**AI&ML PRACTICAL EXAM**

**SIDDHI VAIDYA,ROLL NO.55**

**DATE:13/7/2021**

Q1. Develop to predict the mail as spam or not based on the different word

frequency using spam.csv. [20]

1. Analyse the data Visually.

2. Find limitation with data if any in developing the model.

3. Develop the model using logistic regression algorithm.

4. Perform Cross Validation and find the best results.

5. Compare the performance with K nearest neighbors algorithm.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import accuracy\_score,confusion\_matrix,mean\_absolute\_error,mean\_squared\_error

from sklearn.model\_selection import train\_test\_split

lr=pd.read\_csv('spam.csv')

lr.head(10)

x=lr['word\_freq\_mail']

y=lr['word\_freq\_receive']

sns.regplot(lr['word\_freq\_mail'],y,color='green')

x.head()

y.head()

X\_=x.values.reshape(-1,1)

x.shape

X\_.shape

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X\_,y,test\_size=0.2,random\_state=0)

LR=LinearRegression()

LR.fit(X\_train,y\_train)

y\_pred=LR.predict(X\_test)

LR.score(X\_test,y\_test)

plt.scatter(X\_test,y\_test)

plt.plot(X\_test,y\_pred,color='green')

plt.show()

df = pd.DataFrame({'Actual':y\_test,'Predicted':y\_pred})

df

new\_data=pd.DataFrame([2115,900])

new\_data

LR.predict(new\_data)

LR.predict([[4562]])

df1=df.head(25)

df1.plot(kind='bar',figsize=(16,10))

plt.grid(which='major',linestyle='-',linewidth='0.5',color='green')

plt.grid(which='minor',linestyle=':',linewidth='0.5',color='blue')

plt.show()

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

%matplotlib inline

import seaborn as sns

from sklearn import metrics

from sklearn.model\_selection import KFold,RepeatedKFold,StratifiedKFold,LeaveOneOut,ShuffleSplit

[12:01 PM, 7/13/2021] Ria C KJ: x=lr['word\_freq\_mail']

y=lr['word\_freq\_receive']

[12:01 PM, 7/13/2021] Ria C KJ: kf = KFold(n\_splits=4)

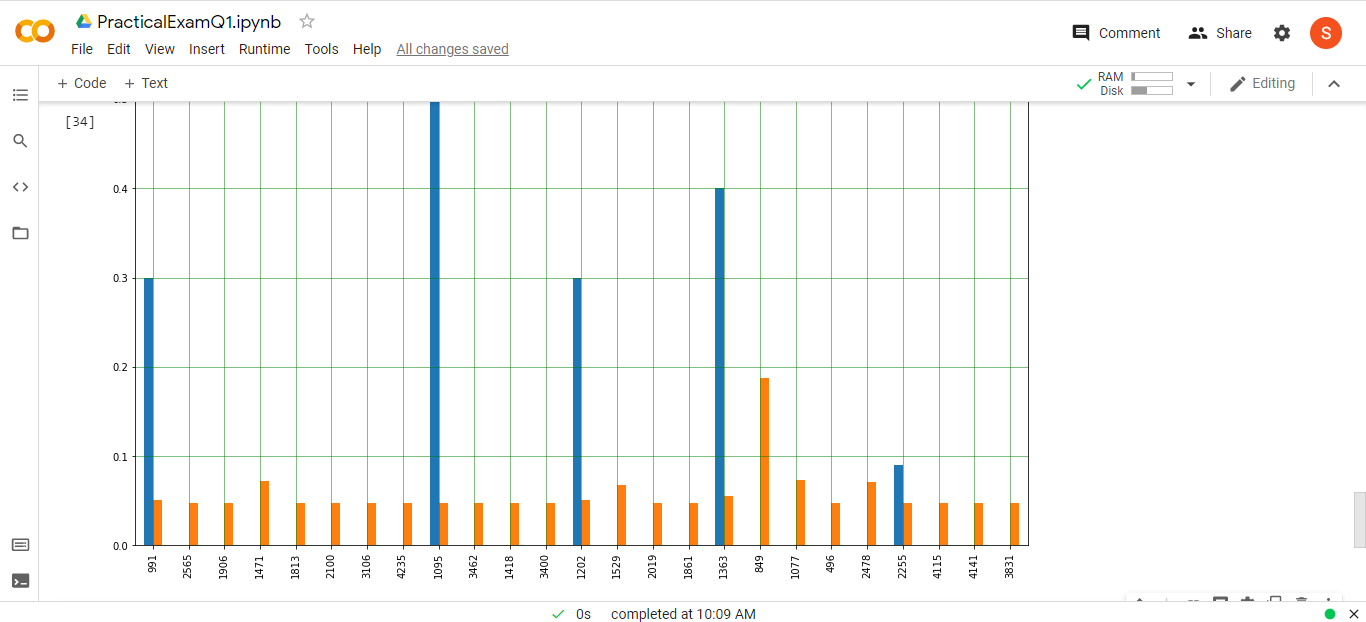
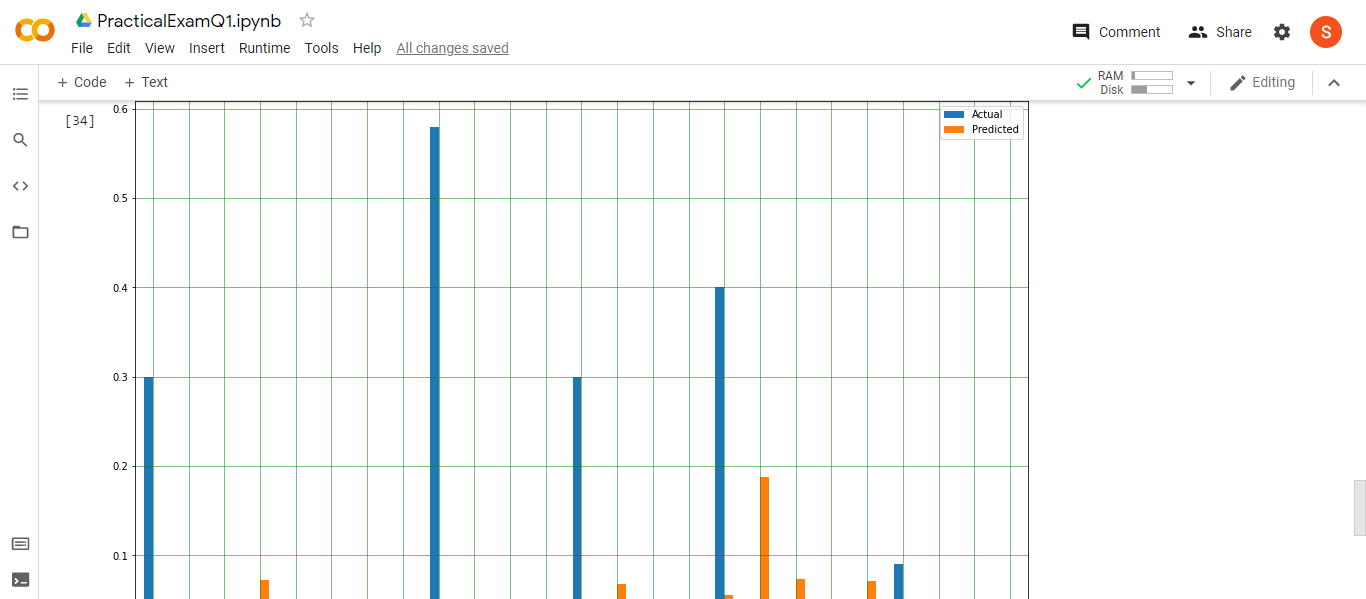
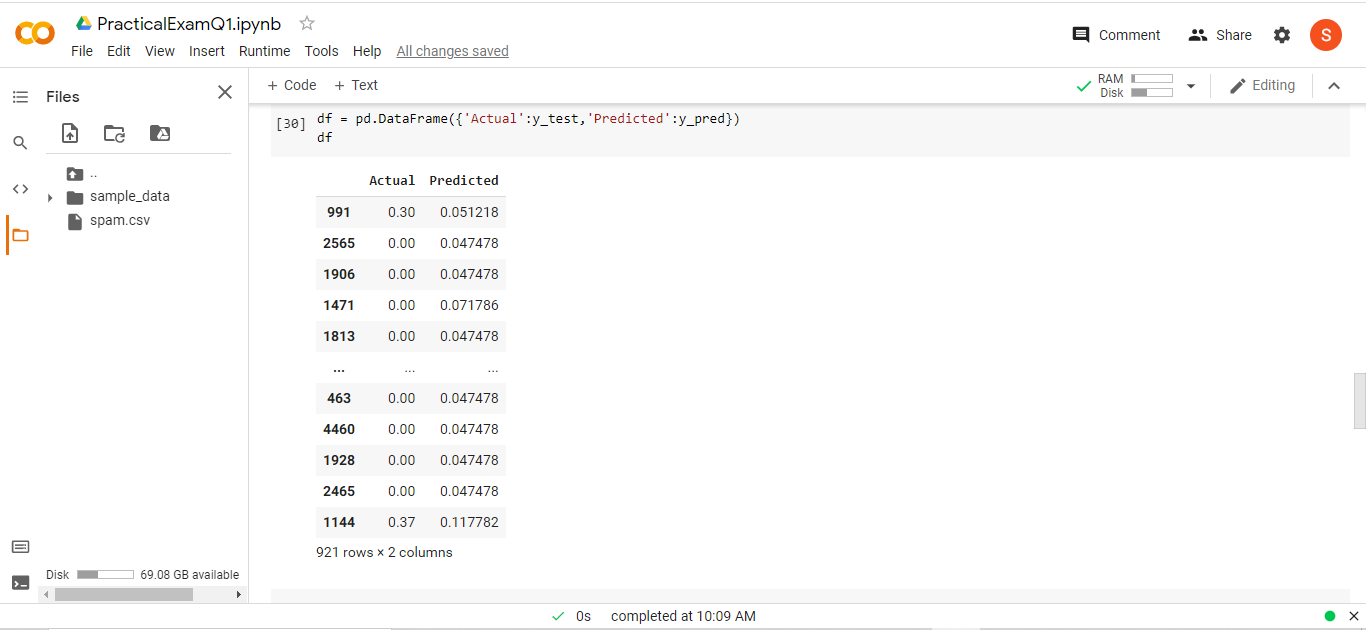
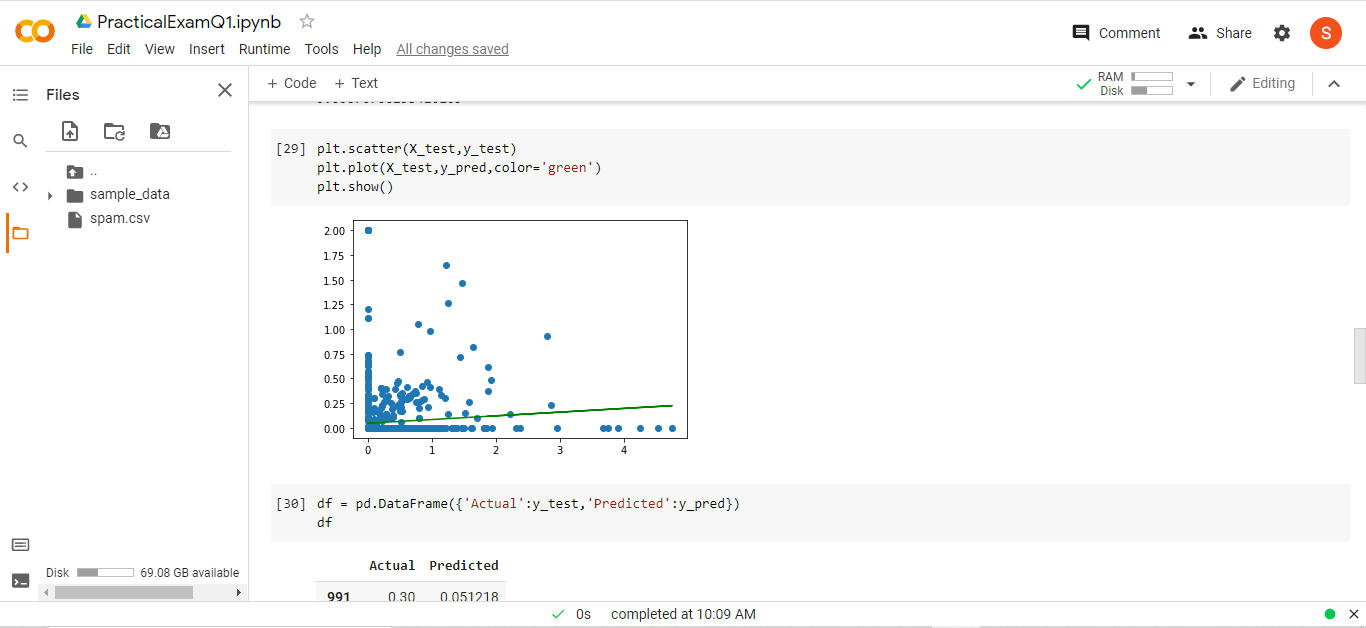
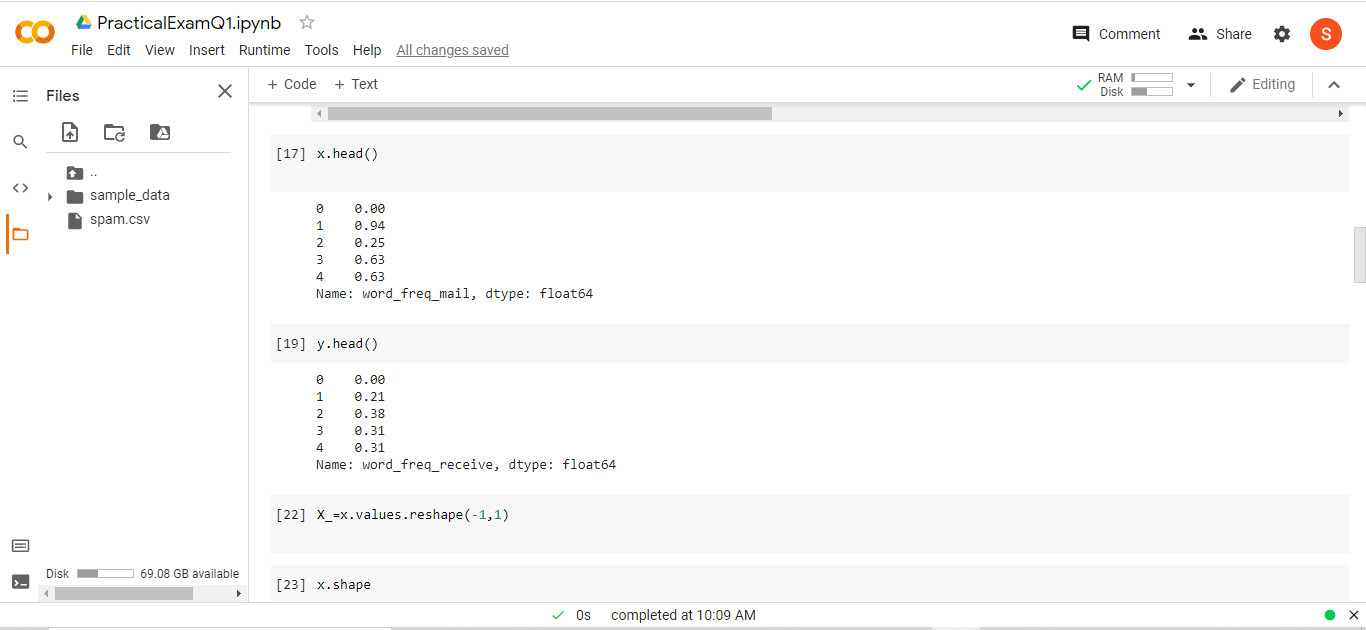
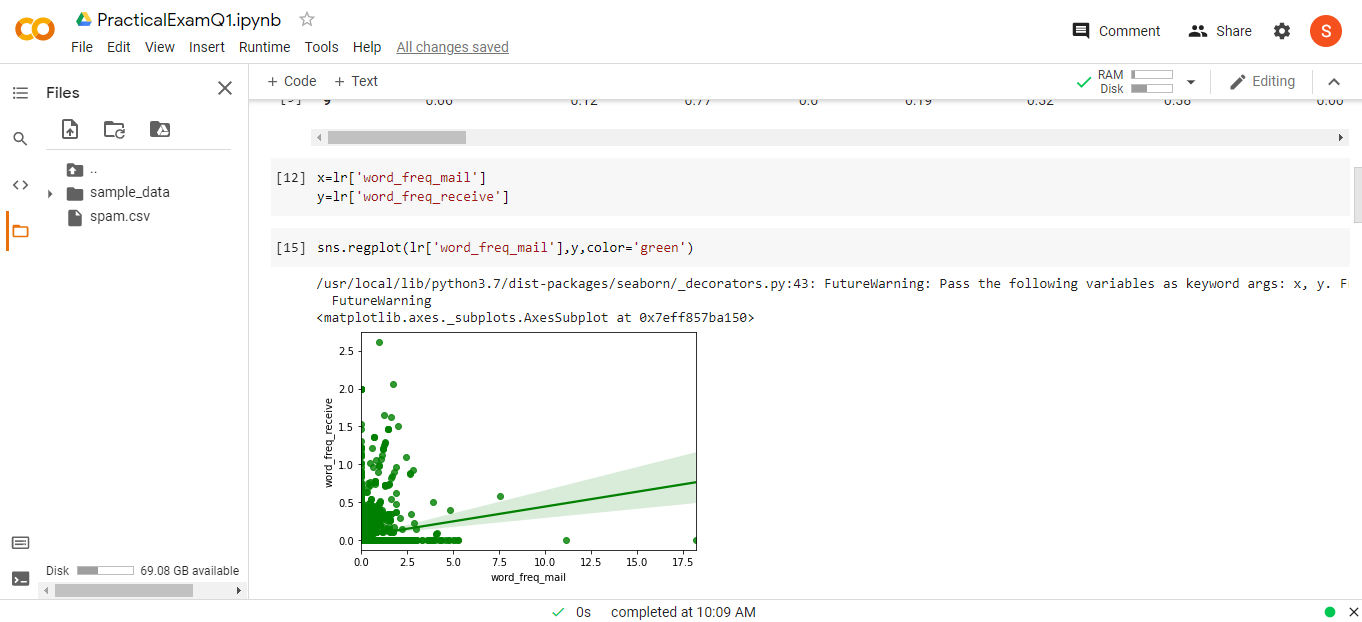
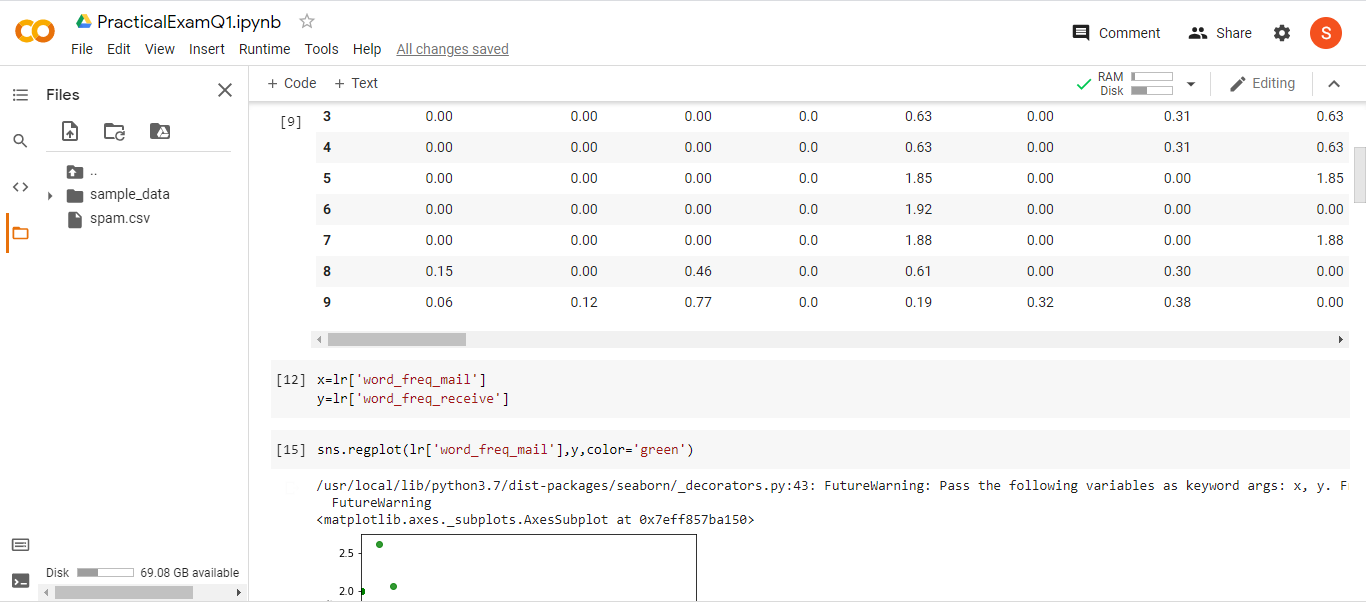
[12:01 PM, 7/13/2021] Ria C KJ: for train, test in kf.split(x):

print("%s %s" % (train,test))

[12:02 PM, 7/13/2021] Ria C KJ: rkf = RepeatedKFold(n\_splits=4,n\_repeats=3)

[12:02 PM, 7/13/2021] Ria C KJ: for train, test in rkf.split(x):

print("%s %s" % (train,test))



Q.2 Using linear regression algorithm to predict the medical insurance charges

based on the data given in med\_insurance.csv. [10]

1. Develop the model to predict the medical insurance's charges.

2. Evaluate the model.

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

import seaborn as sns

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import accuracy\_score,confusion\_matrix,mean\_squared\_error,mean\_absolute\_error

from sklearn.model\_selection import train\_test\_split

df = pd.read\_csv("med\_insurance.csv")

df

df.info() # form this we can say that there are no Null values

df.describe()

# Checking in Visual if there are any null or NA values

sns.heatmap(df.isnull())

df.nunique()

df['sex'].unique()

df['sex'].replace({'female':0,'male':1},inplace=True)

df['smoker'].unique()

df['smoker'].replace({'yes':1,'no':0},inplace=True)

df['region'].unique()

df['region'].replace({'southwest':0,'southeast':1,'northwest':2,'northeast':3},inplace=True)

df.dtypes

df

X = df.drop(["charges"], axis=1).values

y = df['charges'].values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, random\_state=0, test\_size=0.2)

regr = LinearRegression()

regr.fit(X\_train, y\_train)

df.columns

X = ['age', 'bmi', 'children', 'sex', 'smoker', 'region']

coeff\_df = pd.DataFrame(regr.coef\_,X,["Coefficient"])

coeff\_df

y\_pred=regr.predict(X\_test)

dfpred=pd.DataFrame({'Actual': y\_test, 'Predicted': y\_pred})

dfpred

dfpred.plot(kind='line',figsize=(10,8))

plt.grid(which='major',linestyle='-',linewidth='0.5')

plt.grid(which='minor',linestyle=':',linewidth='0.5')

plt.ylabel('Score')

plt.xlabel('Observations')

plt.title('Line Graph of Actual and Predicted Values')

plt.show()

print('mean\_absolute\_error: ',mean\_absolute\_error(y\_test,y\_pred))

print('mean\_squared\_error: ',mean\_squared\_error(y\_test,y\_pred))

print('mean\_absolute\_error: ',np.sqrt(mean\_absolute\_error(y\_test,y\_pred)))

