Project Executable Files

This This phase documents the **practical configurations, datasets, machine learning models, and system outputs** used and generated during the execution of the project: **“Predicting Liver Cirrhosis Using Advanced Machine Learning.”** It ensures that all key project elements—**clinical entities, datasets, predictive models, workflows, and results—are traceable, reproducible, and reusable** for future improvements, clinical assessments, or audits.This is where the **working components and validated modules** of the system are consolidated for clarity, clinical validation, replication, and ongoing system enhancement.

🔹 1. Project Files

Project Executable Files

The following project files were executed in the Python Jupyter Notebook

---

📁 Milestone 1: **Define Problem / Problem Understanding**

📁 **Milestone 2: Data collection & Preparation**

📁 **Milestone 3: Exploratory Data Analysis**

📁 **Milestone 4: Model Building**

📁 **Milestone 5: performance testing & Hyperparameter Tuning**

List of Milestone Tasks with Supporting Screenshots and Descriptions

---

📁 Milestone 1: Define Problem / Problem Understanding

- Specify the Diabetes Results.
- some Diabetes Results for an Liver Cirrhosis predictor using machine learning.
- A literature survey for a liver cirrhosis Prediction project

---

OUTPUT SCREENSHOT

📁 Milestone 2: Data collection & Preparation

- Collecting the Dataset
- This data is collected from Kaggle.com website.
- Data Preparation
  - Handling missing values
  - Handling categorical data
  - Handling Outliers

OUTPUT SCREENSHOT

```
]: # LOADING THE DATASET

df = pd.read_excel('E:\\HealthCareData.xlsx')
df.head()
```

]:

	S.NO	Age	Gender	Place(location where the patient lives)	Duration of alcohol consumption(years)	Quantity of alcohol consumption (quarters/day)	Type of alcohol consumed	Hepatitis B infection	Hepatitis C infection	Diabetes Result	...	Indirect (mg/dl)	Total Protein (g/dl)	Albumin (g/dl)	Globulin (g/dl)	A/G Ratio	A
0	1	55	male	rural	12	2	branded liquor	negative	negative	YES	...	3.0	6.0	3.0	4.0	0.75	
1	2	55	male	rural	12	2	branded liquor	negative	negative	YES	...	3.0	6.0	3.0	4.0	0.75	
2	3	55	male	rural	12	2	branded liquor	negative	negative	YES	...	3.0	6.0	3.0	4.0	0.75	
3	4	55	male	rural	12	2	branded liquor	negative	negative	NO	...	3.0	6.0	3.0	4.0	0.75	
4	5	55	female	rural	12	2	branded liquor	negative	negative	YES	...	3.0	6.0	3.0	4.0	0.75	

5 rows × 42 columns

```
df.shape
```

```
(950, 42)
```

```
df.isnull().any()
```

```
df.isnull().sum()
```

```
S.NO                                                  0
Age                                                   0
Gender                                                0
Place(location where the patient lives)             134
Duration of alcohol consumption(years)                0
Quantity of alcohol consumption (quarters/day)        0
Type of alcohol consumed                              0
Hepatitis B infection                                 0
Hepatitis C infection                                 0
Diabetes Result                                       0
Blood pressure (mmhg)                                 0
Obesity                                               0
Family history of cirrhosis/ hereditary               0
TCH                                                 359
TG                                                  359
LDL                                                 359
HDL                                                 368
Hemoglobin  (g/dl)                                    0
PCV  (%)                                             30
```

```
df.isnull().sum()
```

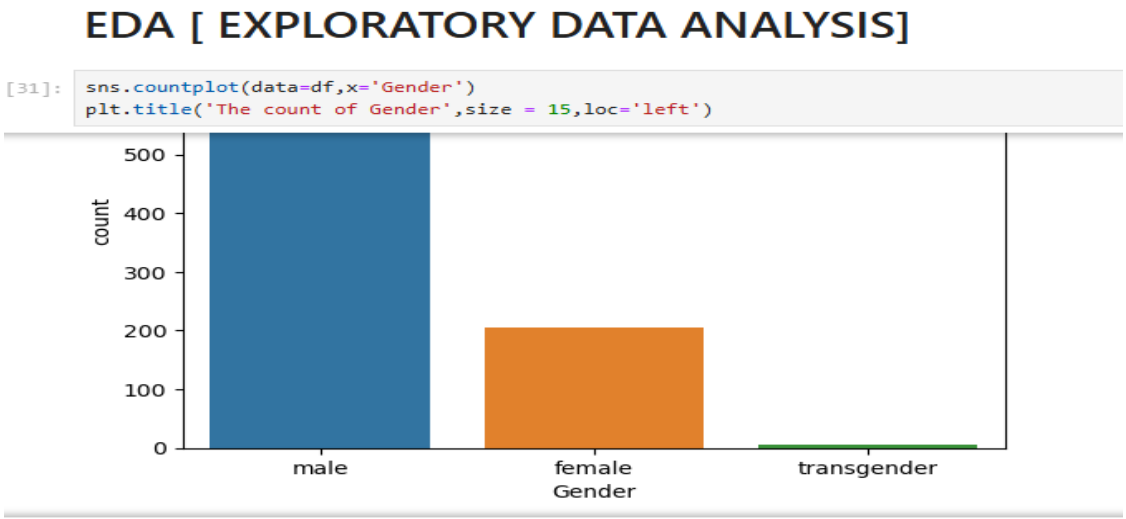
```
S.NO                                                  0
Age                                                   0
Gender                                                0
Place(location where the patient lives)               0
Duration of alcohol consumption(years)                0
Quantity of alcohol consumption (quarters/day)        0
Type of alcohol consumed                              0
Hepatitis B infection                                 0
Hepatitis C infection                                 0
Diabetes Result                                       0
Blood pressure (mmhg)                                 0
Obesity                                               0
Family history of cirrhosis/ hereditary               0
TCH                                                   0
TG                                                    0
LDL                                                   0
HDL                                                   0
Hemoglobin  (g/dl)                                    0
PCV  (%)                                              0
```

📁 Milestone 3: Exploratory Data Analysis

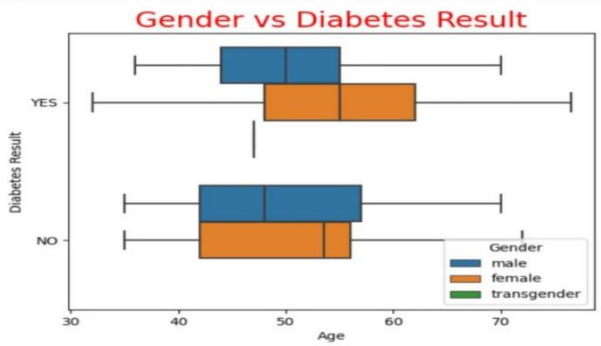
- Descriptive analysis is to study the basic features of data with the statistical process. Here pandas has a worthy function called describe.
- Visual analysis is the process of using visual representations, such as charts, plots, and graphs, to explore and understand data
- Univariate analysis, Bivariate analysis and Multivariate analysis

---

OUTPUT SCREENSHOT

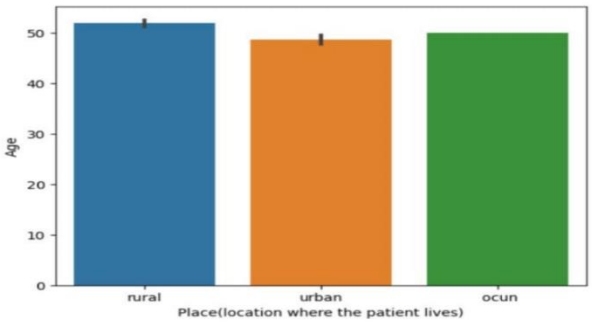


```
sns.boxplot(x='Age',y='Diabetes Result',data=df,hue='Gender')
plt.title('Gender vs Diabetes Result',color='red',size=20)
plt.show()
```



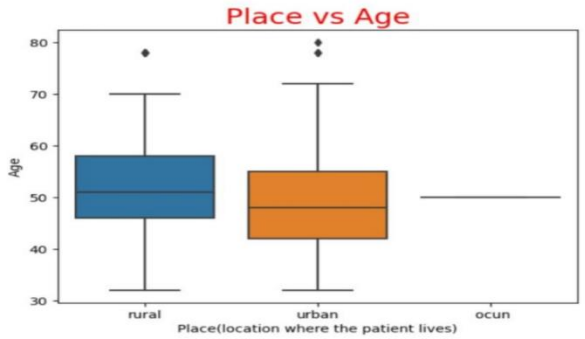

```
sns.barplot(x=df['Place(location where the patient lives)'],y=df['Age'])
```

```
<Axes: xlabel='Place(location where the patient lives)', ylabel='Age'>
```

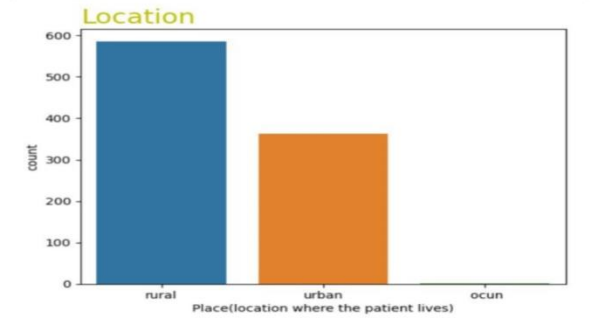



```
sns.boxplot(x='Place(location where the patient lives)',y='Age',data=df)
plt.title('Place vs Age',color='red',size=20)
```

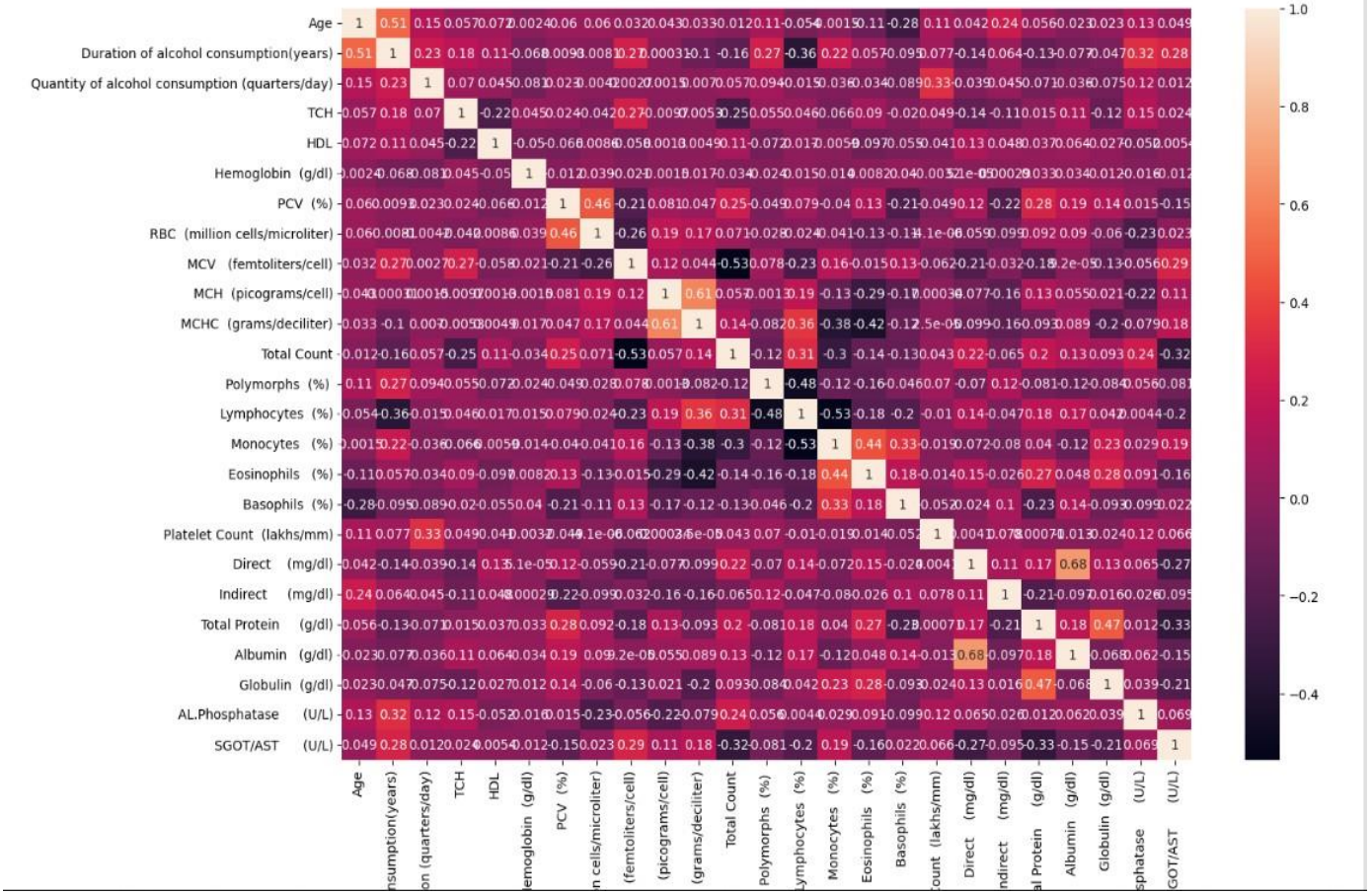
```
Text(0.5, 1.0, 'Place vs Age')
```

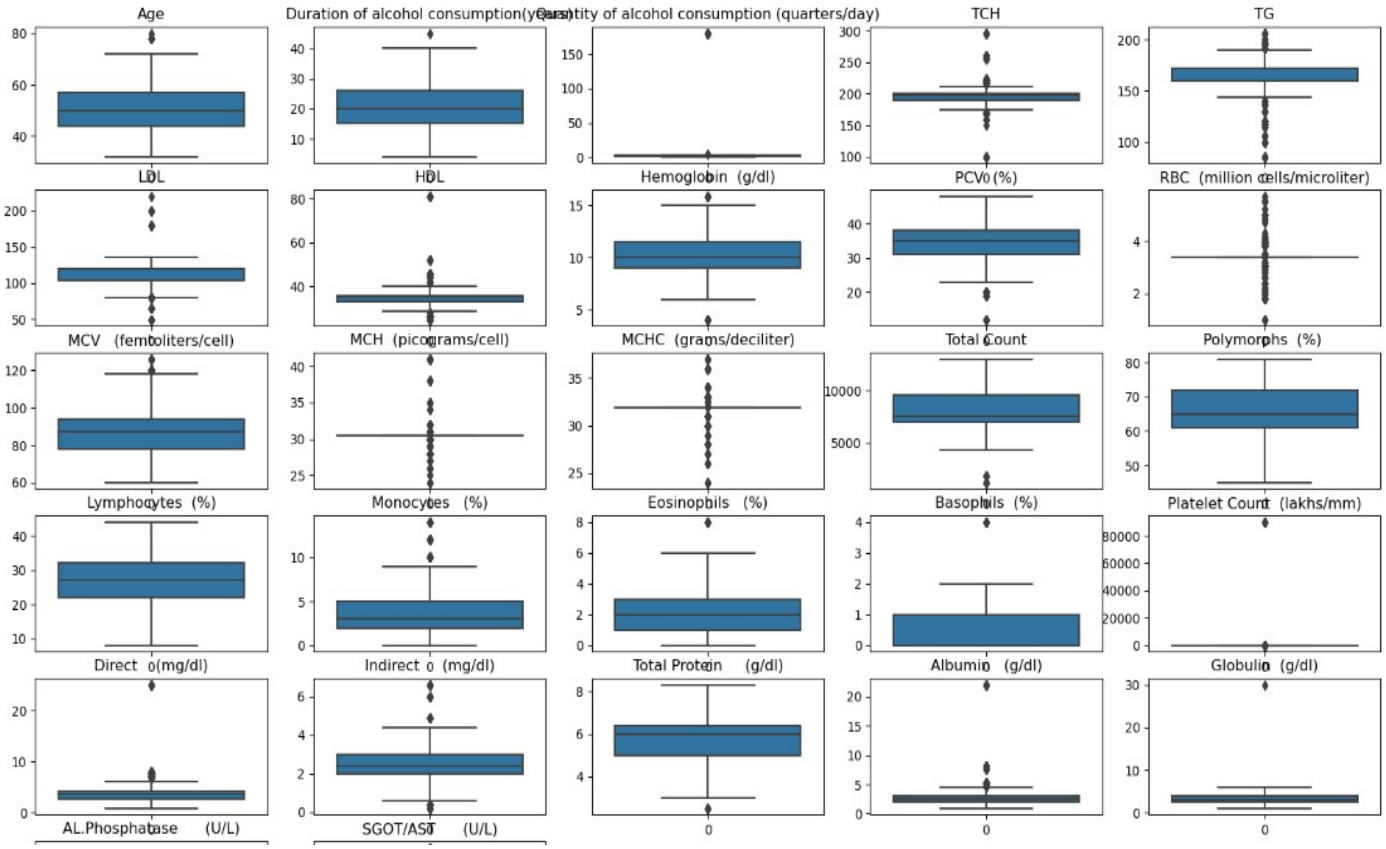



```
sns.countplot(data=df,x='Place(location where the patient lives)')
plt.title("Location",color='y',size=20,loc='left')
plt.show()
```




```
plt.figure(figsize=(15,10))
sns.heatmap(df.corr(),annot=True)
plt.show()
```





📁 Milestone 4: Model building

- Training the model in multiple algorithms.
- logistic regression, logistic regression cv, XGBclassifier, RidgeClassifier, KNN classifier, Random forest classifier and are initialised and training data is passed to the model with fit() function.

OUTPUT SCREENSHOT

Naive Bayes

```
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb.fit(x_train,y_train)
```

```
▾ GaussianNB
GaussianNB()
```

```
x_train
```

```
y_train
```

Random Forest

```
from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier()
rf.fit(x_train,y_train)
```

```
▾ RandomForestClassifier
RandomForestClassifier()
```

```
x_train
```

```
y_train
```

Logistic Regression

```
from sklearn.linear_model import LogisticRegression
log = LogisticRegression()
logistic = log.fit(x_train,y_train)
```

```
x_train
```

```
y_train
```

KNN

```
from sklearn.neighbors import KNeighborsClassifier
```

```
knn = KNeighborsClassifier()
knn.fit(x_train,y_train)
```

```
▾ KNeighborsClassifier
KNeighborsClassifier()
```

```
print ("x_Train",x_train)
print("y_Train",y_train)
```

	Name	Accuracy	F1 Score	Precision	Recall
0	logistic regression	79.47	85.17	91.80	79.43
1	logistic regression CV	81.58	86.49	91.80	81.75
2	naive bayes	35.79	0.00	0.00	0.00
3	XGBoost	35.79	6.15	3.28	50.00
4	Ridge classifier	84.21	88.37	93.44	83.82
5	Random Forest	35.79	0.00	0.00	0.00
6	Support Vector Classifier	35.79	0.00	0.00	0.00
7	KNN	86.32	89.84	94.26	85.82

📁 Milestone 5: Performance Testing & Hyperparameter Tuning

- Testing the model performance
- The function is called by passing the train, test variables. The models are returned and stored in variables as shown below. Clearly, we can see that the models are not performing well on the data. So, we'll optimise the hyperparameters of models using GridsearchCV.

OUTPUT SCREENSHOT

Model Testing

[163]:
```
Diabetes_Results = ['Yes','No']
```

[164]:
```
pred_value = knn.predict([[12.2,13,14,111,3456,245,367,1,9,87,65,34,69,23,55.55,667.67,135,1,4,6,89.876,22,45,60.06,43.356,23.21,8,90.9,73,34,31]])
prediction = int(pred_value[0])
```

[165]:
```
prediction = Diabetes_Results[prediction]
```

[166]:
```
prediction
```

[166]:
```
'Yes'
```

[167]:
```
pd.set_option('display.max_columns', None)
df.head()
```

[167]:

67]:

	Age	Gender	Place(location where the patient lives)	Duration of alcohol consumption(years)	Quantity of alcohol consumption (quarters/day)	Type of alcohol consumed	Diabetes Result	Blood pressure (mmhg)	Obesity	Family history of cirrhosis/ hereditary	Hemoglobin (g/dl)	PCV (%)	RBC (million cells/microliter)	(femtoliter
0	55.0	1	1	12.0	2.0	2	1	32	1	1	12.0	40.0	3.390704	
1	55.0	1	1	12.0	2.0	2	1	32	1	1	9.2	40.0	3.390704	
2	55.0	1	1	12.0	2.0	2	1	32	0	1	10.2	40.0	3.390704	
3	55.0	1	1	12.0	2.0	2	0	32	0	1	7.2	40.0	3.390704	
4	55.0	0	1	12.0	2.0	2	1	32	0	1	10.2	40.0	3.390704	

168]:
```
# Save the cleaned and processed DataFrame to a CSV file
df.to_csv('cleaned_data.csv', index=False)
df.head()
```

168]:

	Age	Gender	Place(location where the patient lives)	Duration of alcohol consumption(years)	Quantity of alcohol consumption (quarters/day)	Type of alcohol consumed	Diabetes Result	Blood pressure (mmhg)	Obesity	Family history of cirrhosis/ hereditary	Hemoglobin (g/dl)	PCV (%)	RBC (million cells/microliter)	(femtoliter
0	55.0	1	1	12.0	2.0	2	1	32	1	1	12.0	40.0	3.390704	
1	55.0	1	1	12.0	2.0	2	1	32	1	1	9.2	40.0	3.390704	
2	55.0	1	1	12.0	2.0	2	1	32	0	1	10.2	40.0	3.390704	
3	55.0	1	1	12.0	2.0	2	0	32	0	1	7.2	40.0	3.390704	
4	55.0	0	1	12.0	2.0	2	1	32	0	1	10.2	40.0	3.390704	

[ ]: