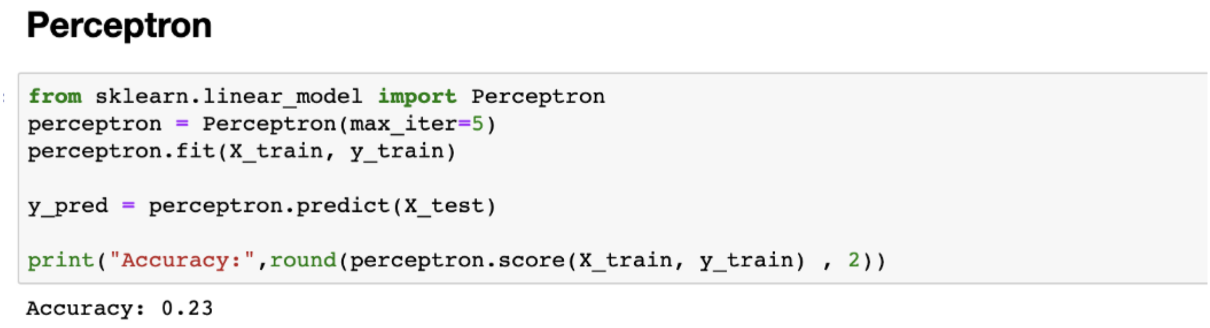
Support Vector Machine

```
from sklearn import svm
```

```
clf_svm = svm.SVC(kernel='linear')
clf_svm.fit(X_train, y_train)
y_pred = clf_svm.predict(X_test)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4)
clf_svm = clf_svm.fit(X_train,y_train)
y_pred = clf_svm.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.24007220216606498
```



Perceptron

```
from sklearn.linear_model import Perceptron
perceptron = Perceptron(max_iter=5)
perceptron.fit(X_train, y_train)

y_pred = perceptron.predict(X_test)

print("Accuracy:",round(perceptron.score(X_train, y_train) , 2))
```

```
Accuracy: 0.23
```

Analyzing ALT Levels

ALT(Alanine Aminotransferase) Test

The alanine aminotransferase (ALT) test is a blood test that checks for liver damage. Higher levels of ALT indicate Greater Liver Damage

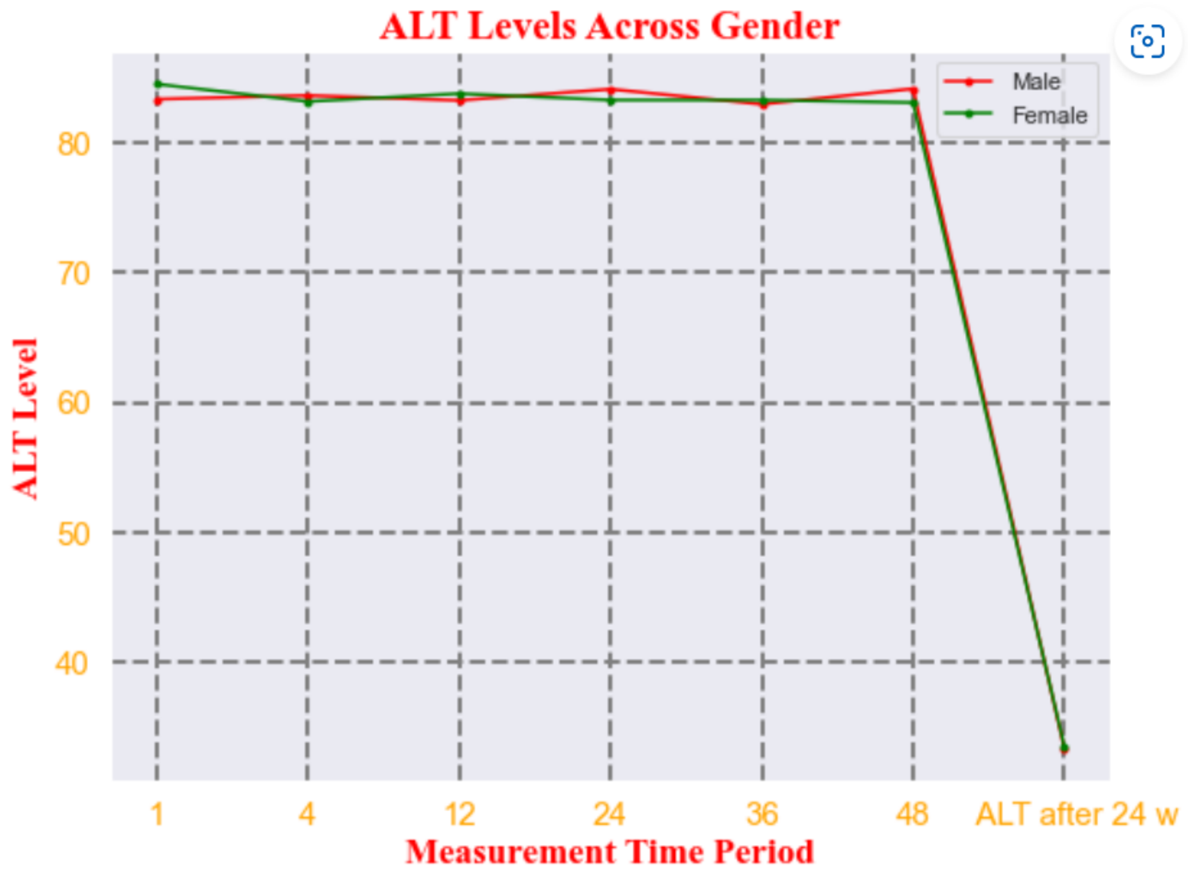
Across Gender

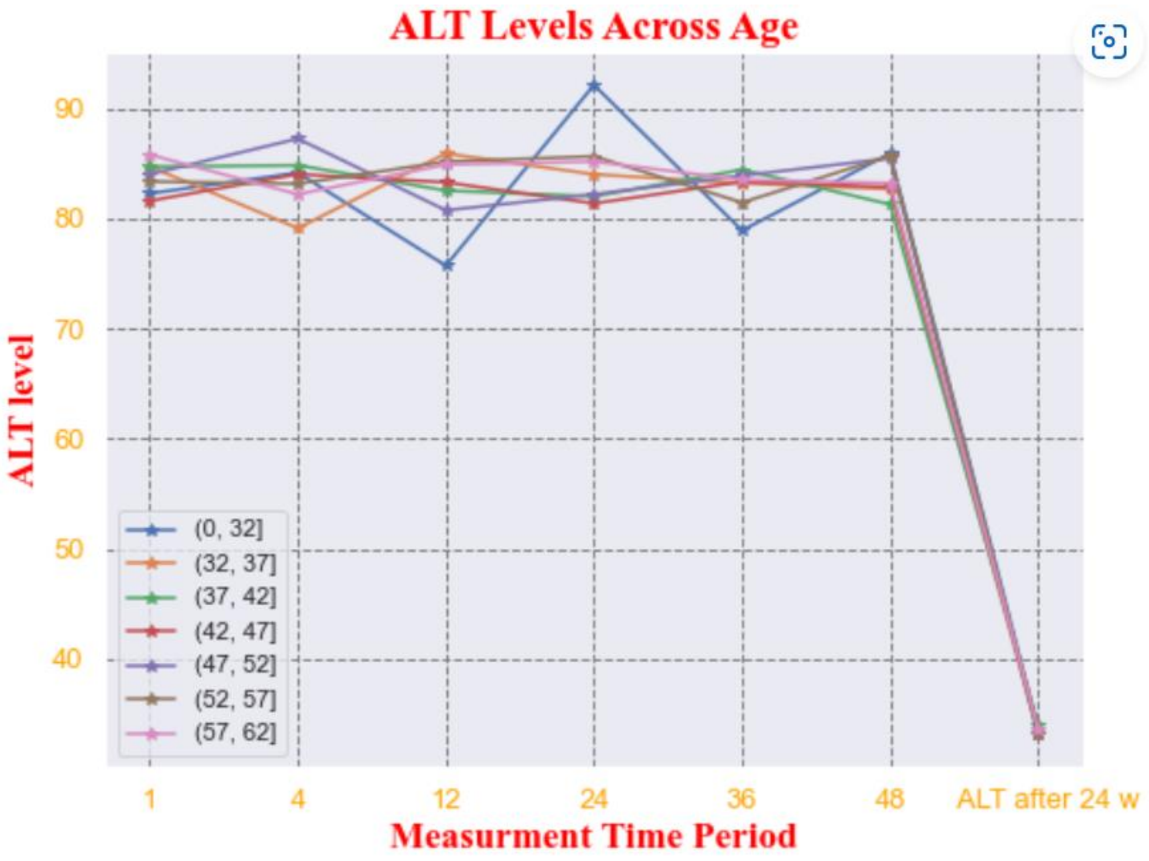
Although there is not much variance between Male and Female ALT (mean) levels, both of them do drop after 24 Weeks showing improvement in Liver Conditions for both

Across Age

Though there is variation in ALT (mean) levels amongst different age groups, all of them follow a general trend and drop together after 24 Weeks also showing improvement in Liver conditions

Plotting graph for ALT level across Gender and Age


ALT Levels Across Gender
ALT Level
80
70
60
50
40
1
4
12
24
36
48
ALT after 24 w
Measurement Time Period
Male
Female


ALT Levels Across Age
ALT level
90
80
70
60
50
40
(0, 32]
(32, 37]
(37, 42]
(42, 47]
(47, 52]
(52, 57]
(57, 62]
1
4
12
24
36
48
ALT after 24 w
Measurment Time Period

Team Member and Work Division:

1. **Hemchand Chandravanshi (Team Leader)**
    - (i) Discretization (joined)
    - (ii) Predictive Model
        - a. Random Forest Classifier
        - b. Decision Tree Classifier
        - c. Logistic Regression
    - (iii) Analysing ALT level
    - (iv) Project Report(joined)

2. **Vivek Singh**
    - (i) Discretization(joined)
    - (ii) Predictive Model
        - a. Gausian Naïve Bayes
        - b. Support Vector Machine
        - c. Perceptron
    - (iii) Project Report(joined)

SWOT Analysis

1. **Strength:**
   This Project has very good strength after Discretized data set. Earlier the dataset was continuous but after discretization this dataset can be used for any kind of predictive model.

2. **Weakness:**
   If we analyse the project than there is no weakness in project but some weakness exist in terms of data provided because if see the predictive models then non of the predictive model has good accuracy, we come to know that the dataset is bit inconsistent.

3. **Opportunities:**
   This project has lot of opportunities if some one wants to analyse the HCV dataset or want to discretize the their own data set then they can use this projects ideology for their own work. The discretization method that we have use is very comprehensive.

4. **Threats:**
   There is no threats available in this project.

Conclusion:

This project conclude that if anyone want to analyse the HCV data set and want to give proper result then the discretization perform by us will help then to analyse any kind of information regarding HCV like we have shown the example of ALT level analysing. But if we build predictive model with this continuous data set then the accuracy of that model will less. For predictive model we conclude that each algorithm gives a different accuracy. Since accuracy is low in all algorithms even after feature selection, we come to know that the dataset is a bit inconsistent. Each algorithm applied here is for classification. We cannot apply any regression algorithm on the classification dataset.