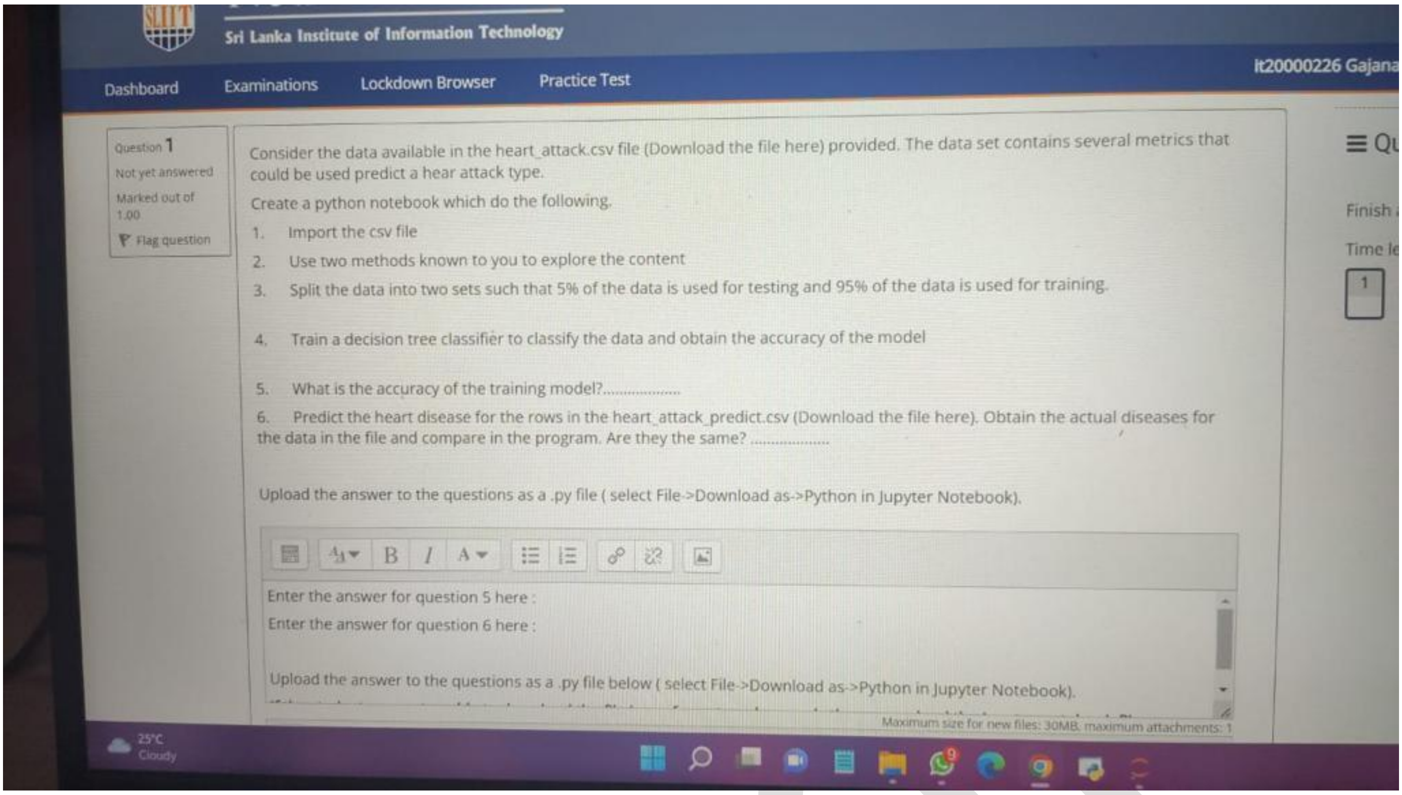
DS LAB TEST 01

Q & A

Mr_dK.pdf



Consider the data available in the heart_attack.csv file (Download the file here) provided. The data set contains several metrics that could be used predict a hear attack type.

Create a python notebook which do the following.

1. Import the csv file
2. Use two methods known to you to explore the content
3. Split the data into two sets such that 5% of the data is used for testing and 95% of the data is used for training.
4. Train a decision tree classifier to classify the data and obtain the accuracy of the model
5. What is the accuracy of the training model?..................
6. Predict the heart disease for the rows in the heart_attack_predict.csv (Download the file here). Obtain the actual diseases for the data in the file and compare in the program. Are they the same? ....

Upload the answer to the questions as a .py file (select File->Download as->Python in Jupyter Notebook).

Enter the answer for question 5 here :

Enter the answer for question 6 here :

Upload the answer to the questions as a .py file below (select File->Download as->Python in Jupyter Notebook).

Maximum size for new files: 30MB, maximum attachments: 1

Step 01-import necessary library

- Import pandas as pd
- Import numpy as np
- from sklearn.tree import DecisionTreeClassifier
- from sklearn.model_selection import train_test_split

1. Import csv file
   a. Df=pd.read_csv("heart_attack.csv")
2. Explore the data
   a. Display(df.head())
   b. Display(df.info())
   c. display(df.shape)
   d. column=['<column name 1>','<column name 2>','<column name 3>']

      df[column].describe()
3. x=df.iloc[: , [<attribute column range>]]

   y=df.iloc[: , <labeled column>]

   X_train,X_test,y_train,y_test=train_test_split(x,y,test_size=0.05)

   Or

   X_train,X_test,y_train,y_test=train_test_split(x,y,train_size=0.95)

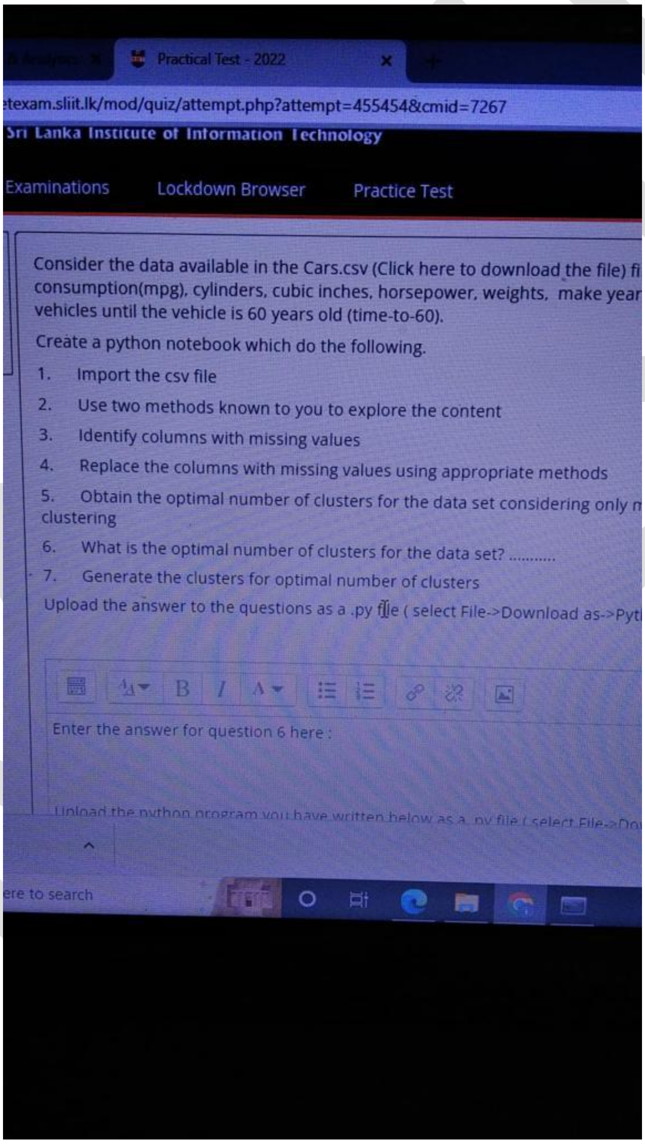
4. hart_classifire=DecisionTreeClassifire(random_state=0)

```
hart_classifire.fit(X_train,y_train)

accuracy=hart_classifire.score(X_test,y_test)

print("Accuracy of building classifier :",accuracy)
```

5. Aqurracy=(out put of the above print command)
6. Predict=hart_classifier.predict(X_test[1:10])

   Print("Predicted disease are :",predict)

Compare above out put and hart_attack_disease.csv file out put and give the answer


Practical Test - 2022
etexam.sliit.lk/mod/quiz/attempt.php?attempt=455454&cmid=7267
Sri Lanka Institute of Information Technology
Examinations
Lockdown Browser
Practice Test
ere to search

Consider the data available in the Cars.csv (Click here to download the file) fi
consumption(mpg), cylinders, cubic inches, horsepower, weights, make year
vehicles until the vehicle is 60 years old (time-to-60).

Create a python notebook which do the following.

1. Import the csv file
2. Use two methods known to you to explore the content
3. Identify columns with missing values
4. Replace the columns with missing values using appropriate methods
5. Obtain the optimal number of clusters for the data set considering only m clustering
6. What is the optimal number of clusters for the data set? ....
7. Generate the clusters for optimal number of clusters

Upload the answer to the questions as a .py file (select File->Download as->Pyt

Enter the answer for question 6 here :

Upload the python program you have written below as a .py file (select File->Do

First import necessary library,

```
Import pandas as pd

Import numpy as np

From matplotlib import pyplot as plt

Import seaborn as sbn

From sklearn.preproccessing import StanderdScaler

From sklearn.clustrt import KMeans

Sns.set(not necessary )
```

1. Df=pd.read_csv(“Cars.csv”)
2. Display(df.head())

   Display(df.info())

3. Df.isnull()   or

   Df.isna()   or

   Df.isnull().sum()   or

   Df,isna().sum()   or

   Df.info()   or

4. Count how many missing vale are there.

   ```
   Df[‘<column name>’].isnull().sum()
   ```

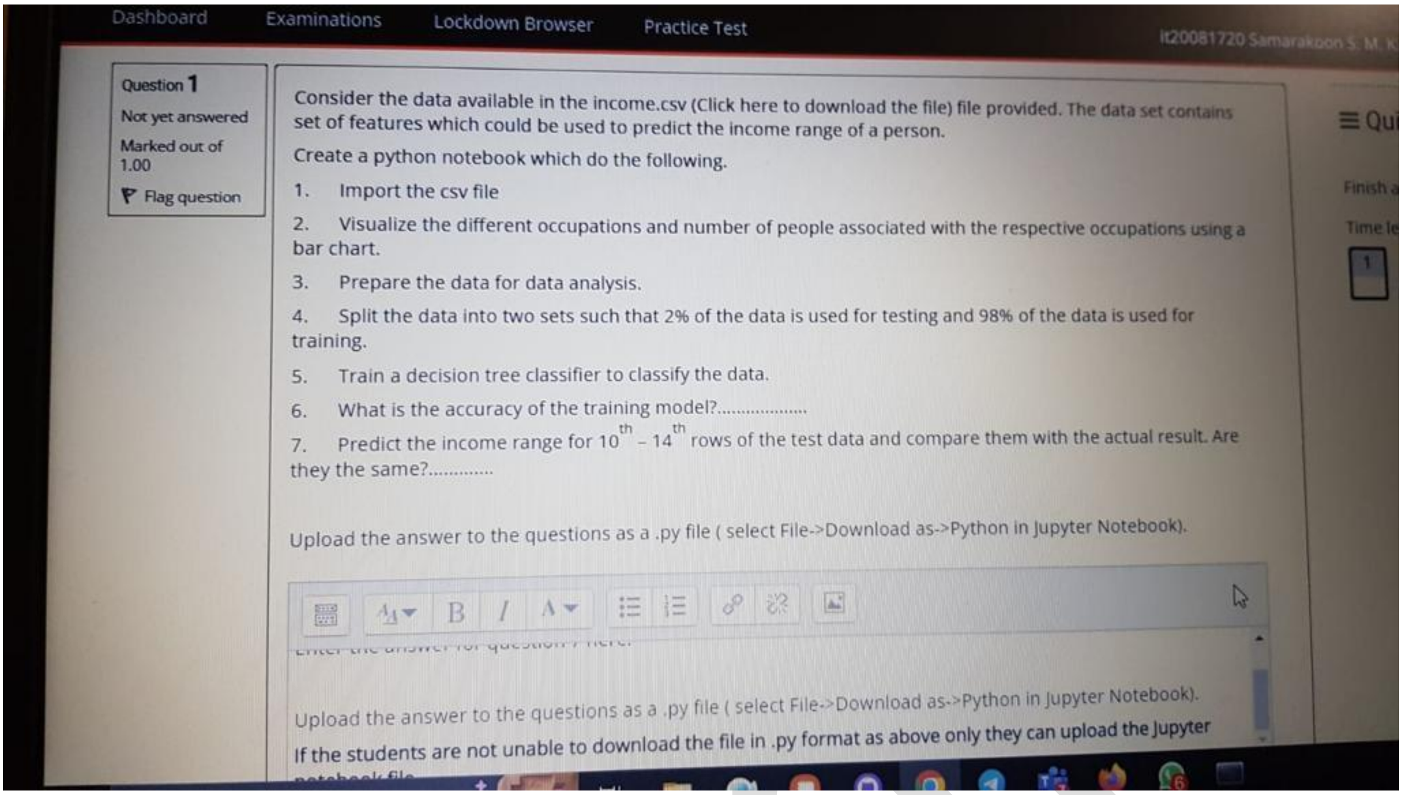
   Fill the missing value

   ```
   Df[‘column name’].fillna(Unknown,inplace=True)

                       Or

   df['column name].fillna(0,inplace=True)

                       or

   df['column name].fillna(df[‘column name’].mean,inplace=True)

                       or

   df[‘column name’].fillna(df[‘column name’].median,inplcae=True)

                       or

   df[‘column name’].fillna(df[‘column name’].mode,inplcae=True)
   ```

5.
   ```
   New_df=df.iloc[ : , [<column range>]]

   Ss=StanderdScaler()

   new_data = pd.DataFrame(ss.fit_transform(new_df), columns=[‘x axis column name’,’y axis column name’])

   wcss=[]

   for I in range(1:10)

               kmeans=KMeans(i)

               kmeans.fit(new_df)

               wcss_iter=kmeans.inertie_

               wcss.append(wcss_iter)


   cluster=range(1:10)

   plt.plot(cluster,wcss)

   plt.title(‘Elbow method’)

   plt.xlable(‘number of clusters’)

   plt.ylable(‘Within-cluster Sum of Squares’)
   ```

   considering the elbow graph find the optimal number of cluster

6. ……………………….
7.
   ```
   kmeans=KMeans(<optimal no of cluster>)

   kmeans.fit(new_df)

   identify_cluster=kmeans.fit_predict(new_df)

   cluster=df.copy()

   cluster[‘ClusterNo’]=identify_cluster

   plt.scatter(cluster[‘x axis column name’],cluster[‘y axis column
   name’],c=cluster[‘ClusterNo’],cmap=’rainbow’)
   ```



Consider the data available in the income.csv (Click here to download the file) file provided. The data set contains set of features which could be used to predict the income range of a person.

Create a python notebook which do the following.

1. Import the csv file
2. Visualize the different occupations and number of people associated with the respective occupations using a bar chart.
3. Prepare the data for data analysis.
4. Split the data into two sets such that 2% of the data is used for testing and 98% of the data is used for training.
5. Train a decision tree classifier to classify the data.
6. What is the accuracy of the training model?..................
7. Predict the income range for 10th – 14th rows of the test data and compare them with the actual result. Are they the same?............

Upload the answer to the questions as a .py file (select File->Download as->Python in Jupyter Notebook).

Enter the answer for question 1 here.

Upload the answer to the questions as a .py file (select File->Download as->Python in Jupyter Notebook).

If the students are not unable to download the file in .py format as above only they can upload the Jupyter notebook file

Import necessary library,

```
    Import pandas as pd

    Import numpy as np

    from sklearn.tree import DecisionTreeClasifire

    from sklearn.model_selection import train_test_split
```

```
df=pd.read_csv("income.csv")
```

- Chart=df['occupations'].value_counts()

  Chart.plot(kind='bar')

- Find null value and fill the null value using appropriate method and colum contain character those are convert to number using factorize method
- X=df.iloc[: , <column name>]

  Y=df.iloc[: , <column name>]

  X_train,X_test,y_train,y_test=train_tset_split(X,Y,test_size=0.02)

  Or

  X_train,X_test,y_train,y_test=train_tset_split(X,Y,train_size=0.98)

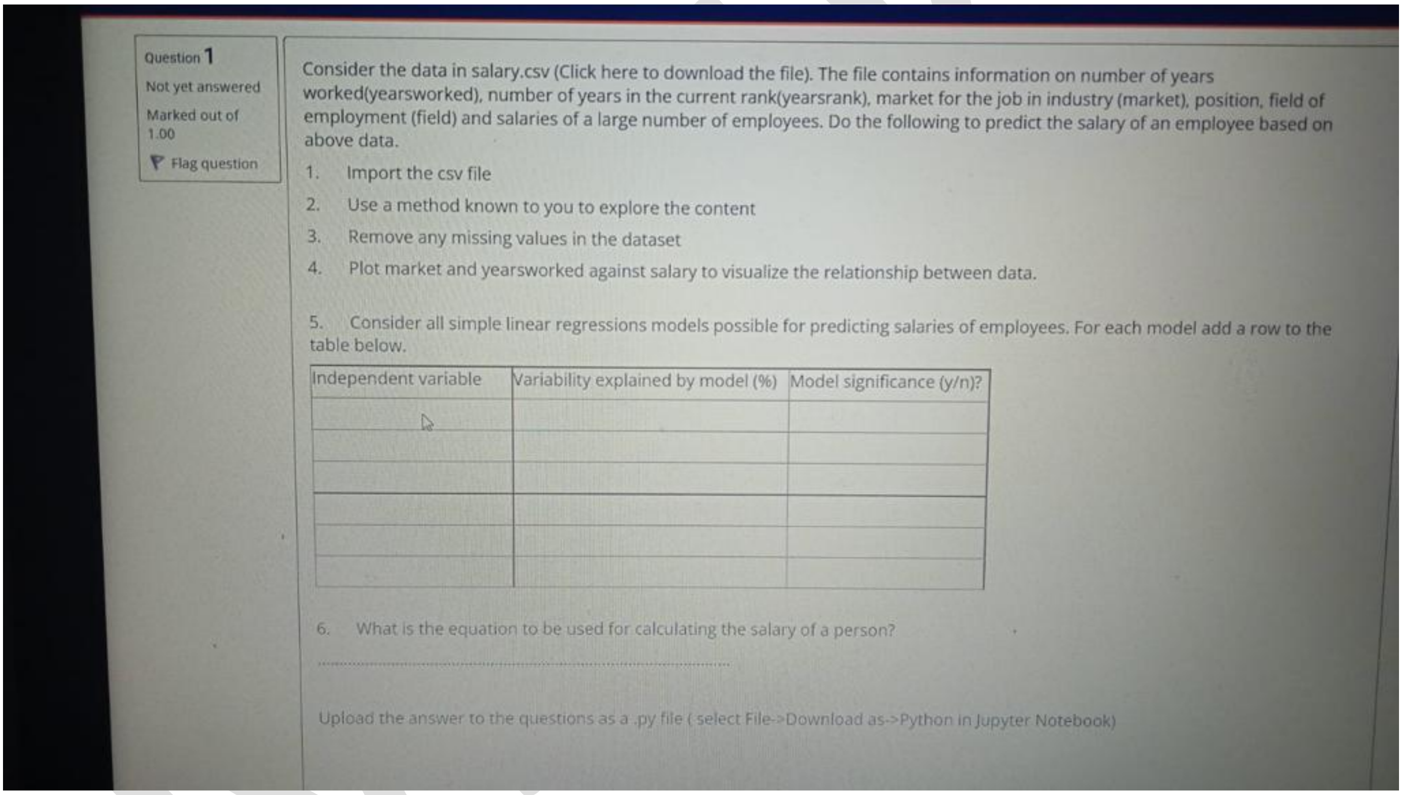
- Income_classifire=DecisionTreeClasfire(random_state=0)

  Income_classifire.fit(X_train,y_train)

- Income_classifire.score(X_test,y_test)

- Income_classifire.predict(X_test[10:14])

  Y_test[10:14]



Consider the data in salary.csv (Click here to download the file). The file contains information on number of years worked(yearsworked), number of years in the current rank(yearsrank), market for the job in industry (market), position, field of employment (field) and salaries of a large number of employees. Do the following to predict the salary of an employee based on above data.

1. Import the csv file
2. Use a method known to you to explore the content
3. Remove any missing values in the dataset
4. Plot market and yearsworked against salary to visualize the relationship between data.

5. Consider all simple linear regressions models possible for predicting salaries of employees. For each model add a row to the table below.

Independent variable	Variability explained by model (%)	Model significance (y/n)?

6. What is the equation to be used for calculating the salary of a person?

....

Upload the answer to the questions as a .py file (select File->Download as->Python in Jupyter Notebook)

First import necessary library's

```
Import pandas as pd
Import numpy as np
From matplotlib import pyplot as plt
Import statsmodels.api as sm
From sklearn.linear_model import LinearRegression
```

1. Df=pd.read_csv(“salary.csv”)
2. Df.head()

   Df.info()

3. Df.dropna(inplace=True)
4. x=df.iloc[: ,<column index range>].values(plot market)

   y=df.iloc[: , [column index]].values (salary)

   Plt.scatter(x,y)

   Model=LinearRegression()

   Model.fit(x,y)

   Const=sm.add_constant(x)

   Model=sm.OLS(y,x).fit()

   Model.summary()

5.

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.957
Model:	OLS	**Adj. R-squared:**	0.955
Method:	Least Squares	**F-statistic:**	622.5
Date:	Fri, 08 Sep 2023	**Prob (F-statistic):**	1.14e-20
Time:	12:09:56	**Log-Likelihood:**	-301.44
No. Observations:	30	**AIC:**	606.9
Df Residuals:	28	**BIC:**	609.7
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	2.579e+04	2273.053	11.347	0.000	2.11e+04	3.04e+04
x1	9449.9623	378.755	24.950	0.000	8674.119	1.02e+04

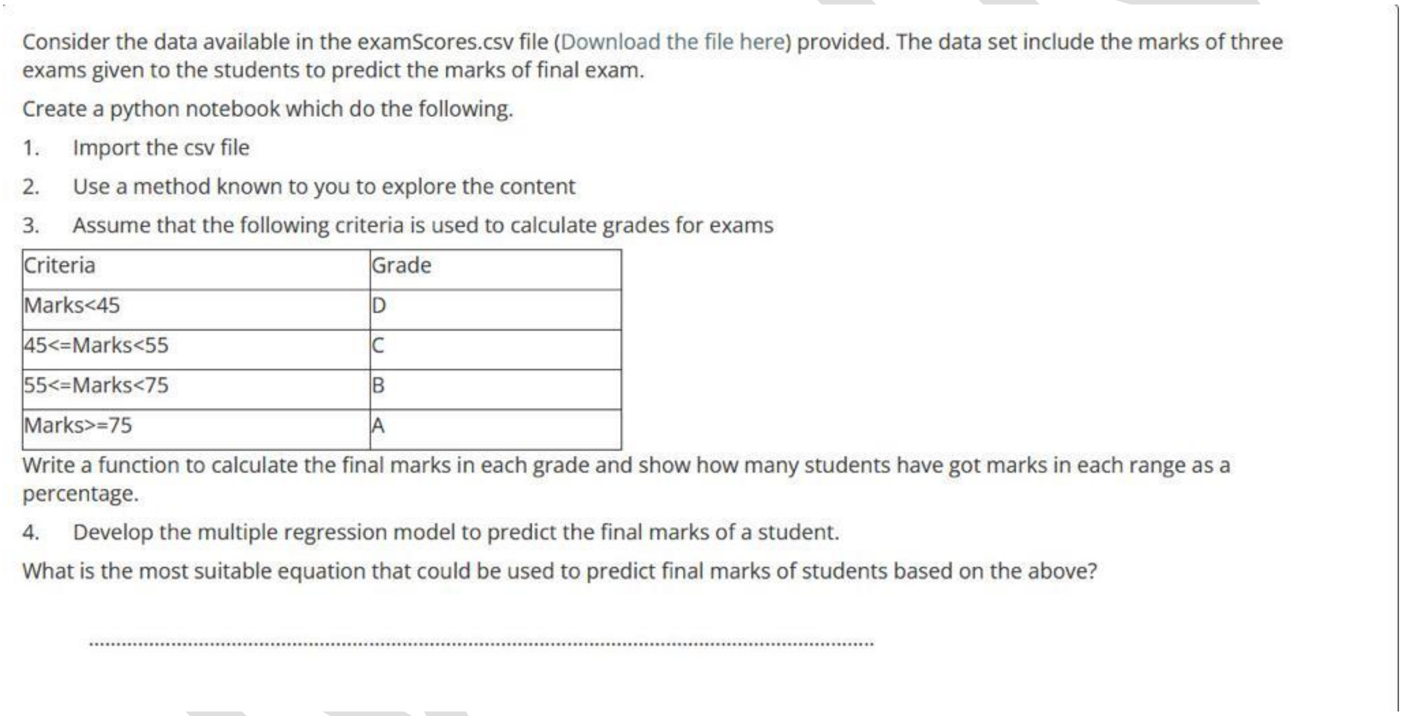
Omnibus:	2.140	**Durbin-Watson:**	1.648
Prob(Omnibus):	0.343	**Jarque-Bera (JB):**	1.569

```
     Skew:    0.363         Prob(JB):    0.456

 Kurtosis:    2.147         Cond. No.      13.2
```

Independent variable	Variability explained by model(%) (R-squared)	Model Significance(Y/N)?
yearworked	0.957	yes

Ptop(f-statistic) value is 1.14e-20=1.14*10^{-20} , this is more closer than to 0 ,therefore model highly significant(Yes)

Other row fill like this



Consider the data available in the examScores.csv file (Download the file here) provided. The data set include the marks of three exams given to the students to predict the marks of final exam.

Create a python notebook which do the following.

1. Import the csv file
2. Use a method known to you to explore the content
3. Assume that the following criteria is used to calculate grades for exams

Criteria	Grade
Marks<45	D
45<=Marks<55	C
55<=Marks<75	B
Marks>=75	A

Write a function to calculate the final marks in each grade and show how many students have got marks in each range as a percentage.

4. Develop the multiple regression model to predict the final marks of a student.

What is the most suitable equation that could be used to predict final marks of students based on the above?

....

3.

```
Def FinalMark (mark):
        If mark < 45:
                Return ‘D’
        If mark >=45 and mark < 55:
                Return ‘C’
        If mark >=55 and mark<75:
                Return ‘B’
```

```
    Else:
        Return ‘A’
Mark=df[‘criteria’].apply(FinalMark)
Mark.value_counts()
F=plt.figure()
Mark.value_counts().plt.pie(autopct=’%1.0f%%’,)
Plt.title(‘Student Mark’)
```