**DM Assignment**

# Team no:

**Team Name:** **Team\_Devops Members:**

* Charan -21BAD014
* Barath Kumar – 21BAD012
* Kavin Bharthi – 21BAD031

**1.BINING:**

## MEAN

**CODE:**

import numpy as np import math

from sklearn.datasets import load\_iris from sklearn import datasets, linear\_model, metrics

dataset = load\_iris() a = dataset.data b = np.zeros(150)

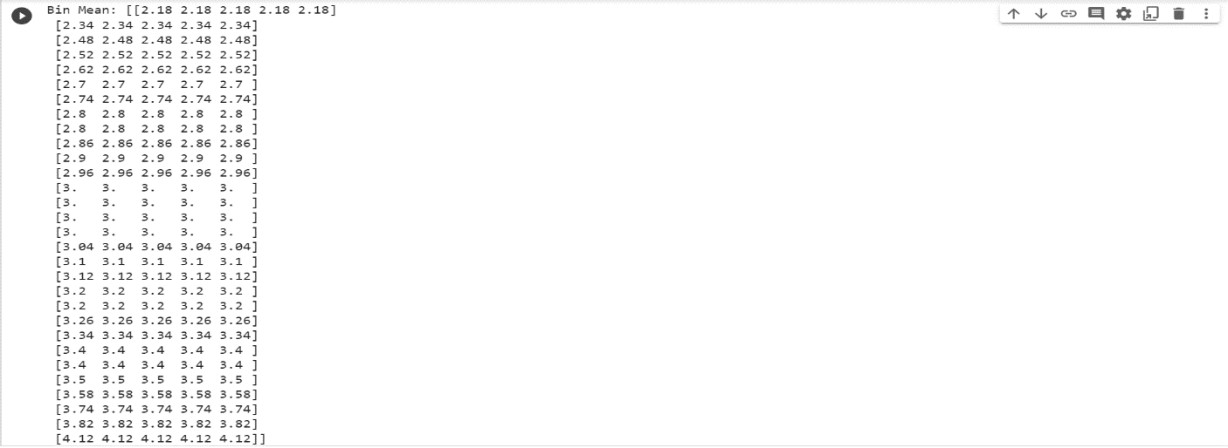
b a for i in range (150):

b[i]=a[i,1] b b=np.sort(b) bin1=np.zeros((30,5)) bin2=np.zeros((30,5)) bin3=np.zeros((30,5)) bin1 bin2

bin3 for i in range (0,150,5):

k=int(i/5) mean=(b[i] + b[i+1] + b[i+2] + b[i+3] + b[i+4])/5 for j in range(5): bin1[k,j]=mean print("Bin Mean:",bin1)

**OUTPUT:**



## BOUNDARIES

**CODE:** for i in range (0,150,5):

k=int(i/5) for j in range (5): if (b[i+j]-b[i]) < (b[i+4]-b[i+j]): bin2[k,j]=b[i] else:

bin2[k,j]=b[i+4] print("Bin Boundaries: \n",bin2) **OUTPUT :**



**MEDIAN CODE:** for i in range (0,150,5):

k=int(i/5) for j in range (5): bin3[k,j]=b[i+2]

print("Bin Median: \n",bin3)

**OUTPUT:**

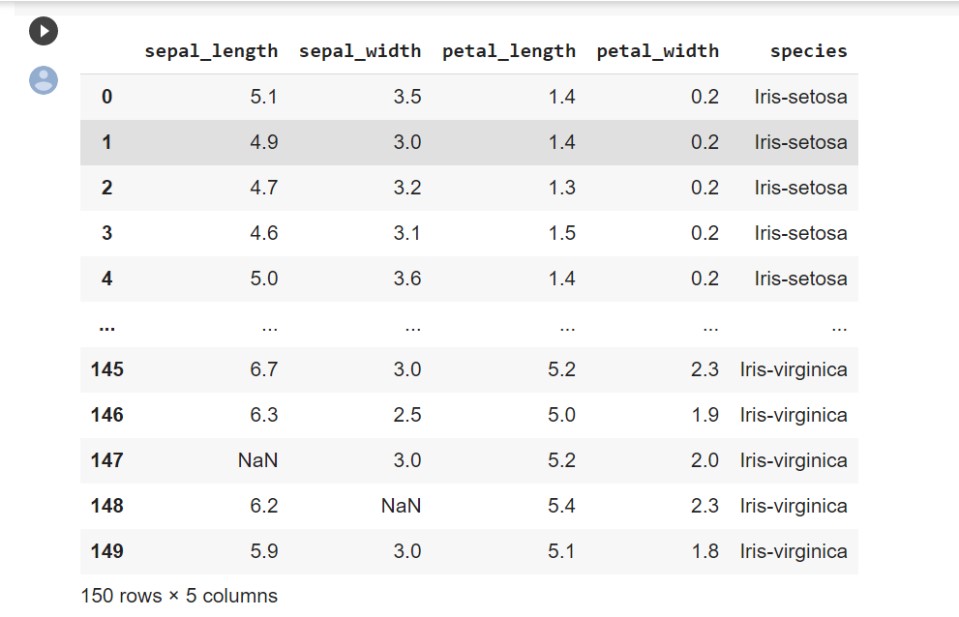


# 2.DATA CLEANING

**CODE AND OUTPUT:**

from google.colab import drive drive.mount('/content/drive') cd/content/drive//My Drive/Dataset import pandas as pd import numpy as np import matplotlib.pyplot as plt import seaborn as sns dataset=pd.read\_csv("iris.csv")

Dataset



dataset.isnull().sum()

plt.figure(figsize=(10,6)) sns.displot(

data=dataset.isna().melt(value\_name="missing"),

y="variable", hue="missing", multiple="fill", aspect=1.25

) plt.savefig("visualizing\_missing\_data\_with\_barplot\_Seaborn\_distplot.png")

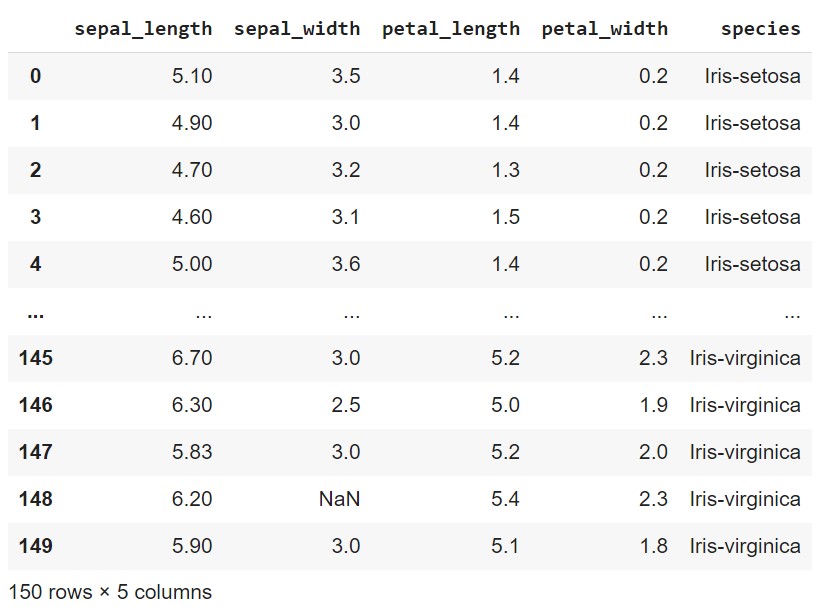
missing\_values=['??','na','X','999999',' '] df=dataset.replace(missing\_values,np.NaN)

Df

## USING MEAN

m=round(df["sepal\_length"].mean(),2) m

**5.83** df["sepal\_length"].fillna(m,inplace=True) df



## BFILL

df1=pd.read\_csv("iris.csv") df1

df1.bfill(inplace=True) df1



# FFILL

df2=pd.read\_csv("iris.csv") df2

df2.ffill(inplace=True) df2

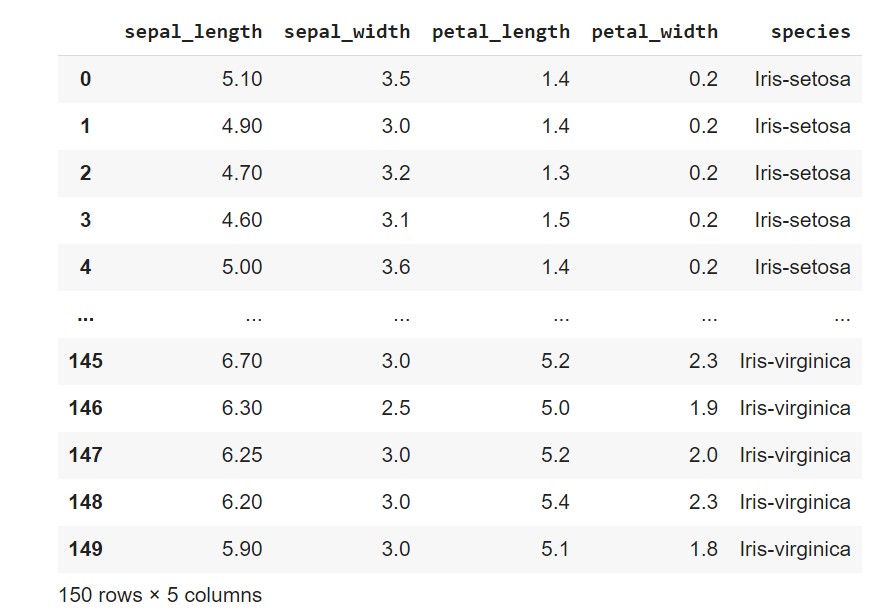


# INTERPOLATION

df3=pd.read\_csv("iris.csv") df3

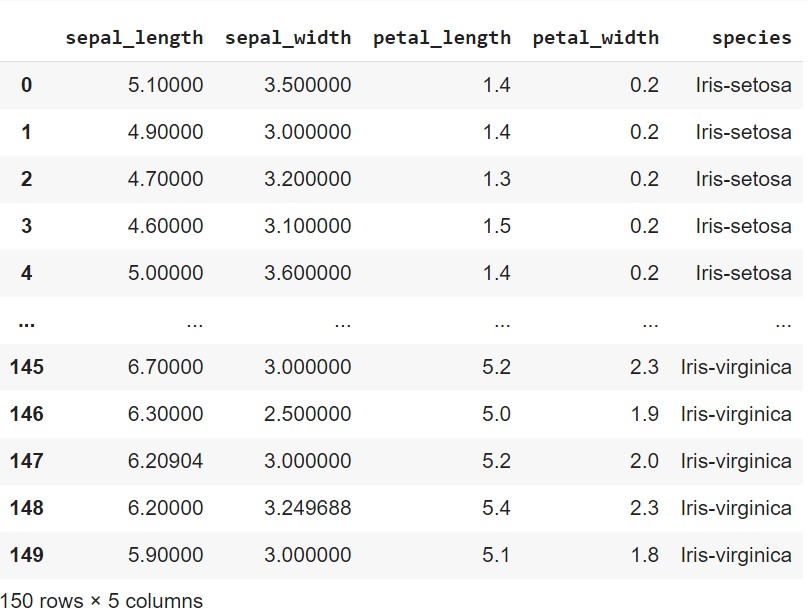
## LINEAR INTERPOLATION

df3.interpolate(method="linear",limit\_direction='forward')

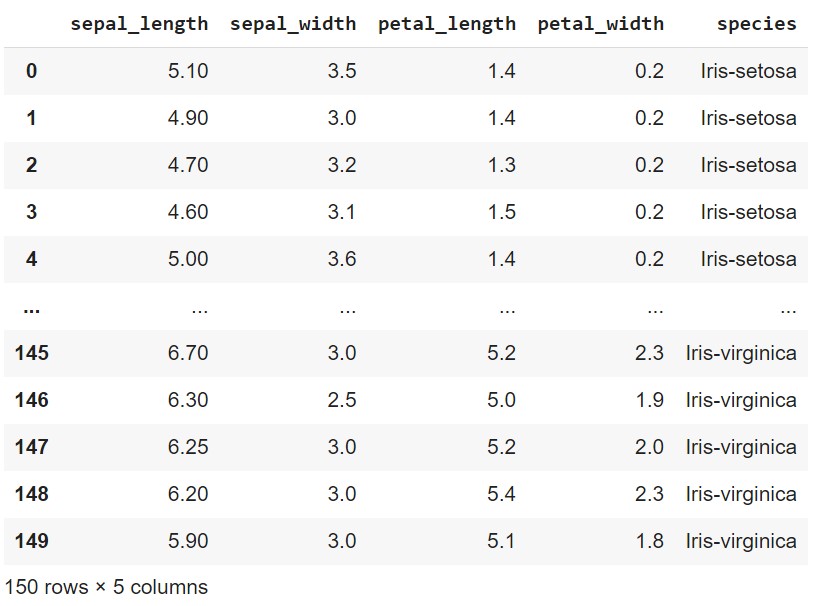


**POLYNOMIAL INTERPOLATION**

df3.interpolate(method="polynomial",order=2,limit\_direction='forward')

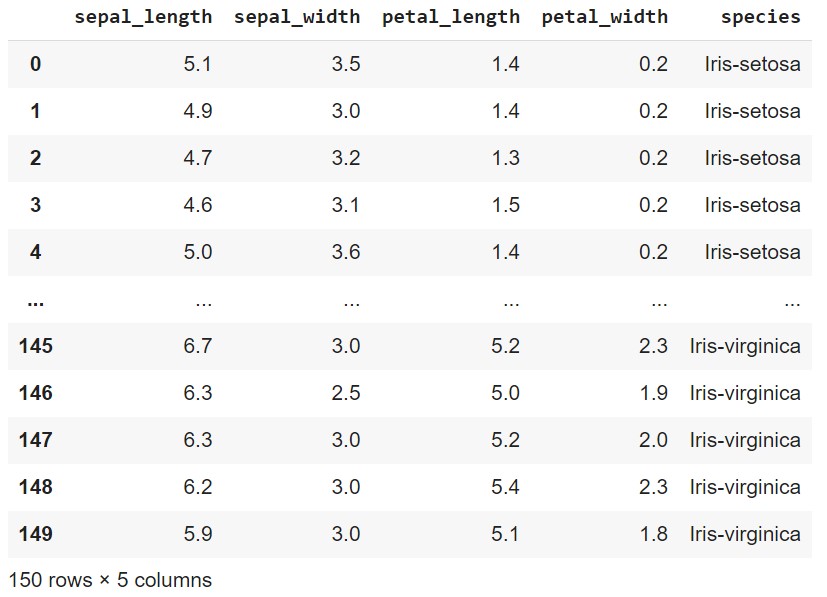


df3.interpolate(method="polynomial",order=1,limit\_direction='forward')



**PADDING**

df3.interpolate(method='pad')



df4=df3.interpolate() df4

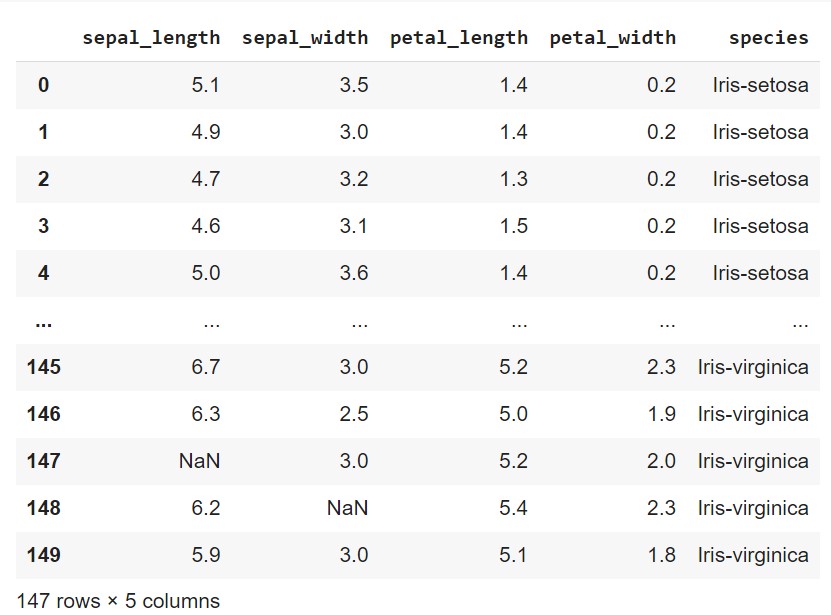


# 3. REMOVING

from google.colab import drivedrive.mount('/content/drive')cd/content/drive//My Drive/Dataset import pandas as pd import numpy as np import matplotlib.pyplot as plt import seaborn as sns dataset=pd.read\_csv("iris.csv") dataset

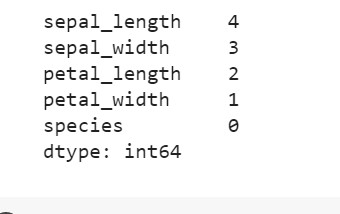
## REMOVING DUPLICATES

display(dataset.drop\_duplicates())

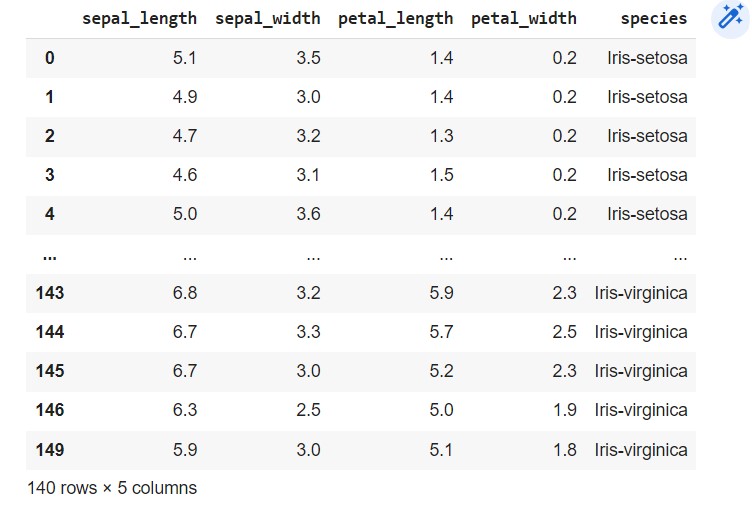


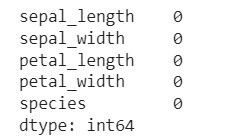
# REMOVING NULL AND NAN VALUES

dataset.isnull().sum()



dataset.dropna(inplace=True) dataset

dataset.isnull().sum()



**4. DATA TRANSFORMATION**

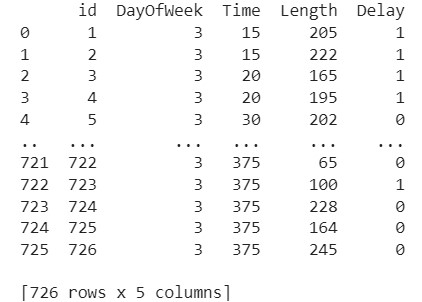
## 1. Normalization

**The dataset is normalized using two methods the first one is Minmax Scaling and second one is Log Scaling.**

## MinMax Scaling

df= pd.read\_csv("airlines.csv") df

df.drop(['Airline', 'Flight', 'AirportFrom','AirportTo'], axis=1, inplace=True) print(df)



from sklearn.preprocessing import MinMaxScaler scaler=MinMaxScaler()

print(scaler.fit(df)) print(scaler.data\_max\_) [726. 3. 375. 410. 1.] print(scaler.data\_min\_) [ 1. 3. 15. 32. 0.]

print(scaler.transform(df))

[[0. 0. 0. 0.45767196 1. ]

[0.00137931 0. 0. 0.5026455 1. ] [0.00275862 0. 0.01388889 0.35185185 1. ]

...

[0.99724138 0. 1. 0.51851852 0. ]

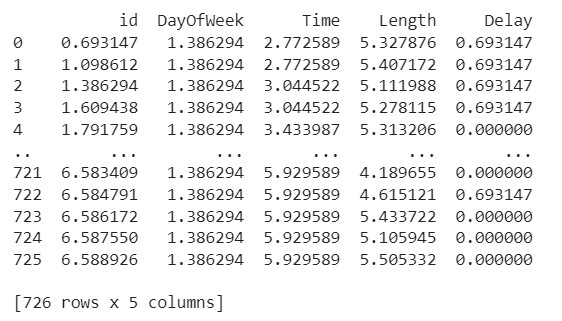
[0.99862069 0. 1. 0.34920635 0. ]

[1. 0. 1. 0.56349206 0. ]]

**2. Log Scaling:**

import numpy as np

from sklearn.preprocessing import FunctionTransformer transformer=FunctionTransformer(np.log1p) t1=transformer.transform(df) print(t1)

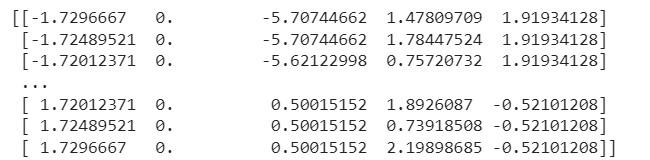


## Standardization

from sklearn.preprocessing import StandardScaler scaler=StandardScaler() print(scaler.fit(df)) print(scaler.mean\_)

[3.63500000e+02 3.00000000e+00 3.45994490e+02 1.22984848e+02

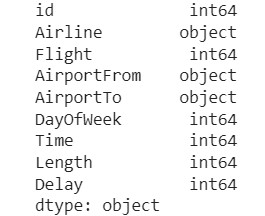
2.13498623e-01] print(scaler.transform(df))



## Type Casting

df= pd.read\_csv("airlines.csv")

df print(df.dtypes)



## Converting int to string

df['Time'] = df['Time'].astype(str) df

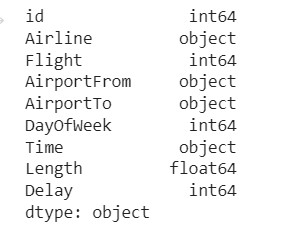
**Converting int to float**

df['Length']=df['Length'].astype(float)

## converting Boolean to int

df['Delay'] = df['Delay'].astype(int)

**Datatypes after converting.**



## 5. Data Preprocessing

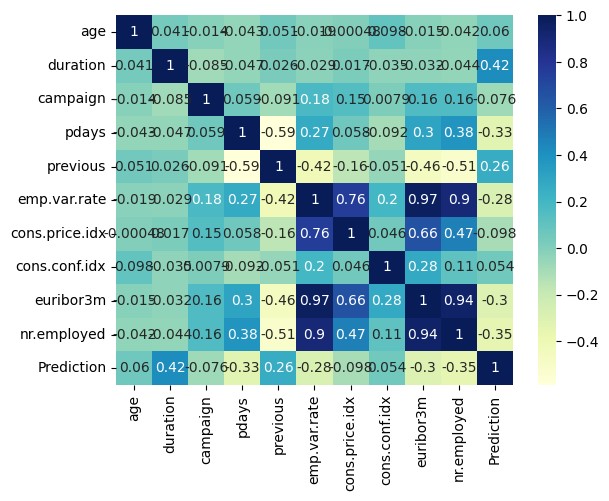
**Code:**

dataset=pd.read\_csv('bank.csv')

dataset

Before preprocessing

dataplot = sns.heatmap(dataset.corr(), cmap="YlGnBu",annot=True)



**Removing the feature with high correlation.**

high\_corr\_features = set() corr\_matrix = dataset.corr().abs() for i in range(len(corr\_matrix.columns)): for j in range(i): if corr\_matrix.iloc[i, j] >= 0.7:

colname = corr\_matrix.columns[i] high\_corr\_features.add(colname) df = dataset.drop(columns=high\_corr\_features)

## After preprocessing

dataplot = sns.heatmap(df.corr(), cmap="YlGnBu",annot=True)

