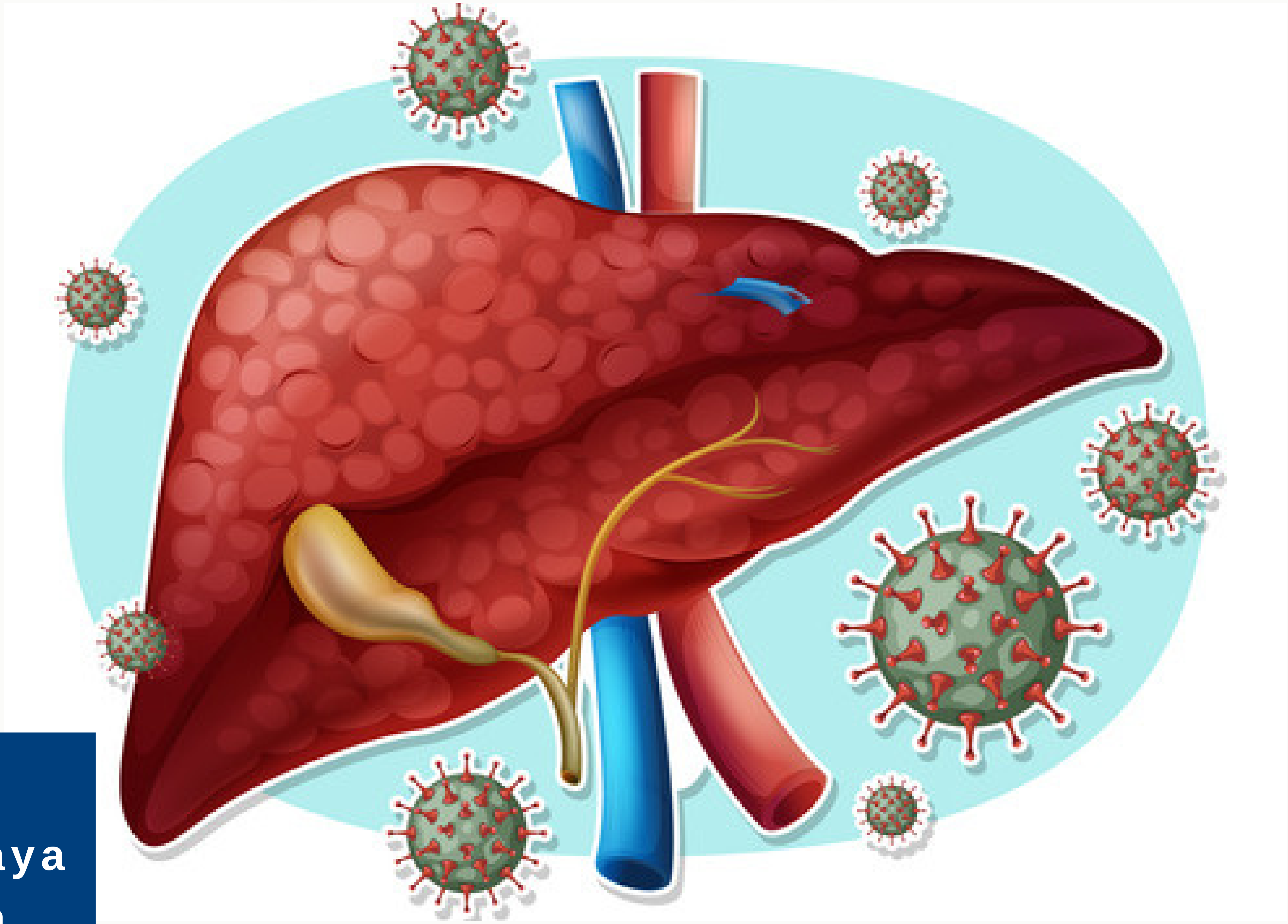
PREDICTION OF HCV

NOVEMBER 2023

7th SEMESTER

PROJECT

Department of Biotechnology



Presented by
- Deepak Kumar
- Devesh Singh

Guided by
- Dr. Chinmaya Mahapatra

Objective

To predict the Hepatitis-C by using machine learning models based on different attributes

TABLE OF CONTENT

01. Introduction

02. Methodology

03. Result

04. Conclusion

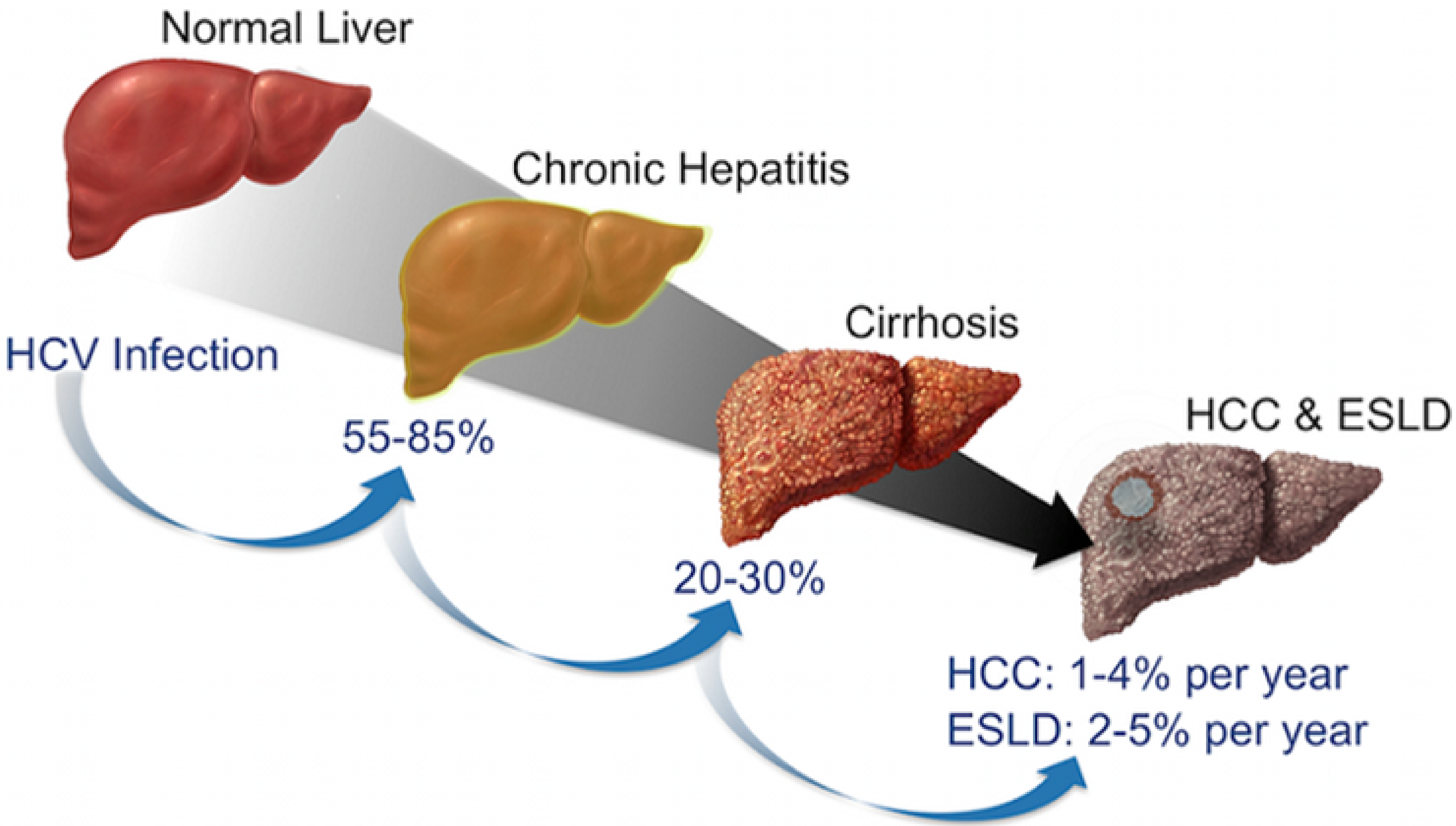
Introduction

Hepatitis C virus (HCV) is an RNA virus and one of the major blood-born human pathogen called as Hepatitis C. HCV infection is largely asymptomatic with little visible symptoms during infection stage. Without treatment, most of the acute infections progress to chronic ones followed by liver diseases such as cirrhosis and hepatocellular carcinoma.

Globally, an estimated 58 million people have chronic hepatitis C virus infection, with about 1.5 million new infections occurring per year. There are an estimated 3.2 million adolescents and children with chronic hepatitis C infection.

WHO estimated that in 2019, approximately 290 000 people died from hepatitis C, mostly from cirrhosis and hepatocellular carcinoma (primary liver cancer). Traditional methods for detecting HCV involve invasive blood tests and liver biopsy, which can be expensive and uncomfortable for patients.

In recent years, machine learning (ML) algorithms have emerged as a promising approach for the early and accurate detection of HCV. ML algorithms, such as KNN, SVM, and logistic regression have been extensively used in medical diagnosis, including HCV detection.


Normal Liver
Chronic Hepatitis
Cirrhosis
HCC & ESLD
HCV Infection
55-85%
20-30%
HCC: 1-4% per year
ESLD: 2-5% per year

Purpose

1) Accuracy and Sensitivity: Provide the exact situation of the patient, means the level of disease; whether it is at initial stage or at critical stage.

2) Analysing the Multiple past lab record of a patient, means understanding the trends. To predict the likelihood of disease progression and guide treatment decisions.

3) Integration with Electronic Health Records: Help to manage your past health record. This means your doctor gets instant help in making decisions, making your healthcare smoother and better.

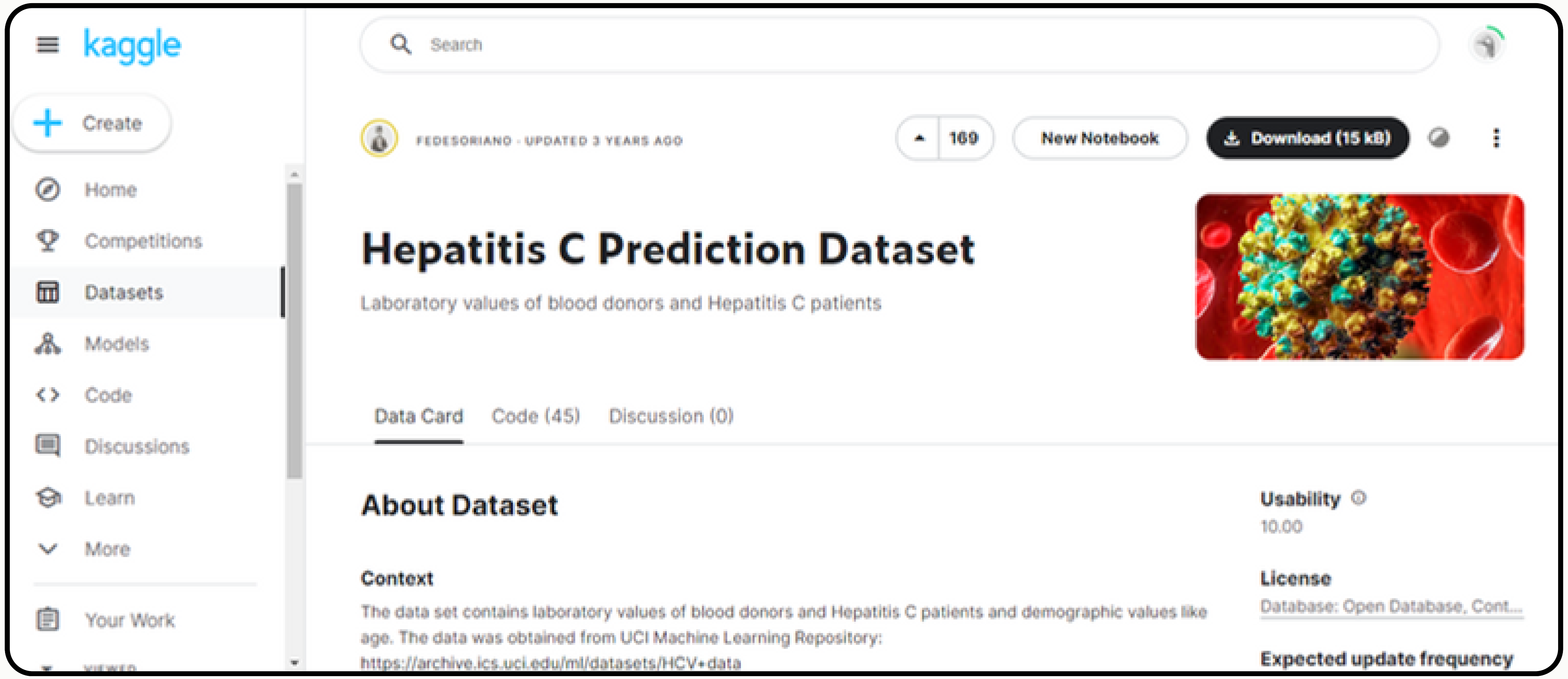
4) ML algorithms can analyze large datasets to identify patterns and associations that can be used to diagnose HCV accurately.

Methodology

STEPS:

1. DATA COLLECTION

The first step in any machine learning project is to obtain a dataset. For this project, the dataset for Hepatitis C detection can be obtained from publicly available repositories such as the UCI Machine Learning Repository, Kaggle, or any other reliable source.



Hepatitis C Prediction Dataset

Laboratory values of blood donors and Hepatitis C patients

About Dataset

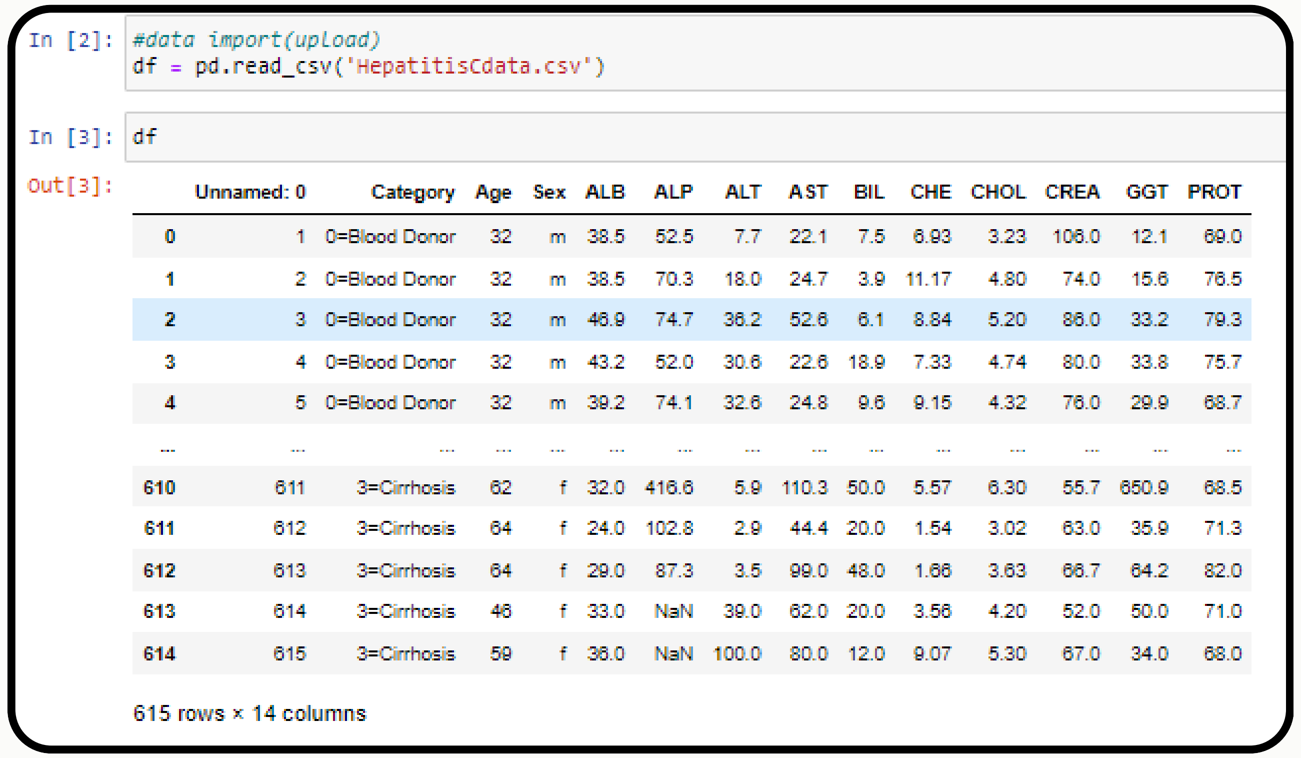
Context

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: https://archive.ics.uci.edu/ml/datasets/HCV+data

Usability 10.00

License Database: Open Database, Cont...

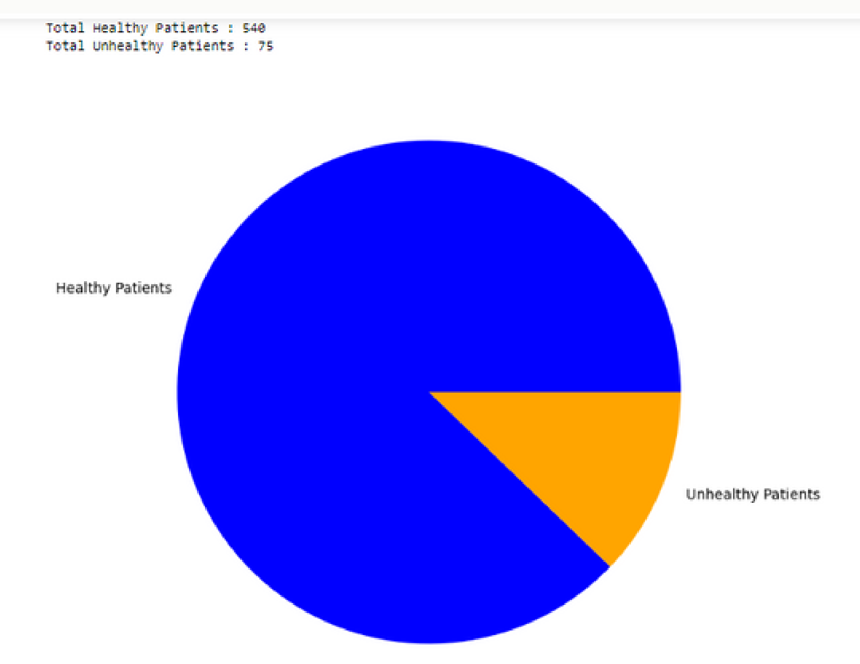
Expected update frequency



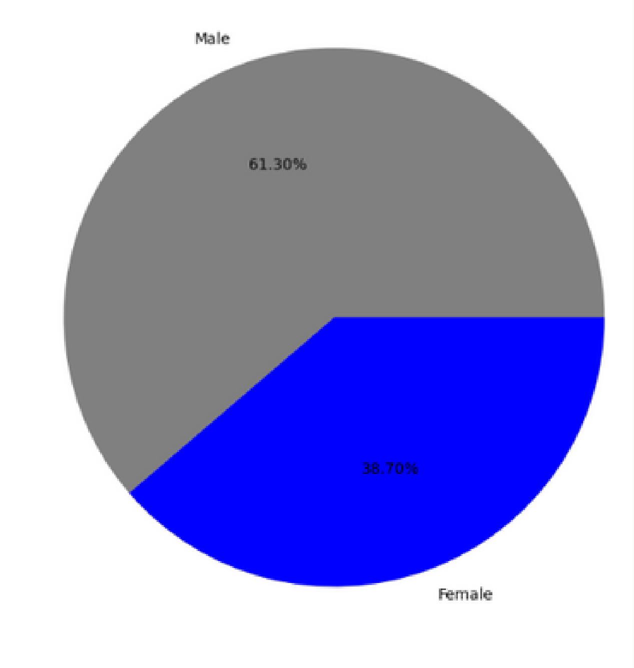
```
In [2]: #data import(upload)
        df = pd.read_csv('HepatitisCdata.csv')

In [3]: df
Out[3]:
```

	Unnamed: 0	Category	Age	Sex	ALB	ALP	ALT	AST	BIL	CHE	CHOL	CREA	GGT	PROT
0	1	0=Blood Donor	32	m	38.5	52.5	7.7	22.1	7.5	6.93	3.23	106.0	12.1	69.0
1	2	0=Blood Donor	32	m	38.5	70.3	18.0	24.7	3.9	11.17	4.80	74.0	15.6	76.5
2	3	0=Blood Donor	32	m	46.9	74.7	36.2	52.6	6.1	8.84	5.20	86.0	33.2	79.3
3	4	0=Blood Donor	32	m	43.2	52.0	30.6	22.6	18.9	7.33	4.74	80.0	33.8	75.7
4	5	0=Blood Donor	32	m	39.2	74.1	32.6	24.8	9.6	9.15	4.32	76.0	29.9	68.7
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
610	611	3=Cirrhosis	62	f	32.0	416.6	5.9	110.3	50.0	5.57	6.30	55.7	650.9	68.5
611	612	3=Cirrhosis	64	f	24.0	102.8	2.9	44.4	20.0	1.54	3.02	63.0	35.9	71.3
612	613	3=Cirrhosis	64	f	29.0	87.3	3.5	99.0	48.0	1.66	3.63	66.7	64.2	82.0
613	614	3=Cirrhosis	46	f	33.0	NaN	39.0	62.0	20.0	3.56	4.20	52.0	50.0	71.0
614	615	3=Cirrhosis	59	f	36.0	NaN	100.0	80.0	12.0	9.07	5.30	67.0	34.0	68.0

615 rows × 14 columns



```
Total Healthy Patients : 540
Total Unhealthy Patients : 75
```

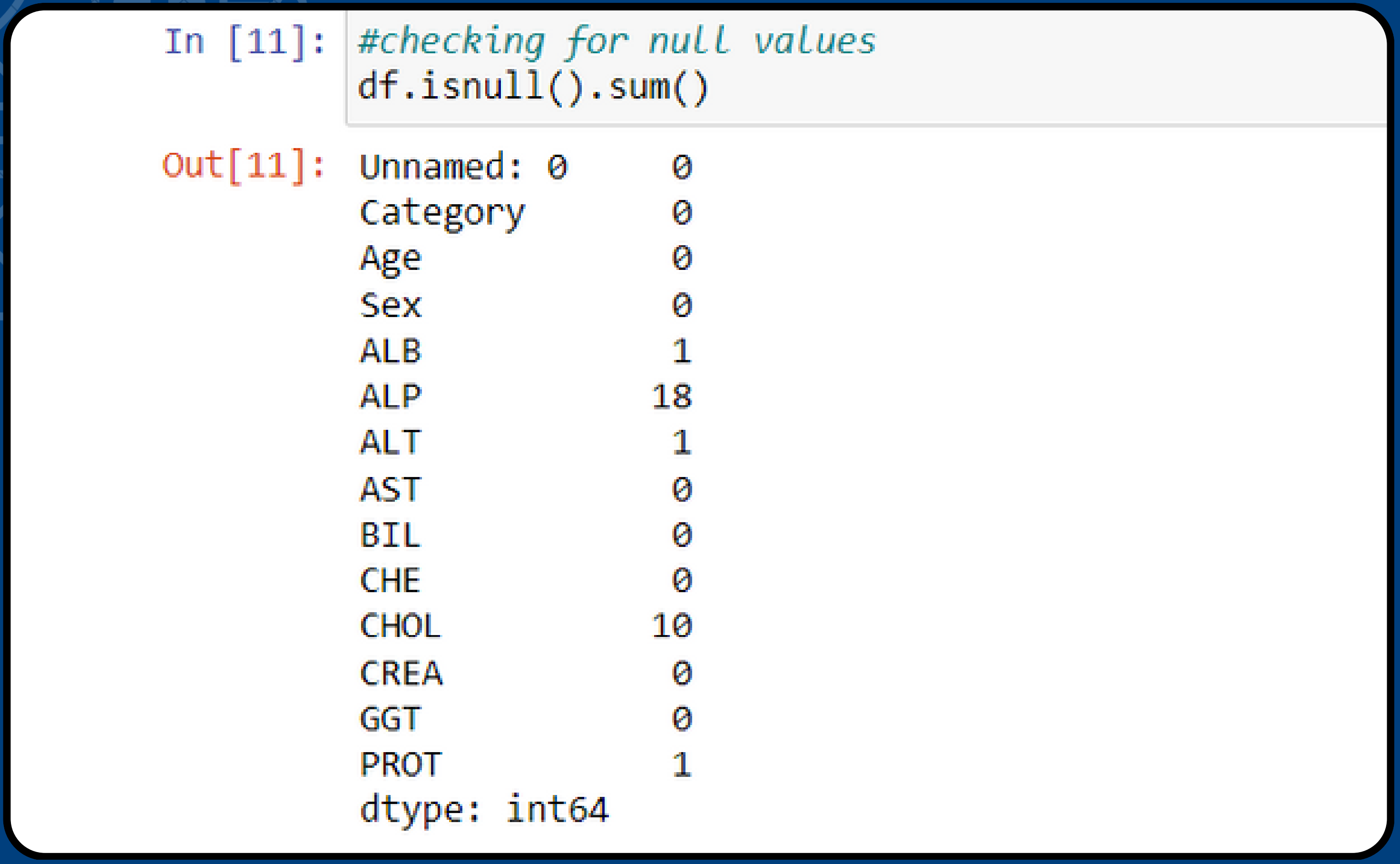


Here's a breakdown of how each of the listed attributes can help in detecting or predicting HCV:

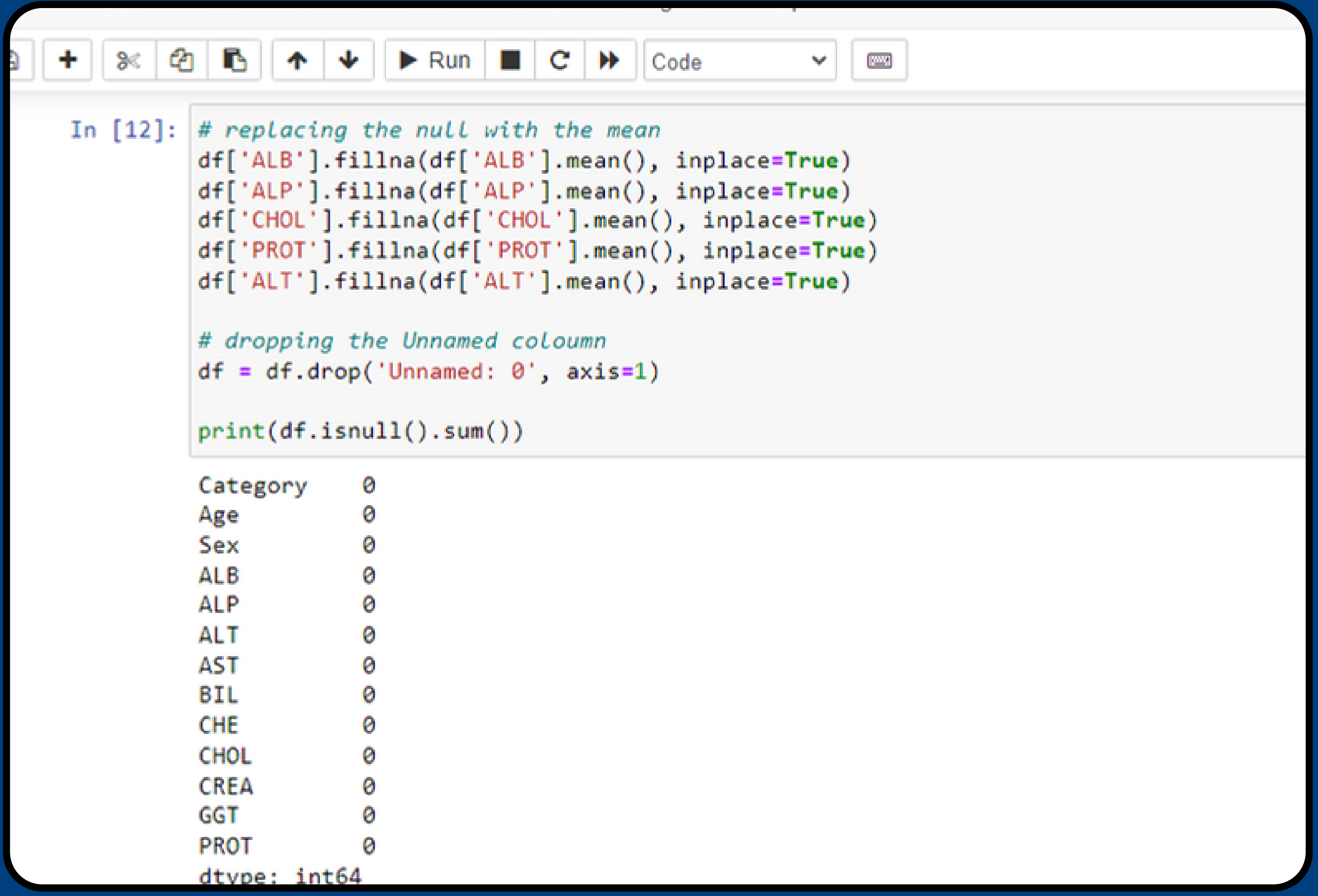
- **ALB (Albumin):** Albumin is a protein produced by the liver, and low levels of albumin can indicate liver damage associated with HCV infection.
- **ALP (Alkaline phosphatase)**: ALP is an enzyme found in the liver, bile ducts, and bones. Elevated levels of ALP can indicate liver damage or inflammation caused by HCV.
- **ALT (Alanine aminotransferase**): ALT is an enzyme found primarily in the liver. A significant increase in ALT levels is a highly sensitive indicator of liver damage, which is a common manifestation of HCV infection.
- **AST (Aspartate aminotransferase)**: AST is an enzyme found in the liver, heart, muscles, and other tissues. Elevated AST levels can indicate liver damage or inflammation, but they are also associated with other conditions, making them less specific for HCV detection.

- **BIL (Bilirubin):** Bilirubin is a yellow pigment produced by the breakdown of red blood cells. Elevated bilirubin levels, especially direct bilirubin, can indicate liver dysfunction, which is often associated with HCV infection.
- **CHE (Cholesterol):** Cholesterol is a fat-like substance found in the blood. While not directly indicative of HCV infection, elevated cholesterol levels can be a risk factor for liver disease, including chronic HCV-related liver disease.
- **CHOL (Total Cholesterol)**: Total cholesterol refers to the total amount of cholesterol in the blood, including LDL (bad) cholesterol, HDL (good) cholesterol, and other types of cholesterol. Elevated total cholesterol levels can increase the risk of liver disease, including HCV-related liver damage.
- **CREA (Creatinine)**: Creatinine is a waste product produced by the muscles and filtered out by the kidneys. Elevated creatinine levels can indicate kidney dysfunction, which can be a complication of advanced HCV infection.
- **GGT (Gamma-glutamyl transferase):** GGT is an enzyme found primarily in the liver and bile ducts. Elevated GGT levels are a sensitive indicator of liver damage and inflammation, making them a useful marker for HCV infection.
- **PROT (Protein)**: Total protein levels in the blood can provide an overall assessment of liver function. Low protein levels can indicate liver damage associated with HCV infection.

It's important to note that these laboratory tests are not definitive for HCV diagnosis and should be interpreted in conjunction with other clinical factors and diagnostic tests, such as antibody and nucleic acid testing.



```
In [11]: #checking for null values
         df.isnull().sum()

Out[11]: Unnamed: 0     0
         Category       0
         Age            0
         Sex            0
         ALB            1
         ALP           18
         ALT            1
         AST            0
         BIL            0
         CHE            0
         CHOL          10
         CREA           0
         GGT            0
         PROT           1
         dtype: int64
```



```
In [12]: # replacing the null with the mean
         df['ALB'].fillna(df['ALB'].mean(), inplace=True)
         df['ALP'].fillna(df['ALP'].mean(), inplace=True)
         df['CHOL'].fillna(df['CHOL'].mean(), inplace=True)
         df['PROT'].fillna(df['PROT'].mean(), inplace=True)
         df['ALT'].fillna(df['ALT'].mean(), inplace=True)

         # dropping the Unnamed coloumn
         df = df.drop('Unnamed: 0', axis=1)

         print(df.isnull().sum())

         Category    0
         Age         0
         Sex         0
         ALB         0
         ALP         0
         ALT         0
         AST         0
         BIL         0
         CHE         0
         CHOL        0
         CREA        0
         GGT         0
         PROT        0
         dtype: int64
```

2. DATA PREPARATION

Here we convert our raw data into meaningful data so that we can predict disease accurately.

- Here we can get noise data/unwanted data like NAN OR DATA WHICH IS OUTLIER
- USING Python library and some model like **LabelOrderEncoder**, filling null value through mean.

Once the dataset is obtained, it needs to be preprocessed to ensure it is suitable for use in machine learning algorithms. This involves data cleaning to remove any missing or erroneous values in the dataset. Feature selection is then performed to identify the most relevant features that can aid in accurate classification.

3. MODEL SELECTION

In this step, the most appropriate machine learning algorithms for Hepatitis C detection are selected. Common algorithms used in classification tasks include Logistic Regression, K-Nearest Neighbours (KNN), and Support Vector Machines (SVM).

- **K-Nearest Neighbors (KNN)**:

Principle: Predictions based on majority class among its k-nearest neighbors in feature space.
Key Points: Simple and intuitive. Performance can vary based on the choice of k and distance metric.

- **Support Vector Machines (SVM):**

Principle: Find the optimal hyperplane that best separates classes while maximizing the margin.
Key Points: Effective in high-dimensional spaces, versatile due to different kernel functions (linear, polynomial, RBF).

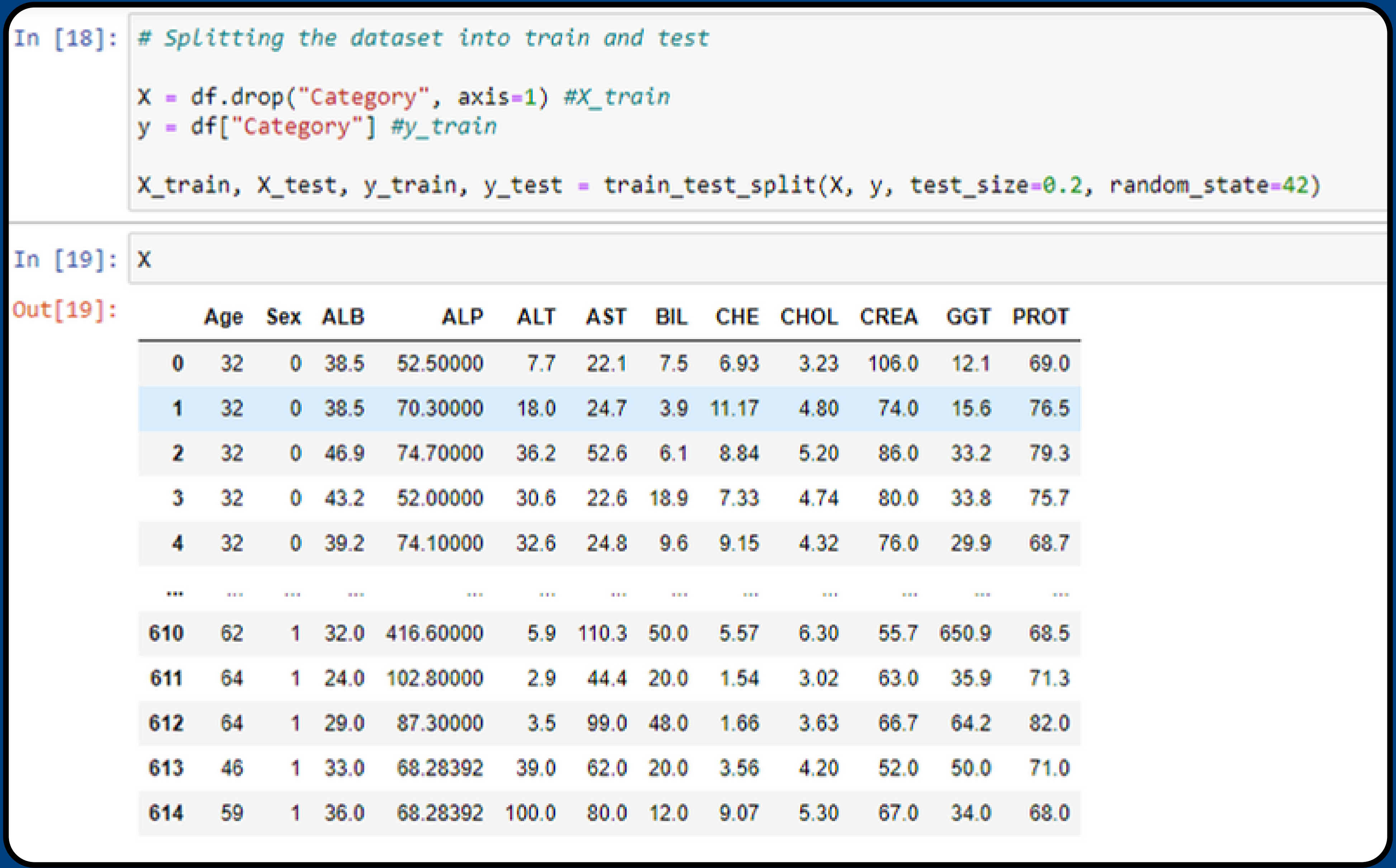
- **Random Forest:**

Principle: Constructs multiple decision trees and merges their predictions to improve accuracy and reduce overfitting.
Key Points: Robust against overfitting, handles large datasets with high dimensionality, provides feature importance.

- **Logistic Regression:**

Principle: Models the probability of a binary outcome using a logistic function.
Key Points: Simple yet powerful, interpretable coefficients representing feature importance, assumes linear relationship between features and log-odds of the outcome.

Each algorithm has its strengths and weaknesses, and the choice often depends on the nature of the problem, the size and quality of the dataset, and the specific requirements of the application.



```
In [18]: # Splitting the dataset into train and test

         X = df.drop("Category", axis=1) #X_train
         y = df["Category"] #y_train

         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
In [19]: X

Out[19]:
       Age  Sex   ALB        ALP    ALT    AST   BIL    CHE  CHOL   CREA    GGT  PROT
  0     32    0  38.5   52.50000    7.7   22.1   7.5   6.93  3.23  106.0   12.1  69.0
  1     32    0  38.5   70.30000   18.0   24.7   3.9  11.17  4.80   74.0   15.6  76.5
  2     32    0  46.9   74.70000   36.2   52.6   6.1   8.84  5.20   86.0   33.2  79.3
  3     32    0  43.2   52.00000   30.6   22.6  18.9   7.33  4.74   80.0   33.8  75.7
  4     32    0  39.2   74.10000   32.6   24.8   9.6   9.15  4.32   76.0   29.9  68.7
...    ...  ...   ...        ...    ...    ...   ...    ...   ...    ...    ...   ...
610     62    1  32.0  416.60000    5.9  110.3  50.0   5.57  6.30   55.7  650.9  68.5
611     64    1  24.0  102.80000    2.9   44.4  20.0   1.54  3.02   63.0   35.9  71.3
612     64    1  29.0   87.30000    3.5   99.0  48.0   1.66  3.63   66.7   64.2  82.0
613     46    1  33.0   68.28392   39.0   62.0  20.0   3.56  4.20   52.0   50.0  71.0
614     59    1  36.0   68.28392  100.0   80.0  12.0   9.07  5.30   67.0   34.0  68.0
```

4. TRAINING OF MODEL

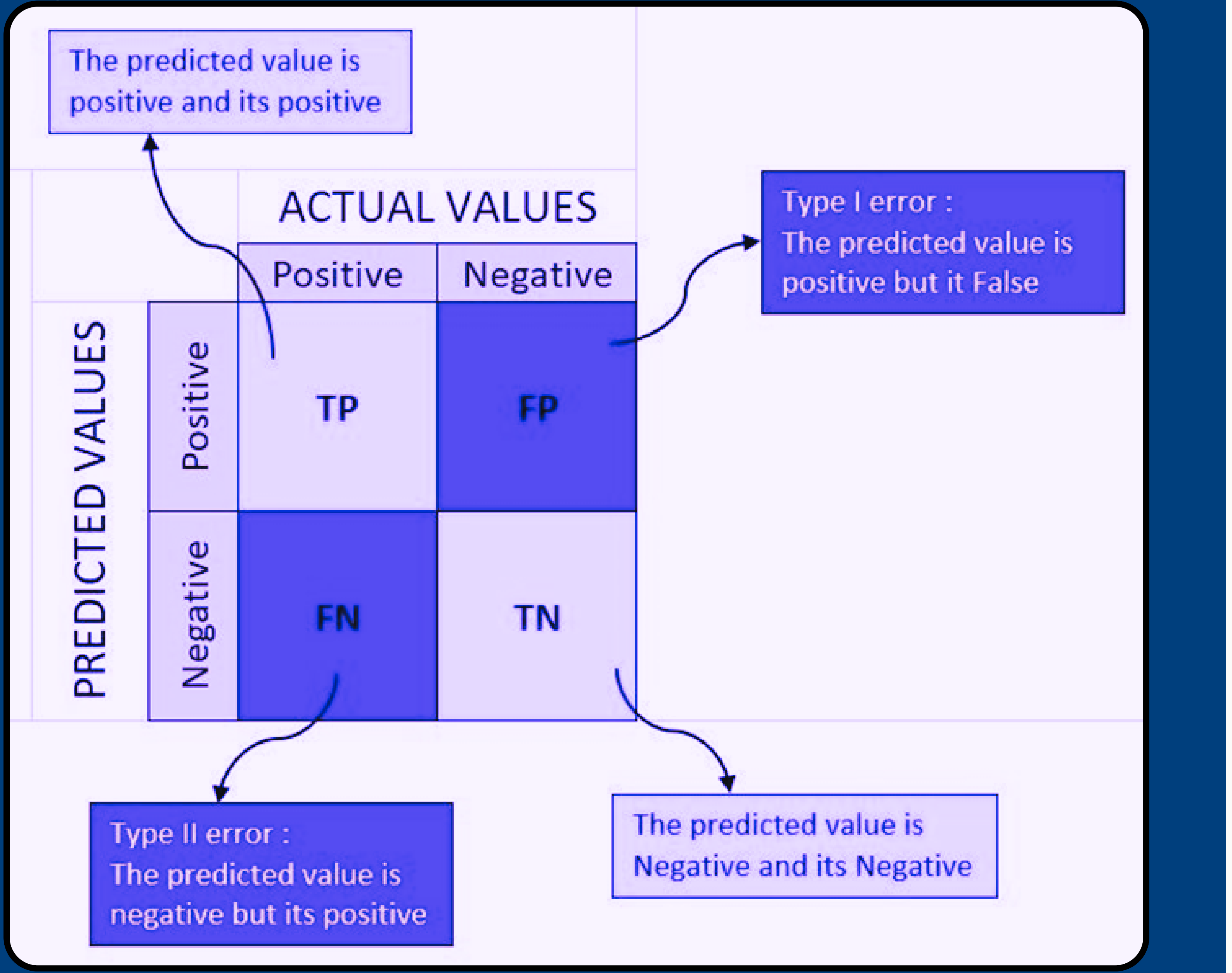
The selected machine learning algorithms are trained on the preprocessed dataset. In this step dataset is divided into two parts one training dataset and one testing dataset to evaluate the performance of the model.

5. MODEL EVALUATION

After training the models, their performance must be analysed to determine which model is the most accurate. In classification tasks, common evaluation criteria include accuracy, precision, recall, and F1-score. Additionally, Receiver Operating Characteristic (ROC) curves can be utilised to assess model performance. Performance Requirements When it comes to performance of a model to Evaluate to know how the model is working.

Factors to evaluate the models are :
1. Confusion Matrix
2. Classification Report (Precision, Recall, F-1 Score & Accuracy Confusion Matrix)

Confusion matrix is a table that shows the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). In this case, we have a binary classification problem, where class 0 is the negative class and class 1 is the positive class.



The predicted value is positive and its positive

		ACTUAL VALUES	
		Positive	Negative
PREDICTED VALUES	Positive	TP	FP
	Negative	FN	TN

Type I error : The predicted value is positive but it False

Type II error : The predicted value is negative but its positive

The predicted value is Negative and its Negative

Accuracy: It is the performance measure ratio of correctly predicted observation of the total observations.
Accuracy = (TP + TN)/(TP + FP+ FN+ TN)

Precision: It is the ratio of correctly predicted positive observations of the total predicted positive observations .
Precision = TP /TP +FP

Recall: It can be also called sensitivity or true positive rate (TPR). It is the ratio of correctly predicted positive observations to all observations in an actual class.
Recall = TP/ TP +FN

where TP is True Positive, TN is True Negative, FP is False Positive, and FN is False Negative.

RESULT & CONCLUSION

In this project we have used some machine learning algorithm to detect the hepatitis C virus this virus can infect the person by getting in the contact with the infected persons blood person infected with this virus can feel a high fever nausea and can have yellow eyes some new medicans can improve the condition of the infected person or some medicans can degrade the persons condition. So it is far more important to detect this virus

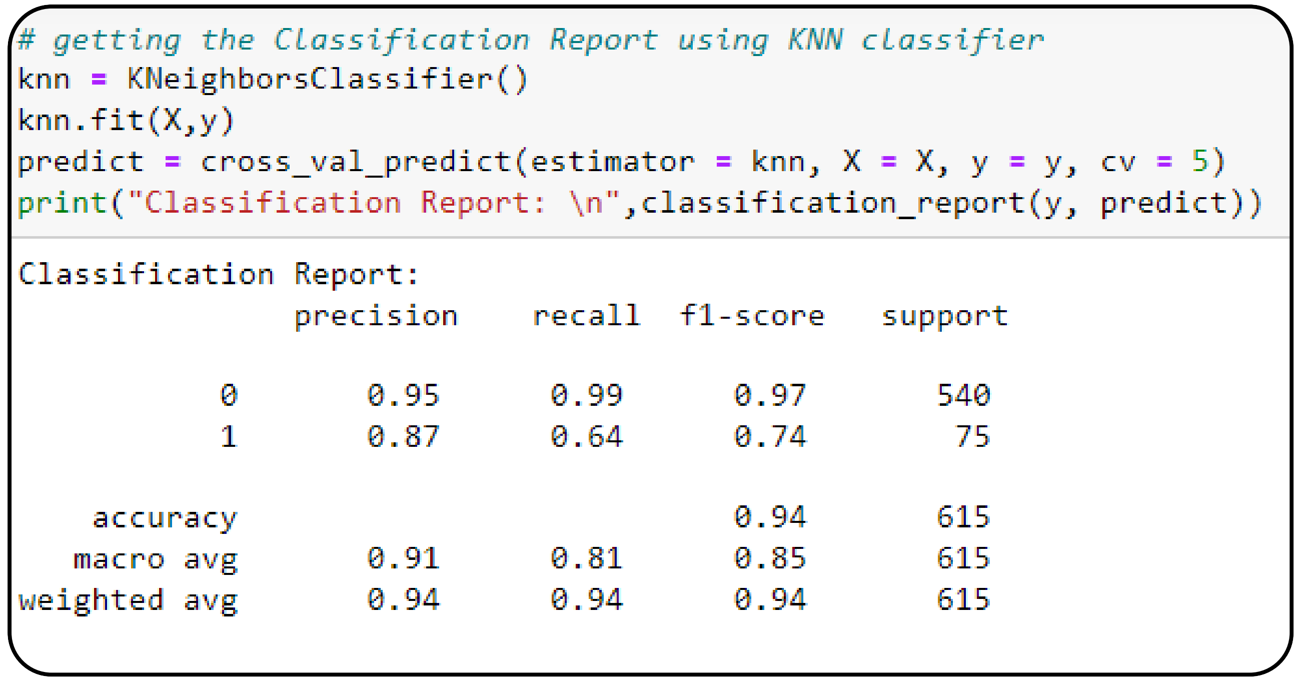
Here we have checked to accuracy of our model by using 3 different algorithms (SVM, KNN, Logistic Regression, Random Forest).

Accuracy:

1) RF (Random Forest) : 96%
2) SVM (Support Vector Machine) : 95 %
3) KNN (K-Neighbour Classifier) : 94 %
4) LR (Logistic Regression) : 94 %

REPORTS

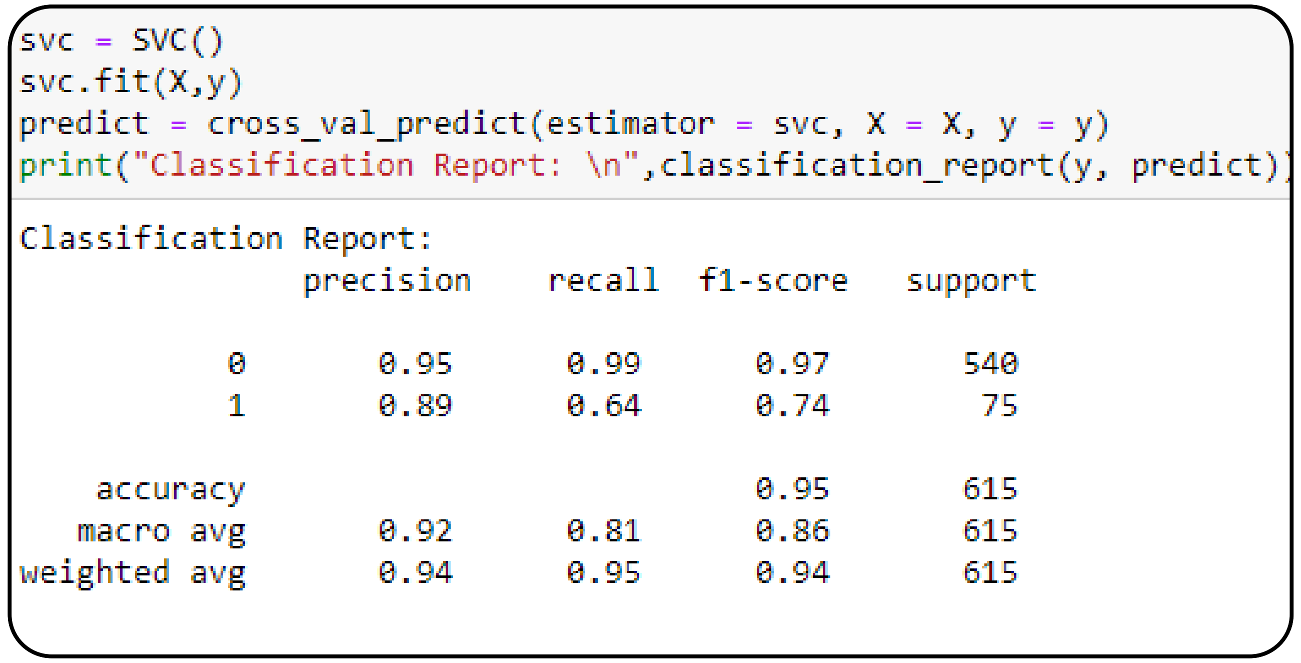
KNN



```
# getting the Classification Report using KNN classifier
knn = KNeighborsClassifier()
knn.fit(X,y)
predict = cross_val_predict(estimator = knn, X = X, y = y, cv = 5)
print("Classification Report: \n",classification_report(y, predict))
```

```
Classification Report:
               precision    recall  f1-score   support

           0       0.95      0.99      0.97       540
           1       0.87      0.64      0.74        75

    accuracy                           0.94       615
   macro avg       0.91      0.81      0.85       615
weighted avg       0.94      0.94      0.94       615
```

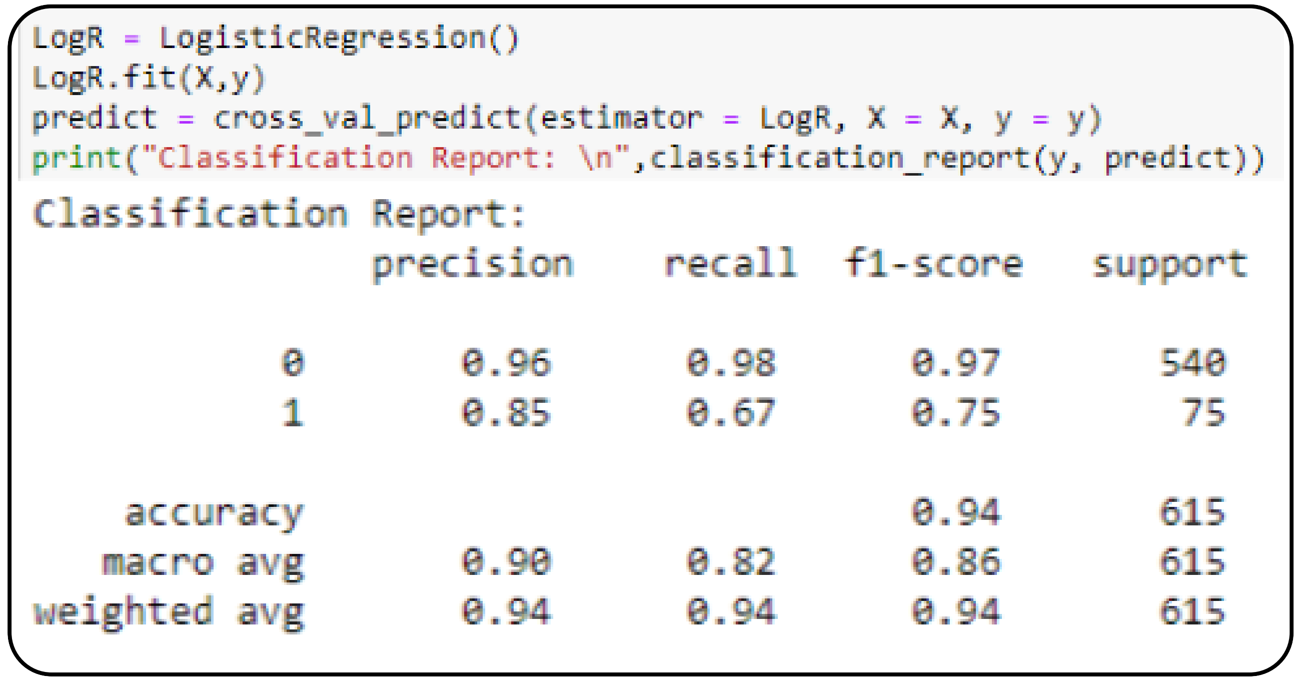
SVC



```
svc = SVC()
svc.fit(X,y)
predict = cross_val_predict(estimator = svc, X = X, y = y)
print("Classification Report: \n",classification_report(y, predict))
```

```
Classification Report:
               precision    recall  f1-score   support

           0       0.95      0.99      0.97       540
           1       0.89      0.64      0.74        75

    accuracy                           0.95       615
   macro avg       0.92      0.81      0.86       615
weighted avg       0.94      0.95      0.94       615
```

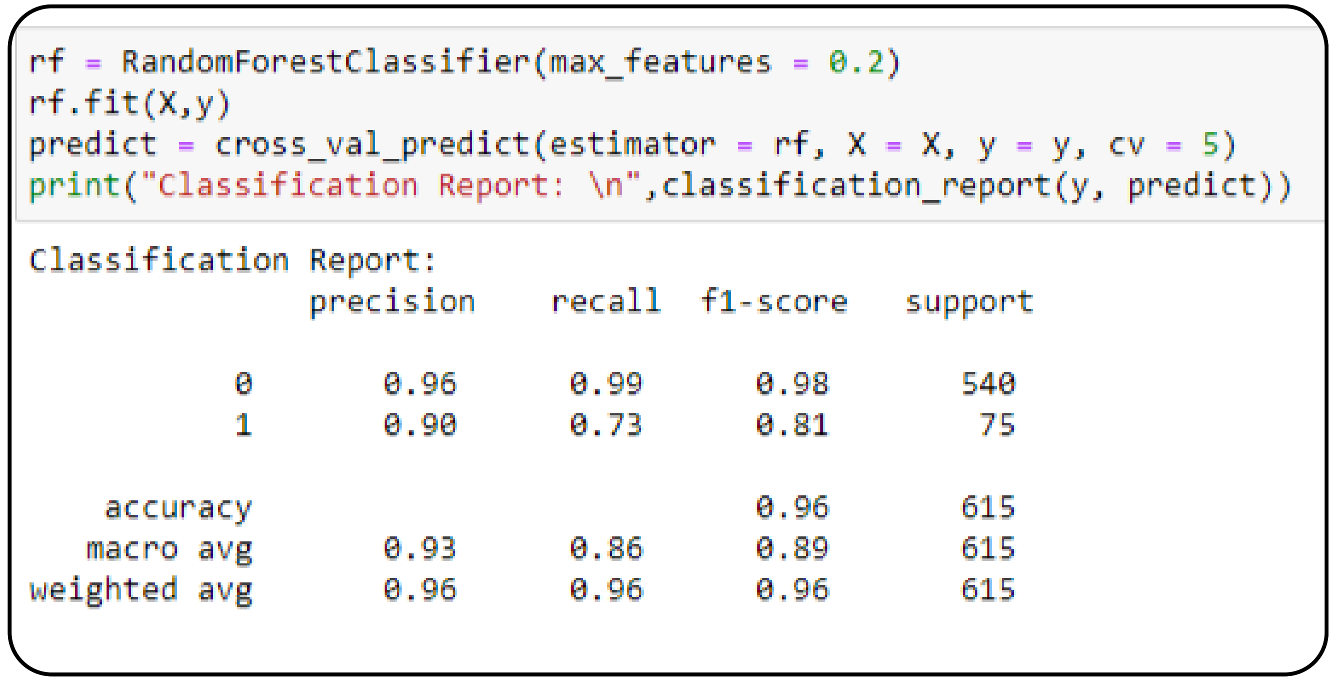
LR



```
LogR = LogisticRegression()
LogR.fit(X,y)
predict = cross_val_predict(estimator = LogR, X = X, y = y)
print("Classification Report: \n",classification_report(y, predict))
```

```
Classification Report:
               precision    recall  f1-score   support

           0       0.96      0.98      0.97       540
           1       0.85      0.67      0.75        75

    accuracy                           0.94       615
   macro avg       0.90      0.82      0.86       615
weighted avg       0.94      0.94      0.94       615
```

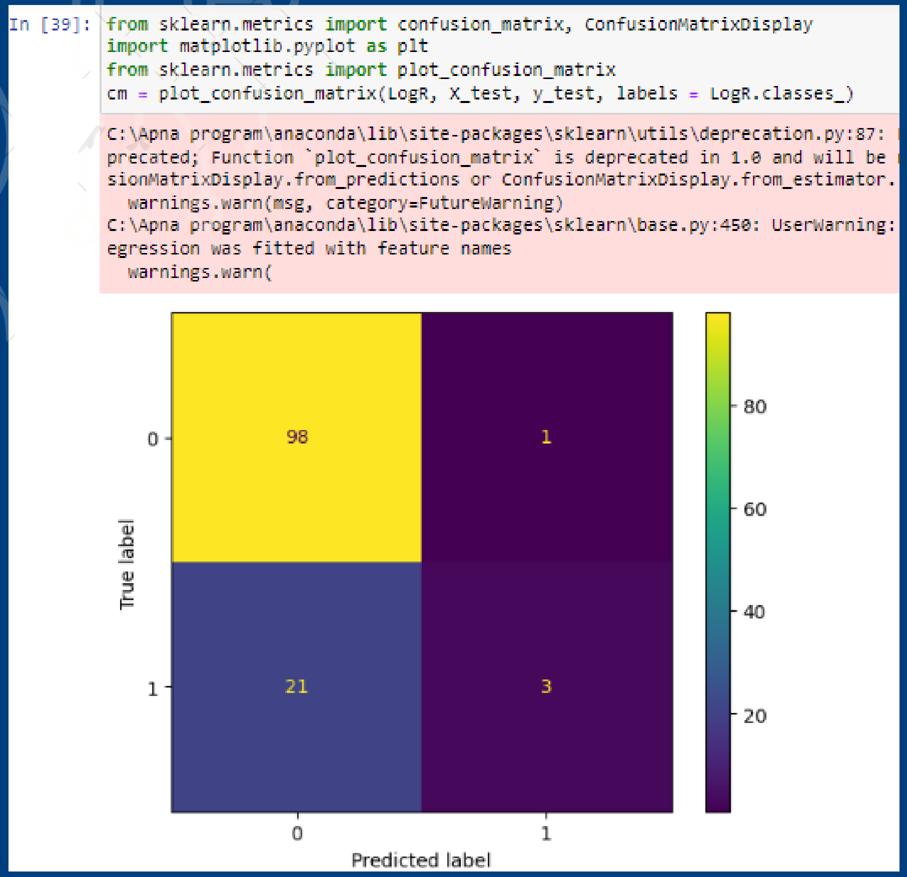
RF



```
rf = RandomForestClassifier(max_features = 0.2)
rf.fit(X,y)
predict = cross_val_predict(estimator = rf, X = X, y = y, cv = 5)
print("Classification Report: \n",classification_report(y, predict))
```

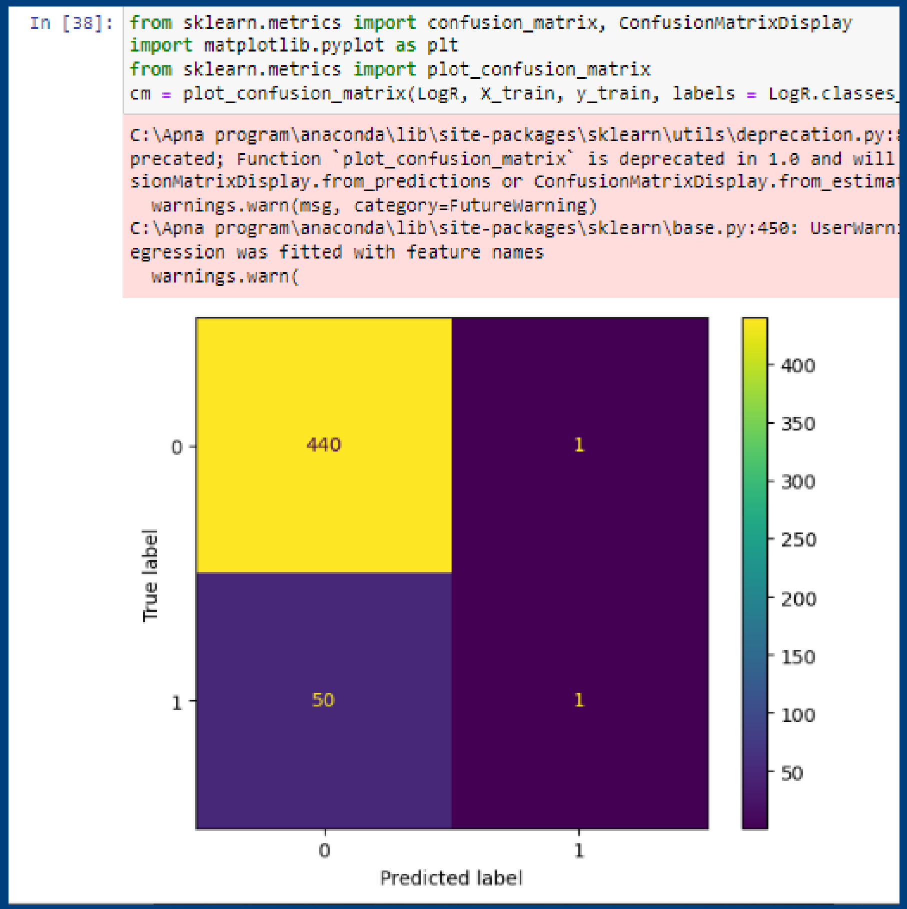
```
Classification Report:
               precision    recall  f1-score   support

           0       0.96      0.99      0.98       540
           1       0.90      0.73      0.81        75

    accuracy                           0.96       615
   macro avg       0.93      0.86      0.89       615
weighted avg       0.96      0.96      0.96       615
```

MODEL COMPARISON



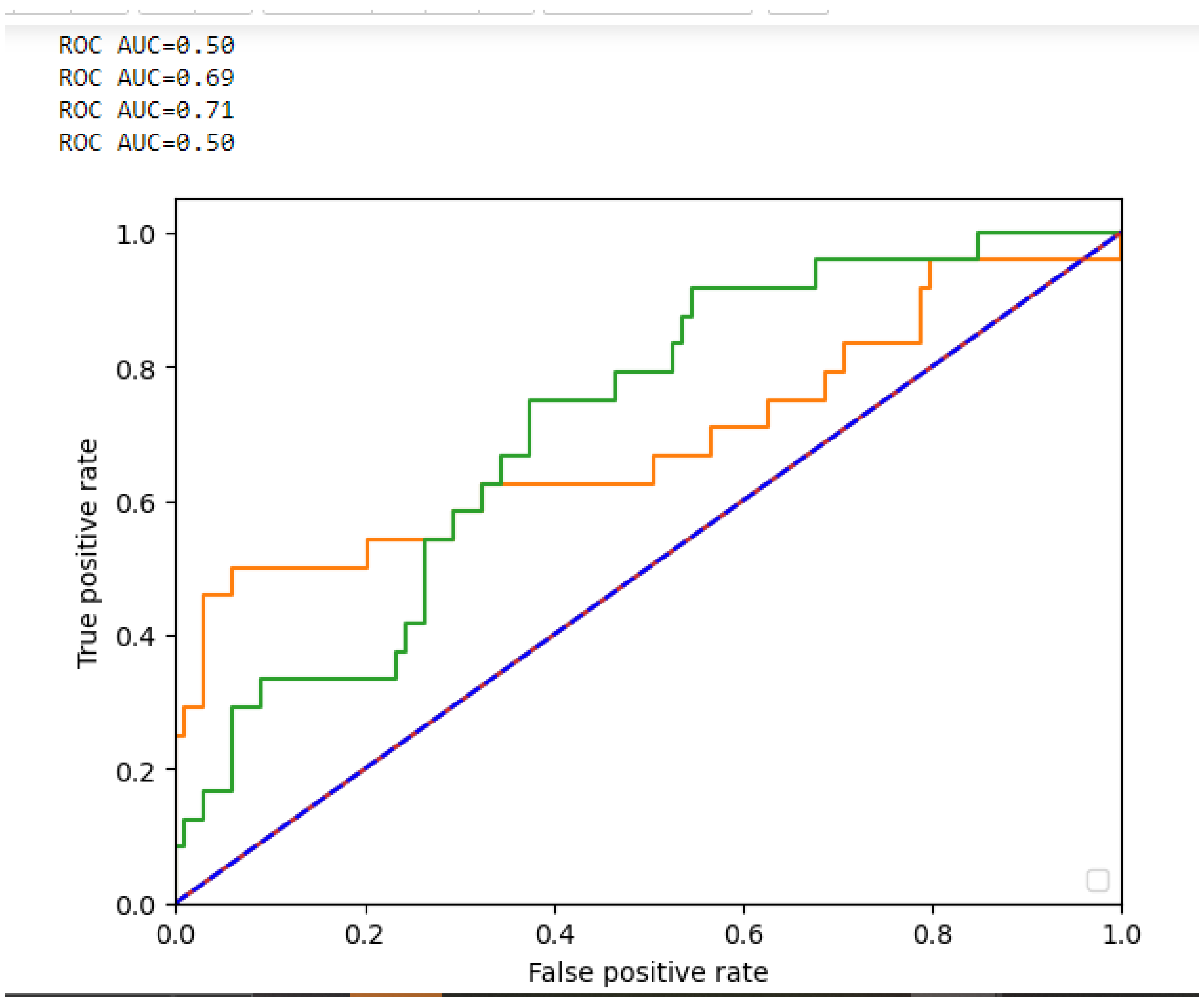
```
In [39]: from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
         import matplotlib.pyplot as plt
         from sklearn.metrics import plot_confusion_matrix
         cm = plot_confusion_matrix(LogR, X_test, y_test, labels = LogR.classes_)
```

```
C:\Apna program\anaconda\lib\site-packages\sklearn\utils\deprecation.py:87:
precated; Function `plot_confusion_matrix` is deprecated in 1.0 and will be
sionMatrixDisplay.from_predictions or ConfusionMatrixDisplay.from_estimator.
  warnings.warn(msg, category=FutureWarning)
C:\Apna program\anaconda\lib\site-packages\sklearn\base.py:450: UserWarning:
egression was fitted with feature names
  warnings.warn(
```



```
In [38]: from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
         import matplotlib.pyplot as plt
         from sklearn.metrics import plot_confusion_matrix
         cm = plot_confusion_matrix(LogR, X_train, y_train, labels = LogR.classes_
```

```
C:\Apna program\anaconda\lib\site-packages\sklearn\utils\deprecation.py:
precated; Function `plot_confusion_matrix` is deprecated in 1.0 and will
sionMatrixDisplay.from_predictions or ConfusionMatrixDisplay.from_estima
  warnings.warn(msg, category=FutureWarning)
C:\Apna program\anaconda\lib\site-packages\sklearn\base.py:450: UserWarn
egression was fitted with feature names
  warnings.warn(
```



```
ROC AUC=0.50
ROC AUC=0.69
ROC AUC=0.71
ROC AUC=0.50
```

REFERENCES

1. UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/Hepatitis+C+Virus+%28HCV%29+for+Iran

2. Kaggle: https://www.kaggle.com/

3. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

4. Alizadeh, S., Azmi, P., & Mahdi, E. (2017). Hepatitis C Diagnosis using Machine Learning Techniques. Journal of Medical Systems, 41(9), 1-10.

5. Gupta, R., & Sandhu, P. S. (2019). Machine Learning Approaches for Diagnosis of Hepatitis C Virus: A Comprehensive Review. Journal of Medical Systems, 43(1), 1-12.

6. Terrault, N.A. (2018). Hepatitis C elimination: challenges and the path forward. Nature Reviews Gastroenterology & Hepatology, 15(4), 221-222.

7. Arora, P., Kumar, A., & Rana, D. (2018). Hepatitis C virus detection using machine learning: A review. Journal of Medical Systems, 42(5), 87.

8. Khan, M., Hayat, M., & Badshah, I. (2021). An optimized machine learning model for hepatitis C virus detection. Journal of Medical Systems, 45(2), 1-12.