Project Development Phase - Sprint Delivery Plan

Sprint 2 -Dataset Collection and Pre-Processing

Date	18 November 2022
Team ID	PNT2022TMID52974
Project Name	Statistical Machine Learning Approaches to Liver Disease Prediction

Downloaded Dataset:

indian_liver_patient.csv (23.93 kB)

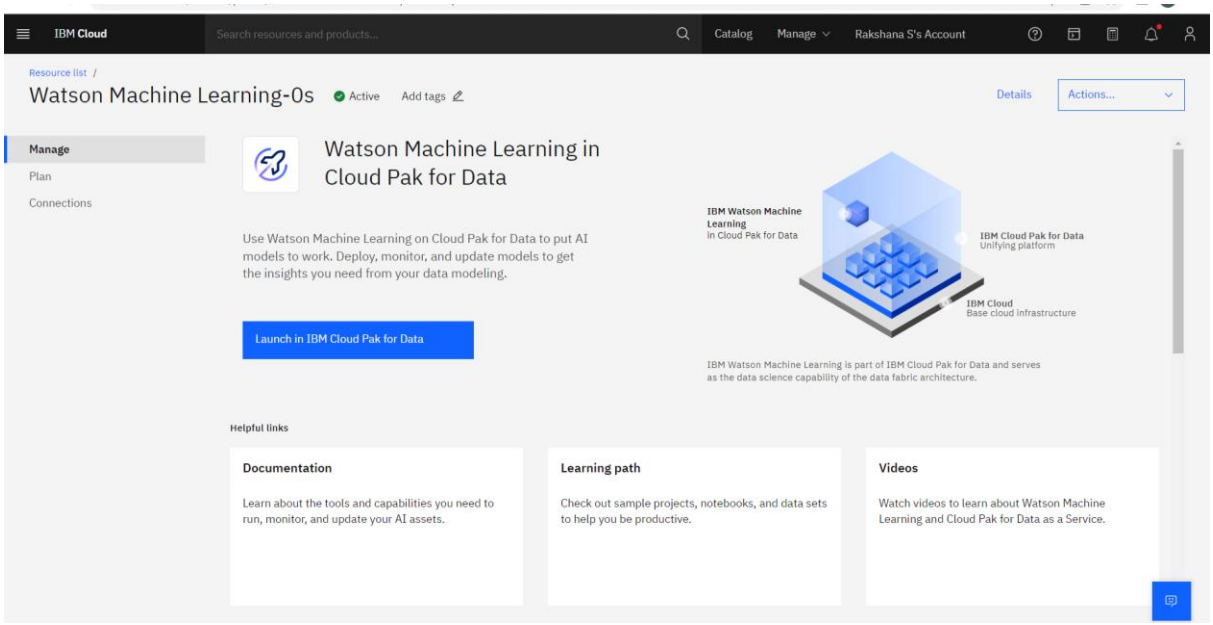
Detail Compact Column 10 of 1

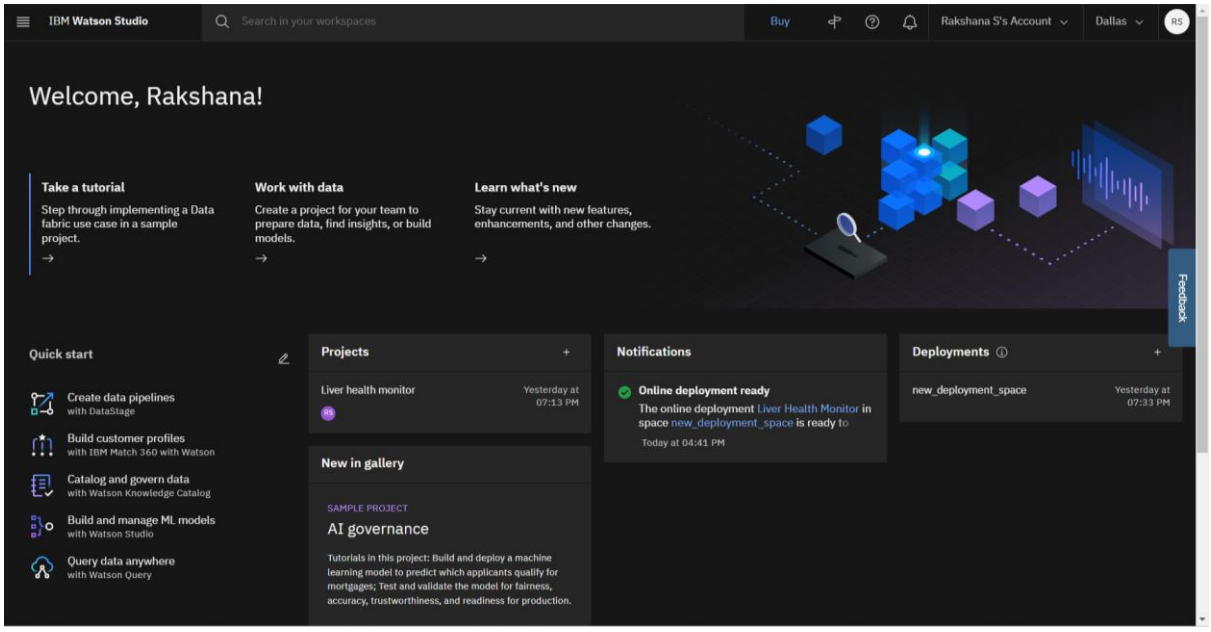
# Age Age of the patients	Gender Sex of the patients	# Total_Bilirubin Total Billirubin in mg/dL	# Direct_Bilirubin Conjugated Billirubin in mg/dL	# Alkaline_Phospho... ALP in IU/L
4 90	Male 76% Female 24%	0.4 75	0.1 19.7	63 2110
65	Female	0.7	0.1	187
62	Male	10.9	5.5	699
62	Male	7.3	4.1	490
58	Male	1	0.4	182
72	Male	3.9	2	195
46	Male	1.8	0.7	208
26	Female	0.9	0.2	154

# Alamine_Aminotr... ALT in IU/L	# Aspartate_Amino... AST in IU/L	# Total_Protiens Total Proteins g/dL	# Albumin Albumin in g/dL	# Albumin_and_Glo... A/G ratio
10 2000	10 4929	2.7 9.6	0.9 5.5	0.3
16	18	6.8	3.3	0.9
64	100	7.5	3.2	0.74
60	68	7	3.3	0.89
14	20	6.8	3.4	1
27	59	7.3	2.4	0.4
19	14	7.6	4.4	1.3
16	12	7	3.5	1

Dataset processing & Visualization:

- Login IBM Watson Studio to Visualize the data:





- Import Libraries:

IBM Watson Studio | Search in your workspaces | Buy | Rakshana S's Account | Dallas | RS

Projects / Liver health monitor / Liver_Patient_Analysis

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pickle
```

- Reading Dataset:

In [3]:
```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='81NwbGz-nOfetpoQbtERs4vVLTRPbT5SX0ohN_GSTc4d',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'liverhealthmonitor-donotdelete-pr-z7bxt6q1mu2huc'
object_key = 'indian_liver_patient.csv'

body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

data = pd.read_csv(body)
data.head()
```

Out[3]:

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	Dataset
0	65	Female	0.7	0.1	187	16	18	6.8	3.3	0.90	1
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	1
2	62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89	1
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	1.00	1
4	72	Male	3.9	2.0	195	27	59	7.3	2.4	0.40	1

- Exploratory Data Analysis:

Head() :To check the first five rows of the dataset, we have a function called head( ).

Tail(): To check the last five rows of the dataset, we have a function called tail().

In [4]:
```
data.head()
```

Out[4]:

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	Dataset
0	65	Female	0.7	0.1	187	16	18	6.8	3.3	0.90	1
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	1
2	62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89	1
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	1.00	1
4	72	Male	3.9	2.0	195	27	59	7.3	2.4	0.40	1

In [5]:
```
data.tail()
```

Out[5]:

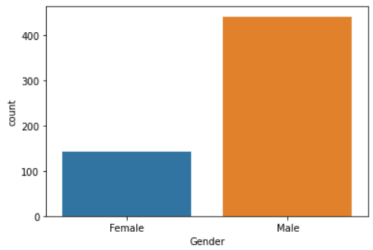
	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	Dataset
578	60	Male	0.5	0.1	500	20	34	5.9	1.6	0.37	2
579	40	Male	0.6	0.1	98	35	31	6.0	3.2	1.10	1
580	52	Male	0.8	0.2	245	48	49	6.4	3.2	1.00	1
581	31	Male	1.3	0.5	184	29	32	6.8	3.4	1.00	1
582	38	Male	1.0	0.3	216	21	24	7.3	4.4	1.50	2

- Checking for Null values and Handling Null values:

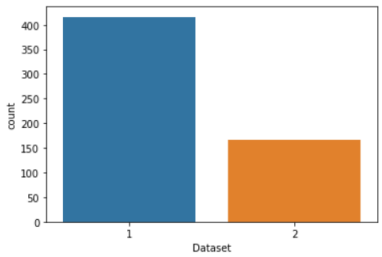
```
In [8]: data.isnull().any()

Out[8]: Age                          False
        Gender                       False
        Total_Bilirubin              False
        Direct_Bilirubin             False
        Alkaline_Phosphotase         False
        Alamine_Aminotransferase     False
        Aspartate_Aminotransferase   False
        Total_Protiens               False
        Albumin                      False
        Albumin_and_Globulin_Ratio    True
        Dataset                      False
        dtype: bool

In [9]: data.isnull().sum()

Out[9]: Age                          0
        Gender                       0
        Total_Bilirubin              0
        Direct_Bilirubin             0
        Alkaline_Phosphotase         0
        Alamine_Aminotransferase     0
        Aspartate_Aminotransferase   0
        Total_Protiens               0
        Albumin                      0
        Albumin_and_Globulin_Ratio   4
        Dataset                      0
        dtype: int64

In [10]: data['Albumin_and_Globulin_Ratio']=data['Albumin_and_Globulin_Ratio'].fillna(0)
         data.isnull().sum()

Out[10]: Age                          0
         Gender                       0
         Total_Bilirubin              0
         Direct_Bilirubin             0
         Alkaline_Phosphotase         0
         Alamine_Aminotransferase     0
         Aspartate_Aminotransferase   0
```

- Data Visualization:

```
In [11]: sns.countplot(data=data,x='Gender',label='Count')
         m,f=data['Gender'].value_counts()
         print("Number of Males:",m)
         print("Number of Females:",f)

         Number of Males: 441
         Number of Females: 142
```




```
In [12]: sns.countplot(data=data,x='Dataset')
         LD,NLD=data['Dataset'].value_counts()
         print("Liver disease patients:",LD)
         print("Non Liver disease patients:",NLD)

         Liver disease patients: 416
         Non Liver disease patients: 167
```



- Splitting the dataset into dependent and independent variables
- Split the features into independent and dependent variables to train set and test set :

```
In [13]: x=data.iloc[:,0:-1]
         y=data.iloc[:,-1]

In [14]: from sklearn.model_selection import train_test_split
         xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2)

In [15]: xtrain.shape

Out[15]: (466, 10)

In [16]: xtest.shape

Out[16]: (117, 10)
```