Data Science Project Training Report

on

Machine Learning Domain Projects for Regression, Classification and Clustering using Various Datasets

BACHELOR OF TECHNOLOGY

Session 2021-22

Information Technology

By
Arjit Shandilya
2000321540015

AATIF JAMSHED
ASSISTANT PROFESSOR

DEPARTMENT OF INFORMATION TECHNOLOGY
ABES ENGINEERING COLLEGE, GHAZIABAD

AFFILIATED TO
DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, U.P., LUCKNOW
(Formerly UPTU)

Student's Declaration

I / We hereby declare that the work being presented in this report entitled **"Machine Learning Domain Projects for Regression, Classification and Clustering using Various Datasets"** is an authentic record of my / our own work carried out under the supervision of Dr. /Mr. /Ms. **AATIF JAMSHED, Assistant Professor, Information Technology.**

Date:

Signature of student
Name: ARJIT SHANDILYA
(Roll No.: 2000321540015)
Department: CSE(DS)

This is to certify that the above statement made by the candidate(s) is correct to the best of my knowledge.

Signature of HOD
Dr. Amit Sinha

Information Technology

Signature of Teacher
Aatif Jamshed

Assistant Professor
Information Technology

Date:.....

Table of Contents

Method 1: Regression

1) Introduction:

- Data Taken: BigMart Sales Data(Regression)

Bigmart is a big supermarket chain, with stores all around the country. The management of the shop had set out a challenge to all Data Scientist to help them create a model that can predict the sales per product for each store. The shop has collected sales data of products across 10 stores in different cities over a given period of time.

- Breakdown of the Problem Statement:

  1) This is a supervised machine learning problem with a target label as (Item_Outlet_Sales).
  2) Also since we are expected to predict the sale price for a given product, it becomes a regression task.

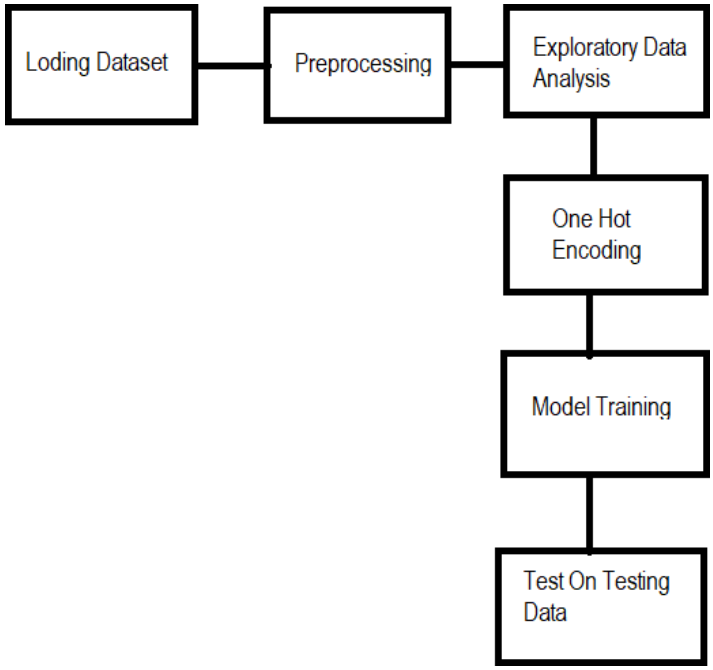
- Regression:

  - Regression is a technique for investigating the relationship between independent variables or features and a dependent variable or outcome. It's used as a method for predictive modelling in machine learning, in which an algorithm is used to predict continuous outcomes.

  - Solving regression problems is one of the most common applications for machine learning models, especially in supervised machine learning. Algorithms are trained to understand the relationship between independent variables and an outcome or dependent variable. The model can then be leveraged to predict the outcome of new and unseen input data, or to fill a gap in missing data.

  - Regression analysis is an integral part of any forecasting or predictive model, so is a common method found in machine learning powered predictive analytics. Alongside classification, regression is a common use for supervised machine learning models. This approach to training models required labelled input and output training data. Machine learning regression models need to understand the relationship between features and outcome variables, so accurately labelled training data is vital.

2) Purpose Methodology:



2.1) Loading Dataset
2.2) Preprocessing
2.3) Exploratory Data Analysis
2.4) One Hot Encoding
2.5) Model Training
2.6) Test On Testing Data

➢ 2.1) Loading the Datasets

Python Command is used to Load the data.
->Import pandas as pd

->Dataset name=”train.csv”

->head(): Used to show First Five Rows

```
In [1]: import pandas as pd
        df=pd.read_csv("train.csv")
        df.head()
```

Out[1]:

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year	Outlet_Size	Outlet_Location
0	FDA15	9.30	Low Fat	0.016047	Dairy	249.8092	OUT049	1999	Medium	
1	DRC01	5.92	Regular	0.019278	Soft Drinks	48.2692	OUT018	2009	Medium	
2	FDN15	17.50	Low Fat	0.016760	Meat	141.6180	OUT049	1999	Medium	
3	FDX07	19.20	Regular	0.000000	Fruits and Vegetables	182.0950	OUT010	1998	NaN	
4	NCD19	8.93	Low Fat	0.000000	Household	53.8614	OUT013	1987	High	

➢ 2.2) Preprocessing

- As we see that Item_weight and Outlet_size has some NULL values so Handle them

- As Item_weight is continous values so replace them with the mean of that column

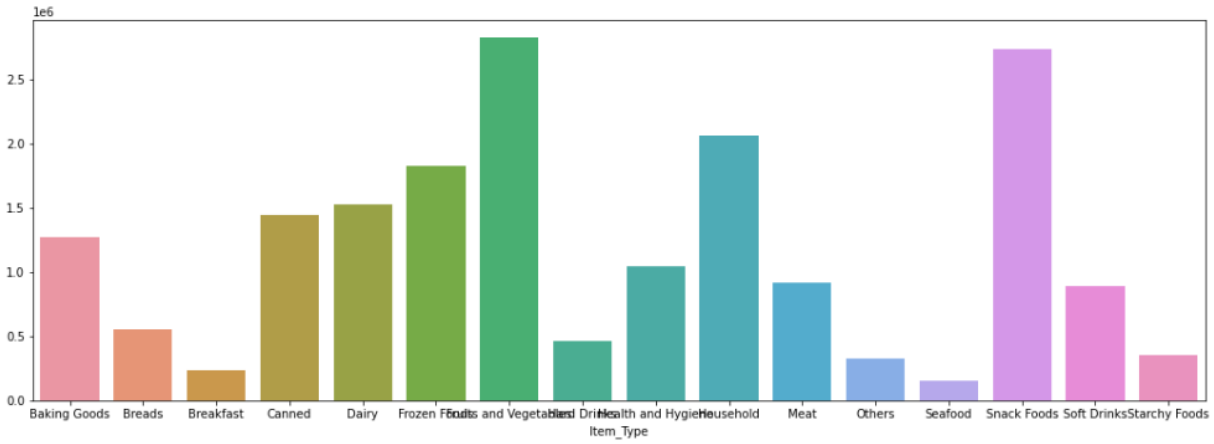
- As Outlet_size has Discreste value so replace them with the mode

```
In [5]: ## As we see that Item_weight and Outlet_size has some NULL values so Handle them

In [6]: ## As Item_weight is continous values so replace them with the mean of that column
        import numpy as np
        df1["Item_Weight"].replace(np.nan,df1["Item_Weight"].mean(),inplace=True)

In [7]: df1["Item_Weight"].isna().sum()

Out[7]: 0

In [8]: ## As Outlet_size has Discreste value so replace them with the mode
        print(df1["Outlet_Size"].mode())
        df1["Outlet_Size"].replace(np.NaN,df1["Outlet_Size"].mode()[0],inplace=True)

        0    Medium
        dtype: object

In [9]: df1["Outlet_Size"].isna().sum()

Out[9]: 0
```

➢ 2.3) Exploratory Data Analysis

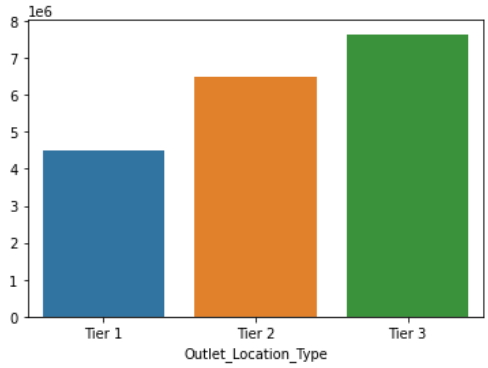
- As we see that the sale of Fruits and Vegitables, snack Foods are maximum

```
In [24]: # As we see that the sale of Fruits and Vegitables, snack Foods are maximum
         plt.figure(figsize=(18,6))
         sns.barplot(x=df_items.index,y=df_items.values)

Out[24]: <AxesSubplot:xlabel='Item_Type'>
```



- Tier 3 has maximum sales

```
In [26]: # Tier 3 has maximum sales
         sns.barplot(x=df_location.index,y=df_location.values)

Out[26]: <AxesSubplot:xlabel='Outlet_Location_Type'>
```



➢ 2.4) One Hot Encoding

- Convert Categorical Values to Numeric Values
- Import OneHotEncoder

```
In [33]: enc=OneHotEncoder()
         enc_data=pd.DataFrame(enc.fit_transform(df3[['Item_Type','Outlet_Establishment_Year','Outlet_Type','Outlet_Location_Type','Outlet
```

```
In [34]: enc_data.head()
```

Out[34]:

	0	1	2	3	4	5	6	7	8	9	...	26	27	28	29	30	31	32	33	34	35
0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	1.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	1.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	...	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0

5 rows × 36 columns

➢ 2.5) Model Training

- Linear Regression

```
In [46]: from sklearn.linear_model import LinearRegression
         model1=LinearRegression()
         model1.fit(x_train,y_train)

Out[46]: LinearRegression()

In [47]: y_pre=model1.predict(x_test)

In [48]: y_pre

Out[48]: array([3416.78066737,  867.37050495, 4003.48764323, ..., 3609.09013417,
                1090.69056717, 2648.84713158])
```

➢ Accuracy:- 55.37

- Lasso Regression

Lasso

```
In [52]: from sklearn.linear_model import Lasso
         l=Lasso()
         l.fit(x,y)

Out[52]: Lasso()

In [64]: l.score(x,y)

Out[64]: 0.5634598910044699
```

➢ Accuracy:- 56.34

- Ridge Model :

Ridge Model

```
In [65]: from sklearn.linear_model import Ridge
         r=Ridge()
         r.fit(x,y)

Out[65]: Ridge()

In [66]: r.score(x,y)

Out[66]: 0.5635861836592834
```

  - Accuracy: 56.35

- ElasticNet Model:

ElasticNet Model

```
In [68]: from sklearn.linear_model import ElasticNet
         e=ElasticNet()
         e.fit(x,y)

Out[68]: ElasticNet()

In [69]: e.score(x,y)

Out[69]: 0.4655244705584345
```

  - Accuracy: 46.55

So, Highest Accuracy is of Ridge model.

➢ 2.6) Predict Price on Testing data

- Predict sales price on Ridge Model.

```
In [68]: # Predict the sale price On Ridge model
         y_predicted_r=r.predict(x_test)
         y_predicted_r

Out[68]: array([2850.68068344, 1599.18461816, 3427.05687563, ..., 1228.83187078,
                4228.77031002, 1731.62441873])
```

3) Result:-

1) We perform Exploratory data analysis on the BigMart Sale Data.
2) We get to Know the Following point:
    - The sale of Fruits and Vegetables, snack Foods are maximum.
    - Tier 3 has maximum sales.
    - In 1985 maximum number of Outlet Establishment.
3) Model Accuracy:
    - Linear Regression: 55.37
    - Lasso: 56.34
    - Ridge: 56.35
    - ElasticNet: 46.55
4) Ridge Regression has Highest Accuracy.

4) Conclusion:

- We train our model on Ridge Model.
- Now we predict value by fit the testing data in it.

```
In [68]: # Predict the sale price On Ridge model
         y_predicted_r=r.predict(x_test)
         y_predicted_r

Out[68]: array([2850.68068344, 1599.18461816, 3427.05687563, ..., 1228.83187078,
                4228.77031002, 1731.62441873])
```

5) Reference(if any)

- https://www.kaggle.com/datasets/brijbhushannanda1979/bigmart-sales-data

Method 2: Classification

1) Introduction:

- Data Taken: Digit Recognizer Data(Regression)

The data files train.csv and test.csv contain gray-scale images of hand-drawn digits, from zero through nine.

Each image is 28 pixels in height and 28 pixels in width, for a total of 784 pixels in total. Each pixel has a single pixel-value associated with it, indicating the lightness or darkness of that pixel, with higher numbers meaning darker. This pixel-value is an integer between 0 and 255, inclusive.

The training data set, (train.csv), has 785 columns. The first column, called "label", is the digit that was drawn by the user. The rest of the columns contain the pixel-values of the associated image.

- Breakdown of the Problem Statement:

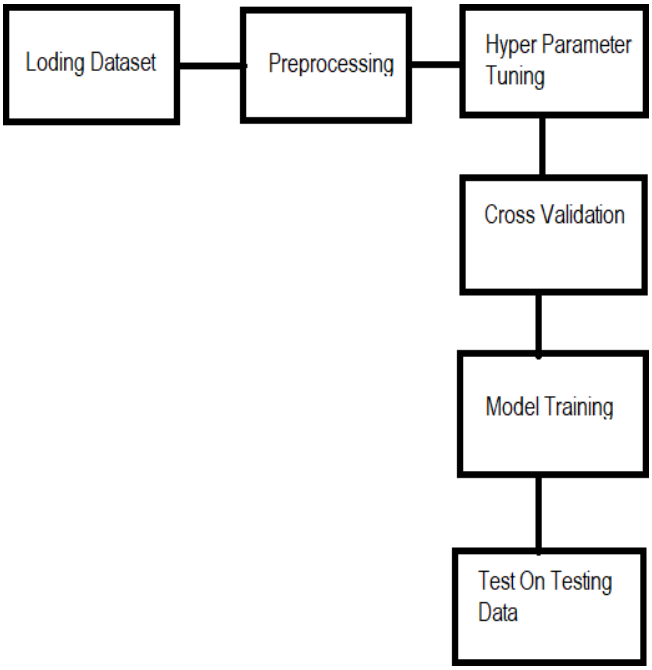
  1) Each pixel column in the training set has a name like pixelx, where x is an integer between 0 and 783, inclusive. To locate this pixel on the image, suppose that we have decomposed x as x = i * 28 + j, where i and j are integers between 0 and 27, inclusive. Then pixelx is located on row i and column j of a 28 x 28 matrix, (indexing by zero).
  2) Your submission file should be in the following format: For each of the 28000 images in the test set, output a single line containing the ImageId and the digit you predict. For example, if you predict that the first image is of a 3, the second image is of a 7, and the third image is of a 8, then your submission file would look like:

- Classification:

  - In machine learning, classification is a supervised learning concept which basically categorizes a set of data into classes. The most common classification problems are – speech recognition, face detection, handwriting recognition, document classification, etc.

  - It can be either a binary classification problem or a multi-class problem too. There are a bunch of machine learning algorithms for classification in machine learning. Let us take a look at those classification algorithms in machine learning.

2) Purpose Methodology:



- Loading Dataset
- Preprocessing
- Hyper parameter Tuning
- Cross validation
- Model training
- Test on testing data

➢ 2.1) Loading the Datasets

Python Command is used to Load the data.
->Import pandas as pd

->Dataset name=”train.csv”

->head(): Used to show First Five Rows

Load the dataset

```
In [19]: import pandas as pd

In [20]: import seaborn as sns
         import matplotlib.pyplot as plt
         %matplotlib inline

In [21]: df=pd.read_csv("train.csv")

In [22]: df.head()
```

Out[22]:

	label	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	...	pixel774	pixel775	pixel776	pixel777	pixel778	pixel779	pixel780	pixel781	pi
0	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
2	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
3	4	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	

5 rows × 785 columns

➢ 2.2) Preprocessing

- Droping NULL Values From the Dataset

Preprocessing

```
In [23]: (df.isnull().sum()==0).count()
Out[23]: 785

In [24]: df.shape
Out[24]: (42000, 785)

In [25]: x=df.drop("label",axis="columns")

In [26]: y=df["label"]
```

➢ 2.3) Hyper Parameter Tuning

- It is used to select the best perameters to train the model.
- We use GridSearchCV

In [28]:
```
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import ShuffleSplit
def find_best_model(x,y):
    model_params={
        'LogisticRegression':{
            'model':LogisticRegression(),
            'params':{
                'C': [1,5,10],
                'penalty': ['l1','l2']
            }
        },
        'DecisionTreeClassifier':{
            'model':DecisionTreeClassifier(),
            'params':{
                'criterion':['gini','entropy']
            }
        },
        'SVC':{
            'model':SVC(),
            'params':{
                'C':[1,2,3],
                'kernel':['rbf','linear']
            }
        },
```

➢ 2.4) Cross validation

- It is used to increase the performance of the training model.
- It will select the different samples for training as well as testing.

Cross validation

In [ ]:
```
cross_val_score(LogisticRegression(),x,y)
```

In [ ]:
```
cross_val_score(DecisionTreeClassifier(),x,y)
```

In [ ]:
```
cross_val_score(GaussianNB(),x,y)
```

In [ ]:
```
cross_val_score(SVC(),x,y)
```

2.5) Model Training

- As SCV gives best result so train the model on SVC.
- And save that model in pickle.

Model Training

```
In [ ]: model=SVC()
        model.fit(x,y)
```

```
In [ ]: import pickle
        with open("Digits_model.pickle","wb") as f:
            pickle.dump(model,f)
```

➢ 2.6) Test Model on Testing data

- Test Model on Testing data.
- Save this result in .csv format.

```
In [7]: y_predict=model.predict(df)
```

```
In [25]: y_predict.shape
Out[25]: (28000,)
```

6) Result:-

1) We perform classification on Digit Recognizer Dataset.
2) Model Accuracy:
   - LogisticRegression: 91.33
   - DecisionTreeClassifier: 85.55
   - GaussianNB: 55.96
   - SVC: 94.65
3) SVC has Highest Accuracy.

4) Conclusion:

- We train our model on SVC.
- Now we predict value by fit the testing data in it.

Model Training

```
In [ ]: model=SVC()
        model.fit(x,y)
```

```
In [ ]: import pickle
        with open("Digits_model.pickle","wb") as f:
            pickle.dump(model,f)
```

5) Reference(if any)

- https://www.kaggle.com/competitions/digit-recognizer/overview

Method 3 : Clustering

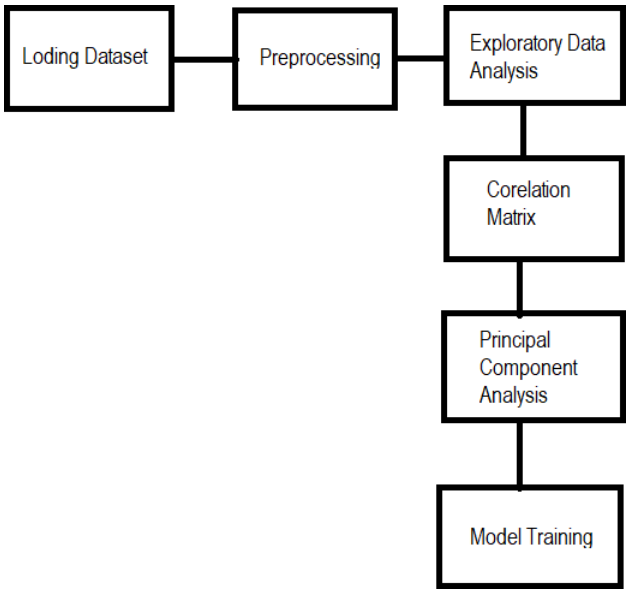
1) Introduction:

- Data Taken: Turkiye Student Evaluation Data Set (Clustering)
- Attributes:

instr: Instructor's identifier; values taken from {1,2,3}
class: Course code (descriptor); values taken from {1-13}
repeat: Number of times the student is taking this course; values taken from {0,1,2,3,...}
attendance: Code of the level of attendance; values from {0, 1, 2, 3, 4}
difficulty: Level of difficulty of the course as perceived by the student; values taken from {1,2,3,4,5}
Q1: The semester course content, teaching method and evaluation system were provided at the start.
Q2: The course aims and objectives were clearly stated at the beginning of the period.
Q3: The course was worth the amount of credit assigned to it.
Q4: The course was taught according to the syllabus announced on the first day of class.
Q5: The class discussions, homework assignments, applications and studies were satisfactory.
Q6: The textbook and other courses resources were sufficient and up to date.
Q7: The course allowed field work, applications, laboratory, discussion and other studies.
Q8: The quizzes, assignments, projects and exams contributed to helping the learning.
Q9: I greatly enjoyed the class and was eager to actively participate during the lectures.
.
.
.
.

- Clustering:
  - Clustering or cluster analysis is a machine learning technique, which groups the unlabelled dataset. It can be defined as "A way of grouping the data points into different clusters, consisting of similar data points.
  - The objects with the possible similarities remain in a group that has less or no similarities with another group.

2) Purpose Methodology:



- Loading Dataset
- Preprocessing
- Exploratory Data Analysis
- Co-relation Matrix
- Principal Component Analysis
- Model Building

➢ 2.1) Loading data set

Python Command is used to Load the data.
Import pandas as pd
Dataset name=" turkiye-student-evaluation_generic.csv"
head(): Used to show First Five Rows

Loading the dataset

In [1]:
```
import pandas as pd
df=pd.read_csv("turkiye-student-evaluation_generic.csv")
df.head()
```

Out[1]:

	instr	class	nb.repeat	attendance	difficulty	Q1	Q2	Q3	Q4	Q5	...	Q19	Q20	Q21	Q22	Q23	Q24	Q25	Q26	Q27	Q28
0	1	2	1	0	4	3	3	3	3	3	...	3	3	3	3	3	3	3	3	3	3
1	1	2	1	1	3	3	3	3	3	3	...	3	3	3	3	3	3	3	3	3	3
2	1	2	1	2	4	5	5	5	5	5	...	5	5	5	5	5	5	5	5	5	5
3	1	2	1	1	3	3	3	3	3	3	...	3	3	3	3	3	3	3	3	3	3
4	1	2	1	0	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1

5 rows × 33 columns

➢ 2.2) Preprocessing

- There is no NULL Value in the dataset.
- And all the values are in integer type.
- So there is no need for preprocessing.

In [6]:
```
df.describe()
```

Out[6]:

	instr	class	nb.repeat	attendance	difficulty	Q1	Q2	Q3	Q4	
count	5820.000000	5820.000000	5820.000000	5820.000000	5820.000000	5820.000000	5820.000000	5820.000000	5820.000000	5820.0
mean	2.485567	7.276289	1.214089	1.675601	2.783505	2.929897	3.073883	3.178694	3.082474	3.1
std	0.718473	3.688175	0.532376	1.474975	1.348987	1.341077	1.285251	1.253567	1.284594	1.2
min	1.000000	1.000000	1.000000	0.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.0
25%	2.000000	4.000000	1.000000	0.000000	1.000000	2.000000	2.000000	2.000000	2.000000	2.0
50%	3.000000	7.000000	1.000000	1.000000	3.000000	3.000000	3.000000	3.000000	3.000000	3.0
75%	3.000000	10.000000	1.000000	3.000000	4.000000	4.000000	4.000000	4.000000	4.000000	4.0
max	3.000000	13.000000	3.000000	4.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.0

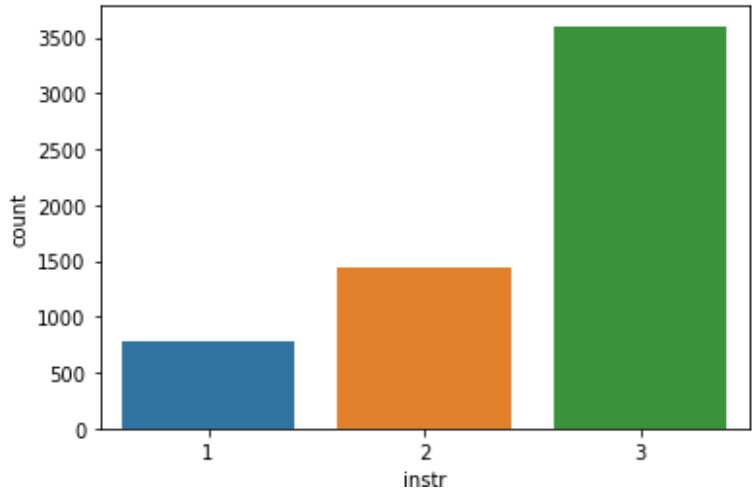
➢ 2.3) Exploratory Data Analysis

- Instruction three has taken more courses

In [14]:
```
sns.countplot(df['instr'])
## Instruction three has taken more courses
```

```
C:\Users\ADITYA\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
12, the only valid positional argument will be `data`, and passing other
ation.
  warnings.warn(
```

Out[14]:
```
<AxesSubplot:xlabel='instr', ylabel='count'>
```

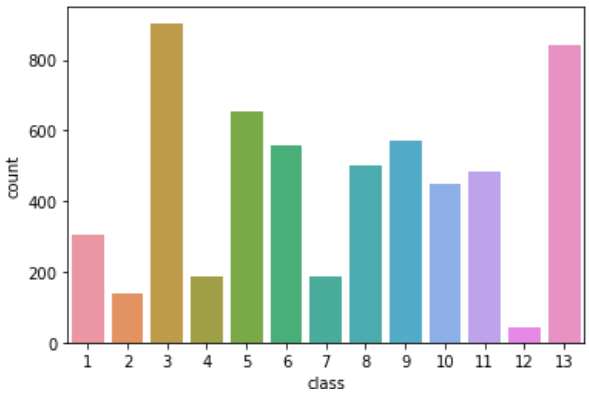


- Maximum number of class

In [19]:
```
sns.countplot(df["class"])
```

```
C:\Users\ADITYA\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futur
12, the only valid positional argument will be `data`, and passing other arg
ation.
  warnings.warn(
```

Out[19]:
```
<AxesSubplot:xlabel='class', ylabel='count'>
```



➢ 2.4) Co-relation Matrix

- Use to Find Co-relation between columns.
- Get useful columns from the data.

```
In [31]:
corr=df.corr()
plt.figure(figsize=(18,18))
sns.heatmap(corr,annot=True,cmap="coolwarm")
```

```
Out[31]: <AxesSubplot:>
```

➢ 2.5) Principal of Component Analysis

- To reduce the dimension of the data.

```
In [79]:
X=df.iloc[:,5:33]
from sklearn.decomposition import PCA
pca=PCA(n_components=2,random_state=42)
X_pca=pca.fit_transform(X)
```

```
In [80]:
X_pca
# So now dimensions are reduced to Two
```

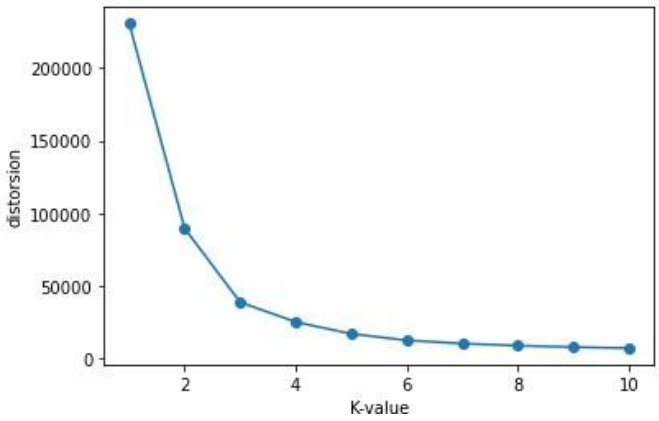
```
Out[80]: array([[ 0.98901533,  0.52279815],
                [ 0.98901533,  0.52279815],
                [-9.59128851,  0.6408021 ],
                ...,
                [-9.59128851,  0.6408021 ],
                [11.56931918,  0.40479421],
                [11.56931918,  0.40479421]])
```

➢ 2.6) Model Building

- Making elbow to find the best value of K

In [85]:
```
from sklearn.cluster import KMeans
distortions=[]
for i in range(1,11):
    km=KMeans(n_clusters=i,init='k-means++',n_init=10,max_iter=300,random_state=0)
    km.fit(X_pca)
    distortions.append(km.inertia_)
plt.plot(range(1,11),distortions,marker='o')
plt.xlabel("K-value")
plt.ylabel("distorsion")
```

Out[85]:
```
Text(0, 0.5, 'distorsion')
```



- Train the model k=3

In [87]:
```
model=KMeans(n_clusters=3,init='k-means++',n_init=10,max_iter=300,random_state=0)
model.fit(X_pca)
```

Out[87]:
```
KMeans(n_clusters=3, random_state=0)
```

In [88]:
```
y=model.predict(X_pca)
```

In [89]:
```
y
```

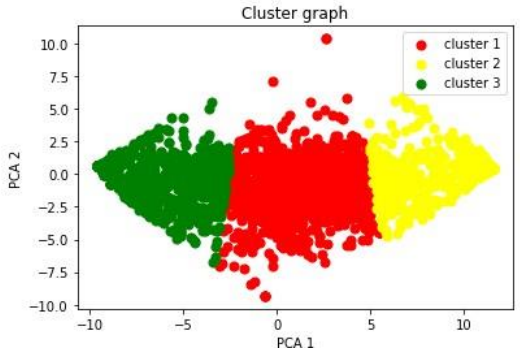
Out[89]:
```
array([0, 0, 2, ..., 2, 1, 1])
```

- Plot Graph to show cluster

Plot the Graph to show the Clusters

In [95]:
```
plt.scatter(X_pca[y==0,0],X_pca[y==0,1],s=50,c="red",label="cluster 1")
plt.scatter(X_pca[y==1,0],X_pca[y==1,1],s=50,c="yellow",label="cluster 2")
plt.scatter(X_pca[y==2,0],X_pca[y==2,1],s=50,c="green",label="cluster 3")
plt.title("Cluster graph")
plt.xlabel("PCA 1")
plt.ylabel("PCA 2")
plt.legend()
```

Out[95]:
```
<matplotlib.legend.Legend at 0x23bdef40e50>
```

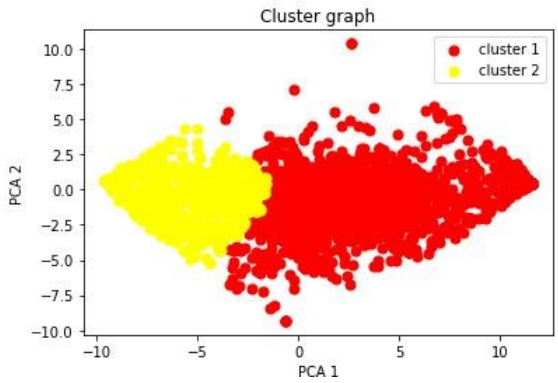


- Agglomerative clustering

In [102…
```
from sklearn.cluster import AgglomerativeClustering
model=AgglomerativeClustering(n_clusters=2,affinity='euclidean',linkage='ward')
y=model.fit_predict(X_pca)
```

In [103…
```
plt.scatter(X_pca[y==0,0],X_pca[y==0,1],s=50,c="red",label="cluster 1")
plt.scatter(X_pca[y==1,0],X_pca[y==1,1],s=50,c="yellow",label="cluster 2")
plt.title("Cluster graph")
plt.xlabel("PCA 1")
plt.ylabel("PCA 2")
plt.legend()
```

Out[103…
```
<matplotlib.legend.Legend at 0x23be1c8b3a0>
```

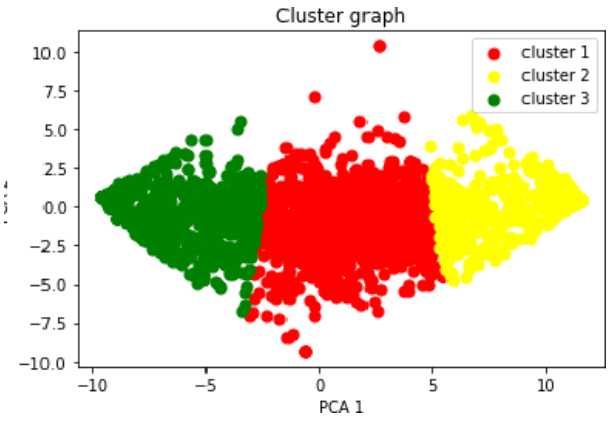


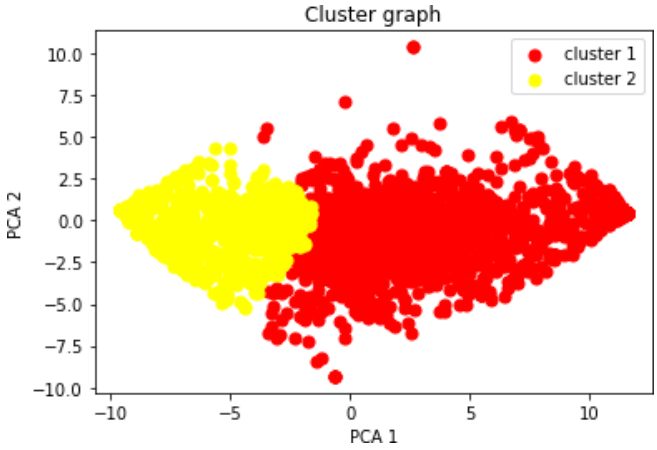
3) Result:-

1) We perform classification on Turkiye Student Evaluation Data.
2) Model:
    - KMeans clustering: k=3
    - Agglomerative clustering: k=2
3) Both the Model are predicting and showing great result.

4) Conclusion:

- We train our model on KMeans and Agglomerative.
- Now we divide the data into clusters.





5) Reference(if any):

- http://archive.ics.uci.edu/ml/datasets/turkiye+student+evaluation