Practical 1	
Aim: To perform data Pre-processing task and demonstrate Classification algorithm of K nearest neighbour for the given dataset.	
Name: Labhesh Joshi	Roll No: KCTBCS030
Performance date: 30-06-2022	Sign:

Theory:

Supervised learning is the type of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output. Where, the labelled data means some input data is already tagged with the correct output.

In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher.

Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y).

How Supervised Learning Works?

In supervised learning, models are trained using labelled dataset, where the model learns about each type of data. Once the training process is completed, the model is tested on the basis of test data (a subset of the training set), and then it predicts the output.

Steps Involved in Supervised Learning:

- First Determine the type of training dataset
- Collect/Gather the labelled training data.
- Split the training dataset into training dataset, test dataset, and validation dataset.
- Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
- Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.
- Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.
- Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate.

Types of Supervised learning:

Supervised learning can be separated into two types of problems when data mining—classification and regression:

- Classification uses an algorithm to accurately assign test data into specific categories. It recognizes specific entities within the dataset and attempts to draw some conclusions on how those entities should be labeled or defined. Common classification algorithms are linear classifiers, support vector machines (SVM), decision trees, k-nearest neighbor, and random forest, which are described in more detail below.
- Regression is used to understand the relationship between dependent and independent variables. It is commonly used to make projections, such as for sales revenue for a given business. Linear regression, logistical regression, and polynomial regression are popular regression algorithms.

Advantages of Supervised learning:

- With the help of supervised learning, the model can predict the output on the basis of prior experiences.
- In supervised learning, we can have an exact idea about the classes of objects.
- Supervised learning model helps us to solve various real-world problems such as **fraud detection, spam filtering**, etc.

Disadvantages of supervised learning:

- Supervised learning models are not suitable for handling the complex tasks.
- Supervised learning cannot predict the correct output if the test data is different from the training dataset.
- Training required lots of computation times.
- In supervised learning, we need enough knowledge about the classes of object.

KNN algorithm.

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- /8K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

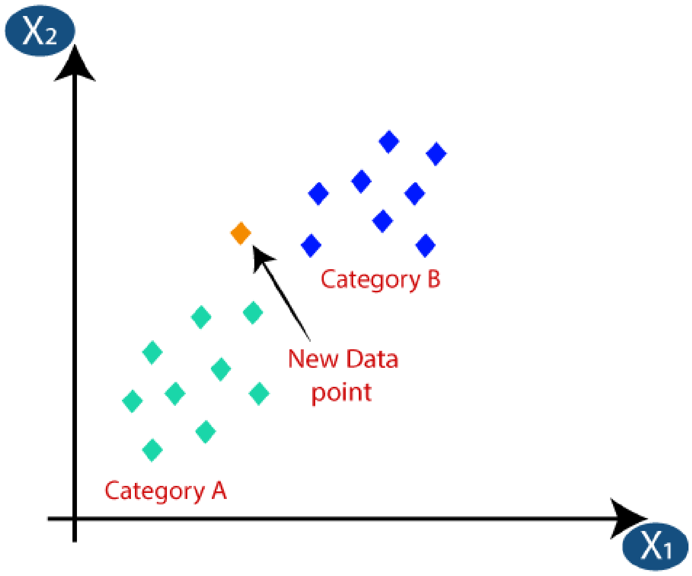
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
- **Example:** Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

How does K-NN work?

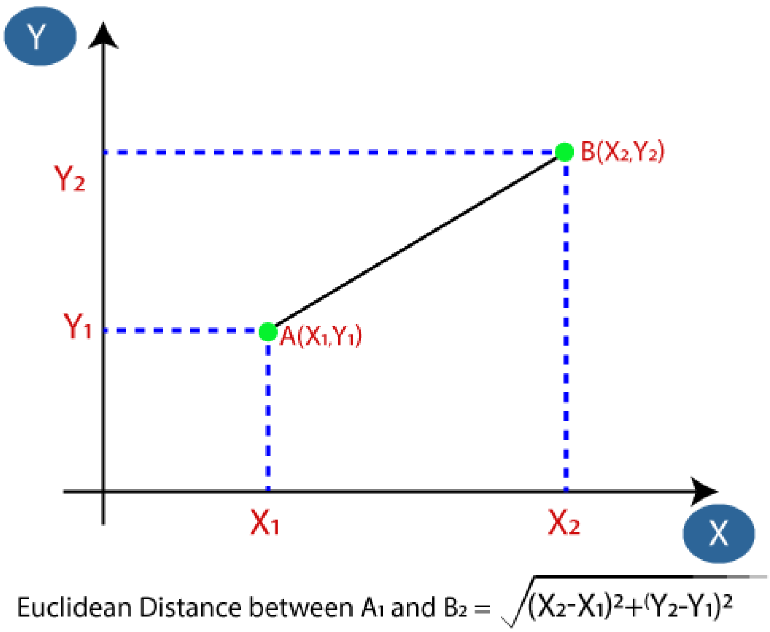
The K-NN working can be explained on the basis of the below algorithm:

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

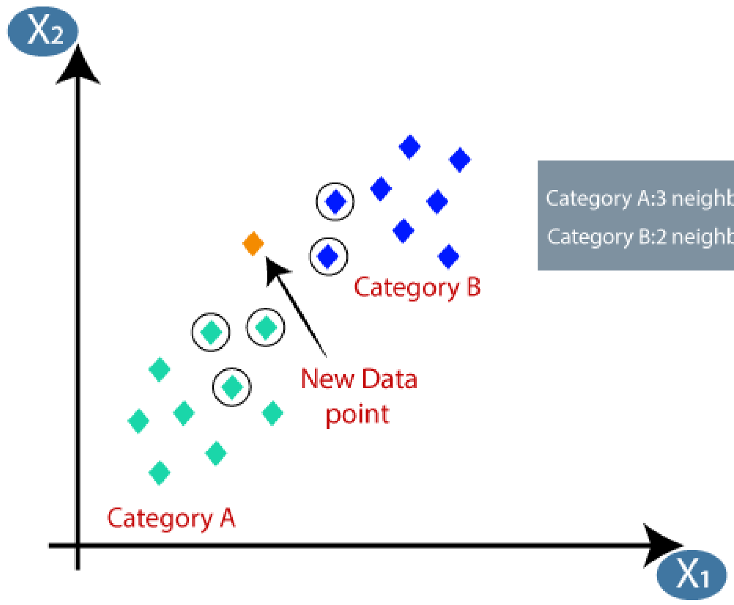
Suppose we have a new data point and we need to put it in the required category. Consider the below image:



- Firstly, we will choose the number of neighbors, so we will choose the k=5.
- Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



- By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:



- As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

How to select the value of K in the K-NN Algorithm?

Below are some points to remember while selecting the value of K in the K-NN algorithm:

- There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
- A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
- Large values for K are good, but it may find some difficulties.

3. Advantages and Disadvantages of KNN.

Advantages of KNN Algorithm:

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

Disadvantages of KNN Algorithm:

- We always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.
- Might not be as accurate as some other learning algorithms.

<u>Code:</u>

***prac1_a**

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
#Importing the dataset
dataset = pd.read_csv("Social_Network_Ads.csv")
x = dataset.iloc[:,[2,3]].values
y = dataset.iloc[:,-1].values
dataset.head()
dataset.describe()
dataset.info()
dataset.isnull().sum()
#Splitting the dataset into training and testing
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x,y,test_size=0.20, random_state = 90)
print("Size of x-training data: ", xtrain.shape)
print("Size of y-training data: ", ytrain.shape)
print("Size of x-test data: ", xtest.shape)
print("Size of y-test data: ", ytest.shape)
#Feature scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
xtrain = sc.fit_transform(xtrain)
xtest = sc.transform(xtest)
#Training the knn model on the training set
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 7, metric = "minkowski", p=2)
classifier.fit(xtrain,ytrain)

#Predict the test set result
```

```
ypred = classifier.predict(xtest)
#Making confusion matrix
from sklearn.metrics import confusion_matrix , accuracy_score
cm = confusion_matrix(ytest, ypred)
ac = accuracy_score(ytest, ypred)
print("\nConfusion Matrix: \n",cm)
print("Accuracy of the model: ",ac)
#plotting elbow method graph
neighbors = np.arange(1,9)
train_accuracy = np.empty(len(neighbors))
test_accuracy = np.empty(len(neighbors))
for i,k in enumerate(neighbors):
    knn = KNeighborsClassifier(n_neighbors = k)
    knn.fit(xtrain, ytrain)
    train_accuracy[i] = knn.score(xtrain, ytrain)
    test_accuracy[i] = knn.score(xtest, ytest)
plt.plot(neighbors, train_accuracy, label="Train Accuracy")
plt.plot(neighbors, test_accuracy, label="Test Accuracy")
plt.legend()
plt.xlabel("n_neighbors")
plt.ylabel("Accuracy")
plt.show()
```

Output:

```
======================== RESTART: C:\SEM5\DWM\prac1_a.py
========================
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   User ID          400 non-null    int64
```

1 Gender 400 non-null object

2 Age 400 non-null int64

3 EstimatedSalary 400 non-null int64

4 Purchased 400 non-null int64

dtypes: int64(4), object(1)

memory usage: 15.8+ KB

Size of x-training data: (320, 2)

Size of y-training data: (320,)

Size of x-test data: (80, 2)

Size of y-test data: (80,)

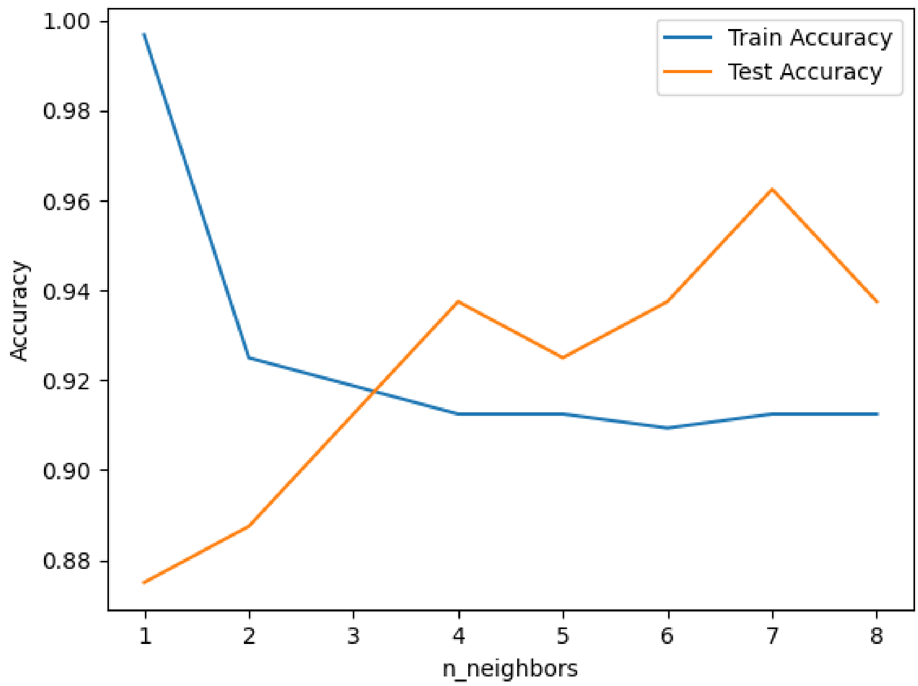
Confusion Matrix:

[[52 2]

[1 25]]

Accuracy of the model: 0.9625

Graph:



Code:

***prac1_b**

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
dataset = pd.read_csv("BankNote_Authentication.csv")
x=dataset.iloc[:,[0,1,2,3]].values
y=dataset.iloc[:,-1].values
print(dataset.count)
from sklearn.model_selection import train_test_split
xtrain, xtest,ytrain,ytest = train_test_split(x,y,test_size = 0.20,random_state=40)

from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
xtrain = sc.fit_transform(xtrain)
xtest = sc.transform(xtest)

from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 3, metric = "minkowski", p=2)
classifier.fit(xtrain,ytrain)

ypred = classifier.predict(xtest)

from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(ytest, ypred)
ac = accuracy_score(ytest, ypred)
print("Confusion Matrix: ",cm)
print("Accuracy Score: ", ac)
```

```
#plotting elbow method graph
neighbors = np.arange(1,20)
train_accuracy = np.empty(len(neighbors))
test_accuracy = np.empty(len(neighbors))

for i,k in enumerate(neighbors):
    knn = KNeighborsClassifier(n_neighbors = k)
    knn.fit(xtrain, ytrain)
    train_accuracy[i] = knn.score(xtrain, ytrain)
    test_accuracy[i] = knn.score(xtest, ytest)

plt.plot(neighbors, train_accuracy, label="Train Accuracy")
plt.plot(neighbors, test_accuracy, label="Test Accuracy")
plt.legend()
plt.xlabel("n_neighbors")
plt.ylabel("Accuracy")
plt.show()
```

Output:

```
======================== RESTART: C:/SEM5/DWM/prac1_b.py
========================
<bound method DataFrame.count of
variance  skewness  curtosis  entropy  class
0     3.62160   8.66610   -2.8073 -0.44699      0
1     4.54590   8.16740   -2.4586 -1.46210      0
2     3.86600  -2.63830    1.9242  0.10645      0
3     3.45660   9.52280   -4.0112 -3.59440      0
4     0.32924  -4.45520    4.5718 -0.98880      0
...       ...       ...       ...      ...    ...
1367   0.40614   1.34920   -1.4501 -0.55949      1
1368  -1.38870  -4.87730    6.4774  0.34179      1
```

1369 -3.75030 -13.45860 17.5932 -2.77710 1

1370 -3.56370 -8.38270 12.3930 -1.28230 1

1371 -2.54190 -0.65804 2.6842 1.19520 1

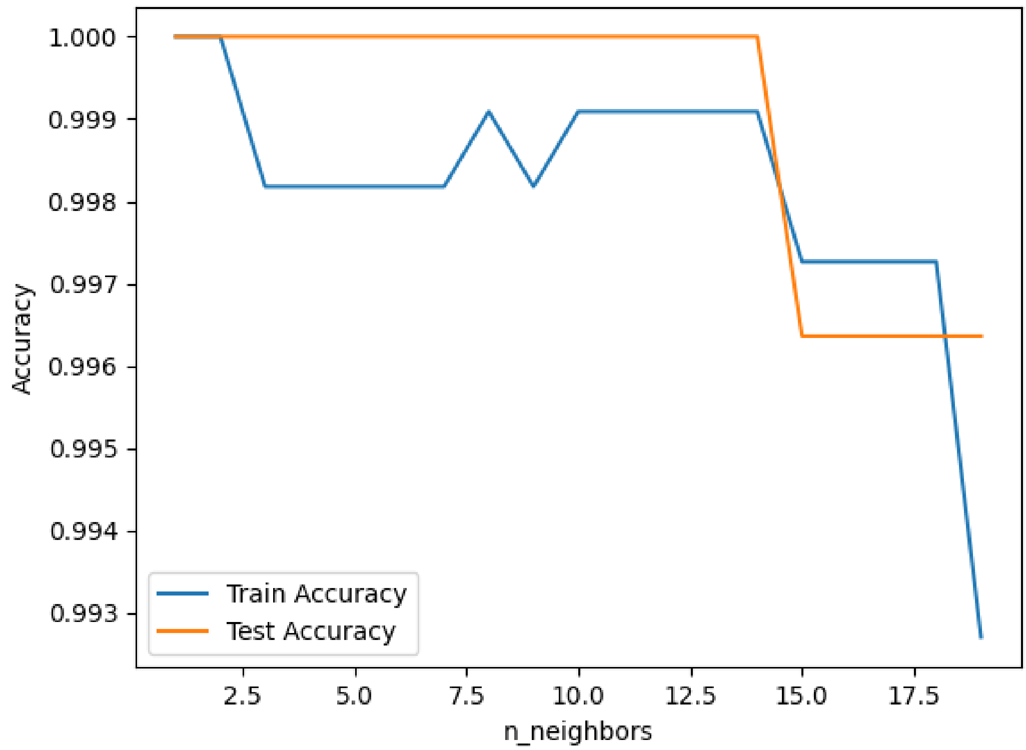
[1372 rows x 5 columns]>

Confusion Matrix: [[147 0]

[0 128]]

Accuracy Score: 1.0

Graph:



Practical 2	
Aim: To Demonstrate Classification algorithm of Decision Tree on the given Dataset.	
Name: Labhesh Joshi	Roll No: KCTBCS030
Performance date: 05-07-2022	Sign:

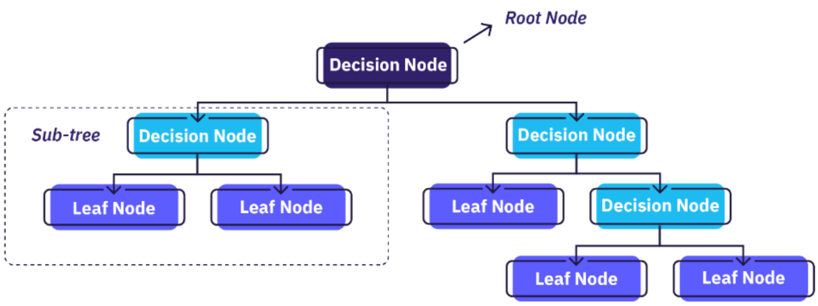
Theory

Classification

Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. Decision trees can handle both categorical and numerical data.

Decision Tree Terminologies

- **Root Node:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
- **Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
- **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
- **Branch/Sub Tree:** A tree formed by splitting the tree.
- **Pruning:** Pruning is the process of removing the unwanted branches from the tree.
- **Parent/Child node:** The root node of the tree is called the parent node, and other nodes are called the child nodes.



2. Algorithm along with formula

The core algorithm for building decision trees called **ID3** by J. R. Quinlan which employs a top-down, greedy search through the space of possible branches with no backtracking

ID3 uses Entropy and Information Gain to construct a decision tree.

The ID3 algorithm follows the below workflow in order to build a Decision Tree:

1. Calculate entropy for dataset.
2. For each attribute/feature.
   2.1. Calculate entropy for all its categorical values.
   2.2. Calculate information gain for the feature.
3. Find the feature with maximum information gain.
4. Repeat it until we get the desired tree.

Two measures are used to decide the best attribute:

1. Information Gain
2. Entropy

Entropy measures the impurity or uncertainty present in the data. It is used to decide how a Decision Tree can split the data.

Equation For Entropy:

$$Entropy = -\Sigma p(x) \log p(x)$$

Information Gain (IG) is the most significant measure used to build a Decision Tree. It indicates how much "information" a particular feature/ variable gives us about the final outcome.

Information Gain is important because it used to choose the variable that best splits the data at each node of a Decision Tree. The variable with the highest IG is used to split the data at the root node.

Equation For Information Gain (IG):

$$Information\ Gain = entropy(parent) - [weighted\ average] * entropy(children)$$

3. Advantages and Disadvantages

Advantages	Disadvantage
1. Compared to other algorithms decision trees requires less effort for data preparation during pre-processing.	1. A small change in the data can cause a large change in the structure of the decision tree causing instability.
2. A decision tree does not require normalization of data.	2. Decision tree often involves higher time to train the model.
3. A decision tree does not require scaling of data as well.	3. For a Decision tree sometimes calculation can go far more complex compared to other algorithms.

4. Missing values in the data also do NOT affect the process of building a decision tree to any considerable extent.	4. The Decision Tree algorithm is inadequate for applying regression and predicting continuous values.
5. A Decision tree model is very intuitive and easy to explain to technical teams as well as stakeholders.	5. Decision tree training is relatively expensive as the complexity and time has taken are more.

Code:

***prac2_a**

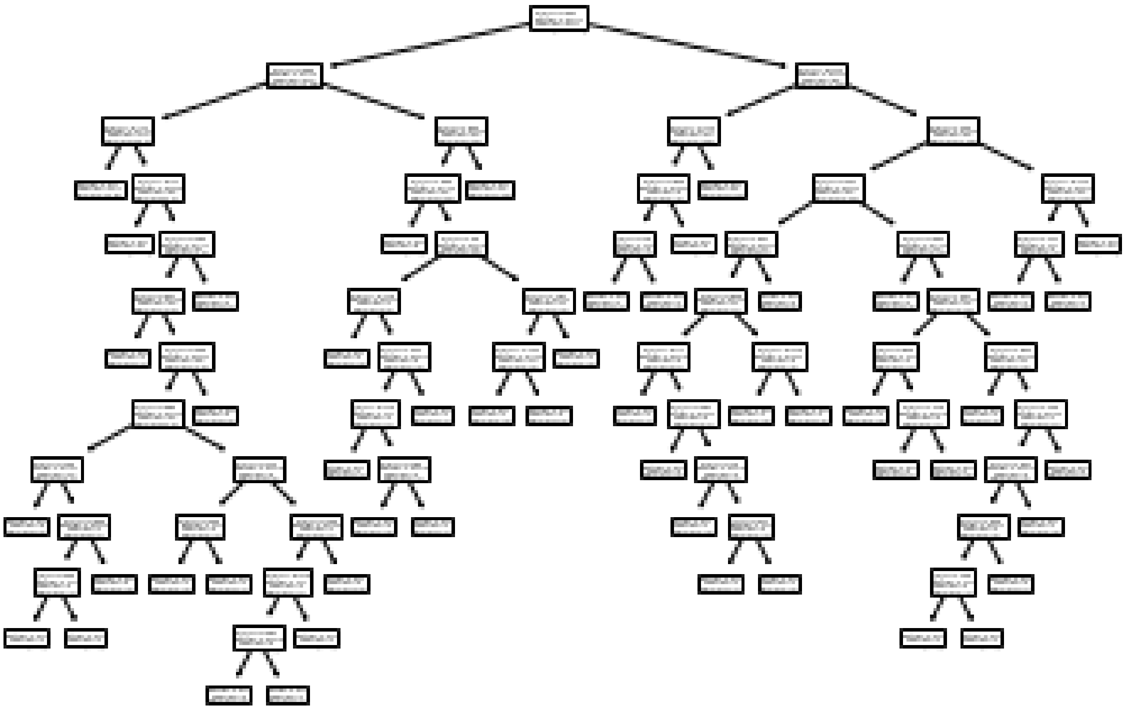
```
import pandas as pd
#importing datasets

dataset= pd.read_csv('./Social_Network_Ads.csv')
#Extracting Independent and dependent Variable
x= dataset.iloc[:, [2,3]].values
y= dataset.iloc[:, 4].values
dataset.count

# Splitting the dataset into training and test set.
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.25, random_state=0)

#feature Scaling
from sklearn.preprocessing import StandardScaler
st_x= StandardScaler()
x_train= st_x.fit_transform(x_train)
x_test= st_x.transform(x_test)

#Fitting Decision Tree classifier to the training set
from sklearn.tree import DecisionTreeClassifier
classifier= DecisionTreeClassifier(criterion='entropy', random_state=0)
classifier.fit(x_train, y_train)



#Predicting the test set result
y_pred= classifier.predict(x_test)
```

```
#Creating the Confusion matrix

from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn import tree
cm = confusion_matrix(y_test, y_pred)
ac = accuracy_score(y_test, y_pred)
print ("Confusion Matrix : \n", cm)
print ("Accuracy : ", ac)

tree.plot_tree(classifier)
```

Output:

```
======================== RESTART: C:/SEM5/DWM/prac2.py
========================
Confusion Matrix :
 [[62  6]
 [ 3 29]]
Accuracy :  0.91
```

Graph:



<u>Code:</u>

***prac2_b**

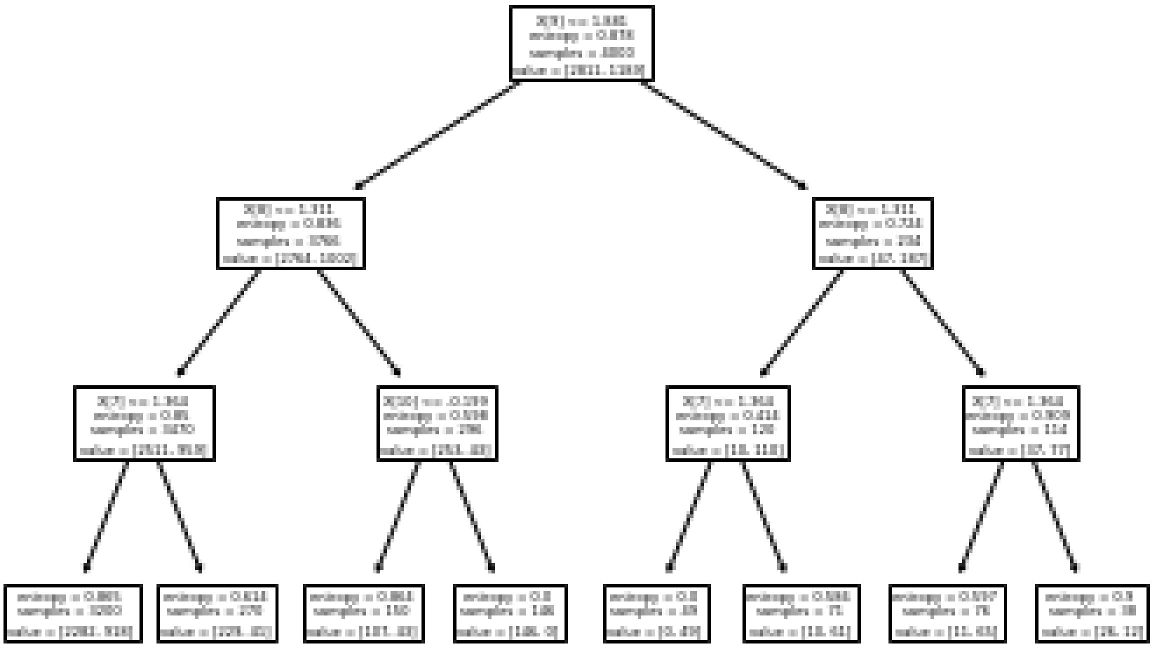
```
import pandas as pd
dataset = pd.read_csv("E:/SEM-5/Data Warehousing and Mining/UniversalBank.csv")
x=dataset.iloc[:,[1,2,3,5,6,7,8,9,10,11,12]].values
y=dataset.iloc[:,-1].values
print(dataset.count)

from sklearn.model_selection import train_test_split
xtrain, xtest,ytrain,ytest = train_test_split(x,y,test_size = 0.20, random_state=40)

from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
xtrain = sc.fit_transform(xtrain)
xtest = sc.transform(xtest)

from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion='entropy', random_state=40,max_depth=3)
classifier.fit(xtrain, ytrain)

ypred = classifier.predict(xtest)

from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn import tree
cm = confusion_matrix(ytest, ypred)
```

```
ac = accuracy_score(ytest, ypred)
print("Confusion Matrix: ",cm)
print("Accuracy Score: ", ac)

tree.plot_tree(classifier)
```

Output:

```
======================== RESTART: C:/SEM5/DWM/prac2_b.py
=======================

<bound method DataFrame.count of

ID  Age  Experience  ...  CD Account  Online  CreditCard

0       1   25          1  ...          0       0          0

1       2   45         19  ...          0       0          0

2       3   39         15  ...          0       0          0

3       4   35          9  ...          0       0          0


4       5   35          8  ...          0       0          1

...    ...  ...        ...  ...        ...     ...        ...

4995  4996   29          3  ...          0       1          0

4996  4997   30          4  ...          0       1          0

4997  4998   63         39  ...          0       0          0

4998  4999   65         40  ...          0       1          0

4999  5000   28          4  ...          0       1          1


[5000 rows x 14 columns]>

Confusion Matrix:  [[712   7]

 [233  48]]

Accuracy Score:  0.76
```

Graph:



Practical 3	
Aim: To Demonstrate Classification algorithm of Decision Tree on the given Dataset.	
Name: Labhesh Joshi	Roll No: KCTBCS030
Performance date: 10-08-2022	Sign:

Theory

Clustering?

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means clustering algorithm mainly performs two tasks:

- Determines the best value for K center points or centroids by an iterative process.
- Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

Algorithm for K-means clustering

The following are the steps involved in K-Means clustering:
Step-1: Select the number K to decide the number of clusters.
Step-2: Select random K points or centroids. (It can be others from the input dataset).
Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.
Step-4: Calculate the variance and place a new centroid of each cluster.
Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.
Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.
Step-7: The model is ready.

Advantages of K-Means Clustering

- Relatively simple to implement.
- Scales to large data sets.
- Guarantees convergence.
- Can warm-start the positions of centroids.
- Easily adapts to new examples.
- Generalizes to clusters of different shapes and sizes, such as elliptical clusters.

Disadvantages of K-Means Clustering

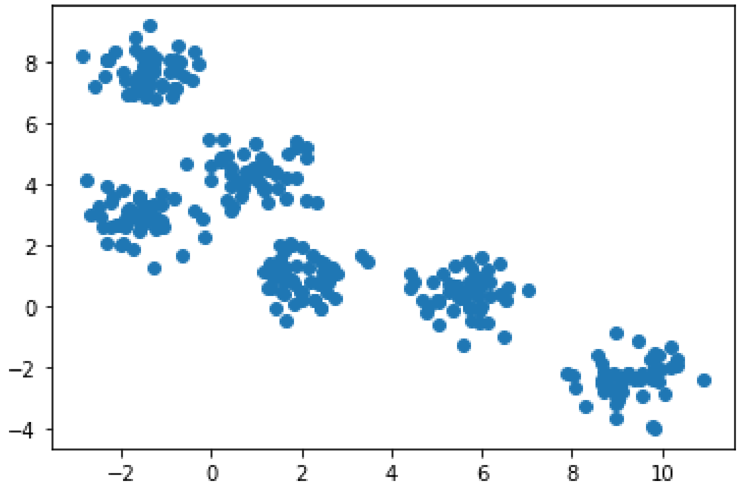
- Choosing k manually.
- Being dependent on initial values.
- Clustering data of varying sizes and density.
- Clustering outliers.
- Scaling with number of dimensions.

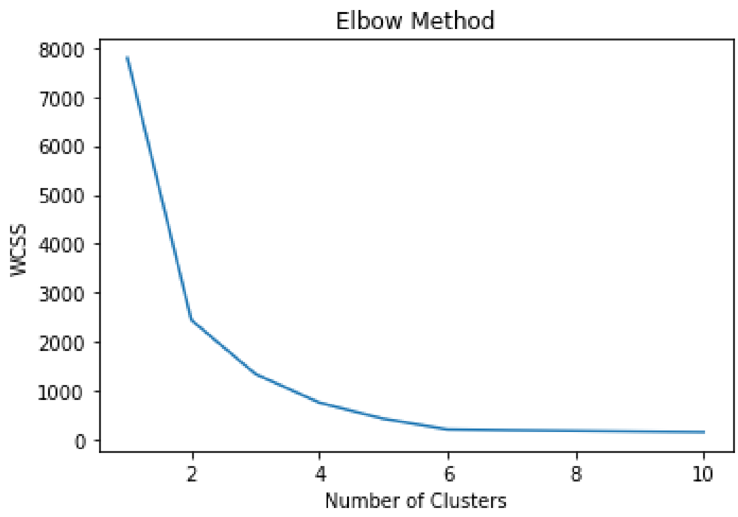
Code: Without Dataset

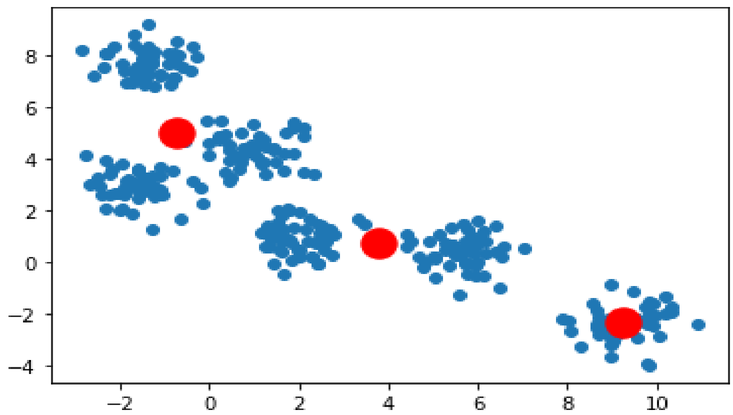
```
import numpy as np

import pandas as pd

from matplotlib import pyplot as plt

from sklearn.datasets import make_blobs

from sklearn.cluster import KMeans
```

```
x, y = make_blobs(n_samples=300, centers=6, cluster_std=0.60, random_state=0)
plt.scatter(x[:,0], x[:,1])
wcss=[]
for i in range(1,11):
   kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=100)
   kmeans.fit(x)
 wcss.append(kmeans.inertia_)
plt.plot(range(1,11), wcss)
plt.title('Elbow Method')
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.show()
kmeans = KMeans(n_clusters=3, init='k-means++', max_iter=100)
pred_y = kmeans.fit_predict(x)
plt.scatter(x[:,0], x[:,1])
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=300, c='red')
plt.show()
```

Output:







Code: With Database

***prac3_b**

import numpy as nm

import matplotlib.pyplot as mtp

import pandas as pd

Importing the dataset

dataset = pd.read_csv('Wholesale customers data.csv')

x = dataset.iloc[:, [3, 4]].values

#finding optimal number of clusters using the elbow method

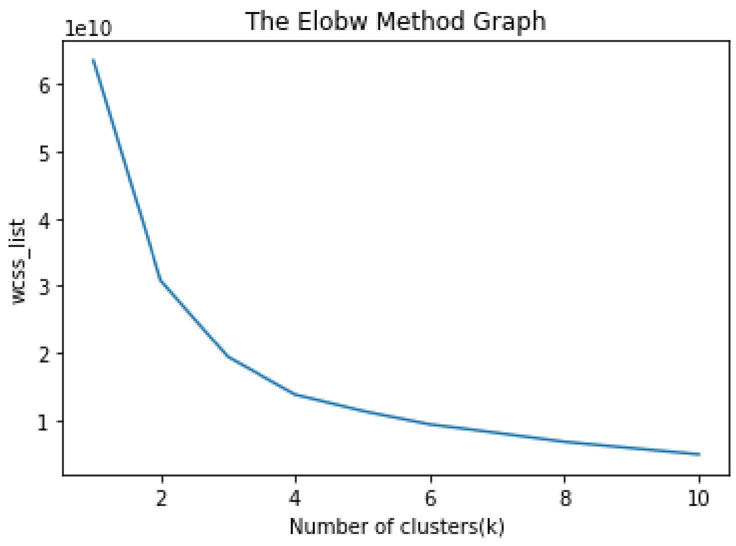
from sklearn.cluster import KMeans

wcss_list= [] #Initializing the list for the values of WCSS

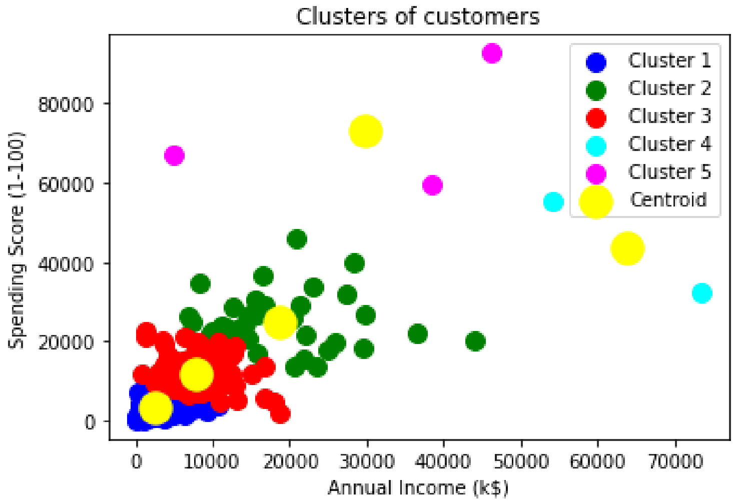
#Using for loop for iterations from 1 to 10.

```
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state= 42)
    kmeans.fit(x)
    wcss_list.append(kmeans.inertia_)
mtp.plot(range(1, 11), wcss_list)
mtp.title('The Elobw Method Graph')
mtp.xlabel('Number of clusters(k)')
mtp.ylabel('wcss_list')
mtp.show()

#training the K-means model on a dataset
kmeans = KMeans(n_clusters=5, init='k-means++', random_state= 42)
y_predict= kmeans.fit_predict(x)

#visulaizing the clusters
mtp.scatter(x[y_predict == 0, 0], x[y_predict == 0, 1], s = 100, c = 'blue', label = 'Cluster 1') #for first cluster
mtp.scatter(x[y_predict == 1, 0], x[y_predict == 1, 1], s = 100, c = 'green', label = 'Cluster 2') #for second cluster
mtp.scatter(x[y_predict== 2, 0], x[y_predict == 2, 1], s = 100, c = 'red', label = 'Cluster 3') #for third cluster
mtp.scatter(x[y_predict == 3, 0], x[y_predict == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4') #for fourth cluster
mtp.scatter(x[y_predict == 4, 0], x[y_predict == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5') #for fifth cluster
mtp.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 300, c = 'yellow', label = 'Centroid')
mtp.title('Clusters of customers')
mtp.xlabel('Annual Income (k$)')
mtp.ylabel('Spending Score (1-100)')
mtp.legend()
mtp.show()
```

Output:



The Elobw Method Graph
1e10
6
5
4
3
2
1
wcss_list
2
4
6
8
10
Number of clusters(k)



Practical 4	
<u>Aim</u>: To implement any one Hierarchical Clustering method for the given dataset.	
Name: Labhesh Joshi	Roll No: KCTBCS030
Performance date: __/__/2022	Sign:

Theory:

Definitions for Hierarchical Clustering.

- **Agglomerative Hierarchical Clustering:** This method is also called a bottom-up approach. In this method, each node represents a single cluster at the beginning; eventually, nodes start merging based on their similarities until all nodes belong to the same cluster.

- **Divisive Hierarchical Clustering:** This method is also called a top-down approach. Initially, all nodes belong to the same cluster; eventually, each node forms its own cluster. Divisive approach is less widely used due to its complexity compared with agglomerative approach.
- **Dendrogram:** A Dendrogram is a tree-like diagram that records the sequences of merges or splits.
- **Single Linkage:** It is the Shortest Distance between the closest points of the clusters.
- **Complete Linkage:** It is the farthest distance between the two points of two different clusters. It is one of the popular linkage methods as it forms tighter clusters than single-linkage.
- **Average Linkage:** It is the linkage method in which the distance between each pair of datasets is added up and then divided by the total number of datasets to calculate the average distance between two clusters. It is also one of the most popular linkage methods.
- **Centroid Linkage:** It is the linkage method in which the distance between the centroid of the clusters is calculated.

1. Hierarchical Clustering Algorithm Steps.

- **Step-1:** Compute the proximity matrix.
- **Step-2:** Assign each data point as a single cluster.
- **Step-3:** Merge two closest data points or clusters to form one cluster.
- **Step-4:** Update the proximity matrix.
- **Step-5:** Repeat Step 3 and Step 4 until only a single cluster remains.
- **Step-6:** Once all the clusters are combined into one big cluster, develop the dendrogram to divide the clusters as per the problem.

Advantages and disadvantages of Hierarchical Clustering.

Advantages:

- Dendrograms help us in clear visualization, which is practical and easy to understand.

- No prior information about the number of clusters is required.
- Easy to use and implement.
- We can obtain the optimal number of clusters from the model itself, human intervention not required.

Disadvantages:

- Not suitable for large datasets due to high time and space complexity.
- There is no mathematical objective for Hierarchical clustering.
- The order of the data has an impact on the final results.
- It is very sensitive to outliers.

Code:

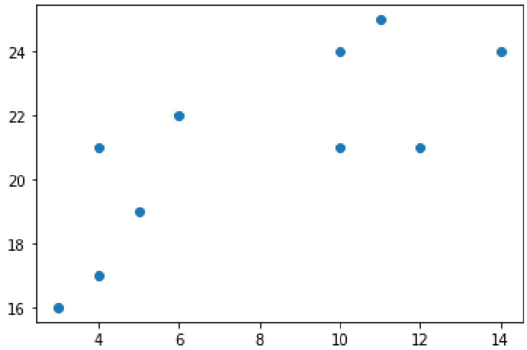
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import AgglomerativeClustering

x = [4, 5, 10, 4, 3, 11, 14 , 6, 10, 12]
y = [21, 19, 24, 17, 16, 25, 24, 22, 21, 21]

plt.scatter(x, y)
plt.show()

from scipy.cluster.hierarchy import dendrogram, linkage

data = list(zip(x, y))
print(data)


linkage_data = linkage(data, method='ward', metric='euclidean')
```

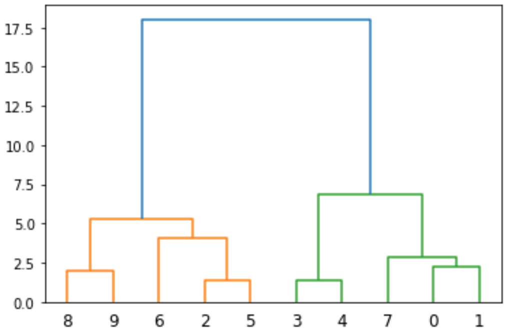
```
dendrogram(linkage_data)

plt.show()


hierarchical_cluster = AgglomerativeClustering(n_clusters=2, affinity='euclidean',
linkage='ward')
labels = hierarchical_cluster.fit_predict(data)

plt.scatter(x, y, c=labels,cmap ='rainbow')
plt.show()


data = pd.read_csv('E:/SEM-5/Data Warehousing and Mining/Wholesale customers
data(1).csv')
data.head()

from sklearn.preprocessing import normalize
data_scaled = normalize(data)
data_scaled = pd.DataFrame(data_scaled, columns=data.columns)
data_scaled.head()


import scipy.cluster.hierarchy as shc
plt.figure(figsize=(10, 7))
plt.title("Dendrograms")
dend = shc.dendrogram(shc.linkage(data_scaled, method='ward'))


from sklearn.cluster import AgglomerativeClustering
cluster = AgglomerativeClustering(n_clusters=2, affinity='euclidean', linkage='ward')
cluster.fit_predict(data_scaled)

plt.figure(figsize=(10, 7))
plt.scatter(data_scaled['Milk'], data_scaled['Grocery'], c=cluster.labels_, cmap ='rainbow')
```

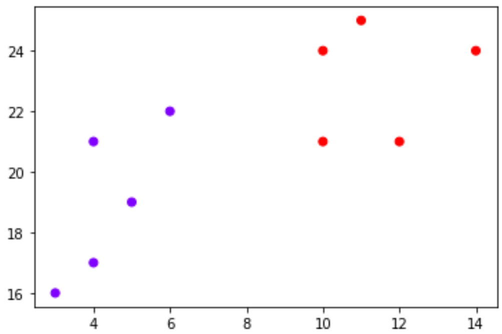
Output:

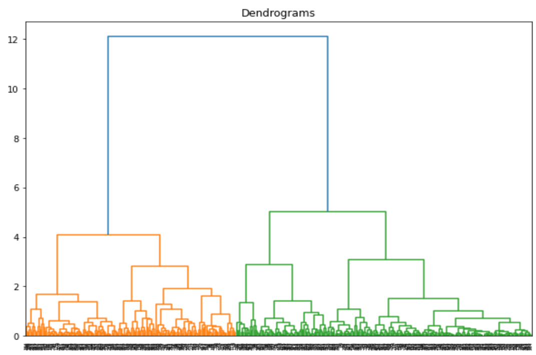
```
runfile('E:/SEM-5/Data Warehousing and Mining/Prac4.py', wdir='E:/SEM-5/Data
Warehousing and Mining')
[(4, 21), (5, 19), (10, 24), (4, 17), (3, 16), (11, 25), (14, 24), (6, 22), (10, 21), (12, 21)]
```

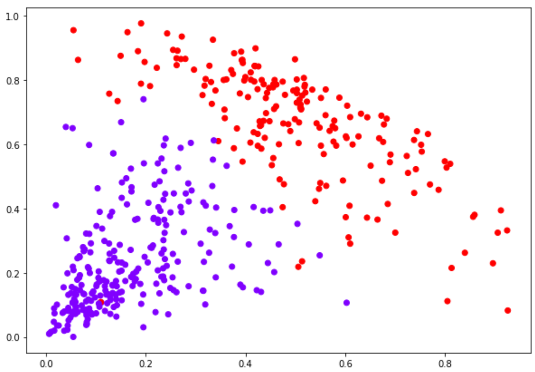
Graphs:











Practical 5	
Aim: Implement Association Rule Mining algorithm (Apriori).	
Name: Labhesh Joshi	Roll No: KCTBCS030
Performance date: __/__/2022	Sign:

Theory:

Unsupervised learning

Unsupervised learning is a machine learning technique in which models are not supervised using training dataset. Instead, models are trained using unlabeled dataset and are allowed to act on that data without any supervision. It can be compared to learning which takes place in the human brain while learning new things.

- Unsupervised learning is helpful for finding useful insights from the data.
- Unsupervised learning is much similar as a human learns to think by their own experiences, which makes it closer to the real AI.
- Unsupervised learning works on unlabeled and uncategorized data which make unsupervised learning more important.
- In real-world, we do not always have input data with the corresponding output so to solve such cases, we need unsupervised learning.

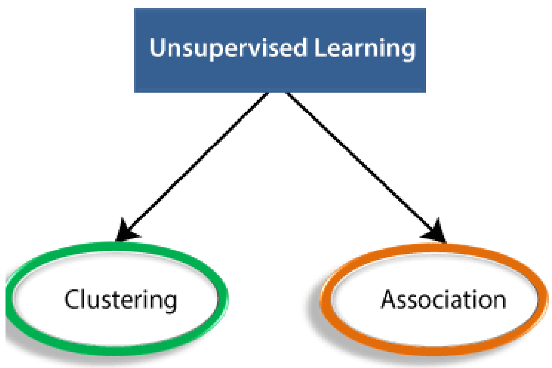
The unsupervised learning algorithm can be further categorized into two types of problems:

Clustering: Clustering is a method of grouping the objects into clusters such that objects with most similarities remains into a group and has less or no similarities with the objects of another group. Cluster analysis finds the commonalities between the data objects and categorizes them as per the presence and absence of those commonalities.

Association: An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database. It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective. Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item. A typical example of Association rule is Market Basket Analysis.

Below is the list of some popular unsupervised learning algorithms:

- K-means clustering
- Hierarchal clustering
- Apriori algorithm



Advantages of Unsupervised Learning

Unsupervised learning is used for more complex tasks as compared to supervised learning because, in unsupervised learning, we don't have labeled input data. Unsupervised learning is preferable as it is easy to get unlabeled data in comparison to labeled data.

Disadvantages of Unsupervised Learning

Unsupervised learning is intrinsically more difficult than supervised learning as it does not have corresponding output. The result of the unsupervised learning algorithm might be less accurate as input data is not labeled, and algorithms do not know the exact output in advance.

Market basket analysis

Market basket analysis is a data mining technique used by retailers to increase sales by better understanding customer purchasing patterns. It involves analyzing large data sets, such as purchase history, to reveal product groupings and products that are likely to be purchased together.

Market Basket Analysis techniques can be categorized based on how the available data is utilized. Here are the following types of market basket analysis in data mining, such as:

A. **Descriptive market basket analysis:** This type only derives insights from past data and is the most frequently used approach. The analysis here does not make any predictions but rates the association between products using statistical techniques. For those familiar with the basics of Data Analysis, this type of modelling is known as unsupervised learning.

B. **Predictive market basket analysis:** This type uses supervised learning models like classification and regression. It essentially aims to mimic the market to analyze what causes what to happen. Essentially, it considers items purchased in a sequence to determine cross-selling.

C. **Differential market basket analysis:** This type of analysis is beneficial for competitor analysis. It compares purchase history between stores, between seasons, between two time periods, between different days of the week, etc., to find interesting patterns in consumer behaviour.

1. Apriori algorithm

Apriori algorithm refers to the algorithm which is used to calculate the association rules between objects. It means how two or more objects are related to one another. In other words, we can say that the apriori algorithm is an association rule leaning that analyzes that people who bought product A also bought product B.

The primary objective of the apriori algorithm is to create the association rule between different objects. The association rule describes how two or more objects are related to one another. Apriori algorithm is also called frequent pattern mining. Generally, you operate the Apriori algorithm on a database that consists of a huge number of transactions.

2. Advantages and disadvantages of apriori

Advantages of Apriori Algorithm

- It is used to calculate large itemsets.
- Simple to understand and apply.

Disadvantages of Apriori Algorithms

- Apriori algorithm is an expensive method to find support since the calculation has to pass through the whole database.
- Sometimes, you need a huge number of candidate rules, so it becomes computationally more expensive.

Code:

```
import numpy as np
import pandas as pd
from apyori import apriori

store_data=pd.read_csv('E:/SEM-5/Data Warehousing and Mining/Day 1.csv',header=None)
print(store_data)
records=[]
for i in range(0,21):
    records.append([str(store_data.values[i,j]) for j in range(0,5)])
print(records)

a_rule=apriori(records, min_supprt=0.5, min_confidence=0.7, min_lift=1.2, min_length=2)
a_results=list(a_rule)

print(len(a_results))
print(a_results)

for i in a_results:
    print(i)
    print('\n')
```

Output:

runfile('E:/SEM-5/Data Warehousing and Mining/Prac5.py', wdir='E:/SEM-5/Data Warehousing and Mining')

```
      0      1      2     3      4
0   Wine  Chips  Bread  Milk  Apple
1   Wine    NaN  Bread  Milk    NaN
2    NaN  Chips  Bread  Milk    NaN
3    NaN  Chips    NaN   NaN  Apple
4   Wine  Chips  Bread  Milk  Apple
5   Wine  Chips    NaN  Milk    NaN
6   Wine  Chips  Bread   NaN  Apple
7   Wine  Chips    NaN  Milk  Apple
8   Wine    NaN  Bread   NaN  Apple
9   Wine    NaN  Bread  Milk    NaN
10   NaN  Chips  Bread   NaN  Apple
11  Wine    NaN    NaN  Milk  Apple
12  Wine  Chips  Bread  Milk    NaN
13  Wine    NaN  Bread  Milk  Apple
14  Wine    NaN  Bread  Milk  Apple
15  Wine  Chips  Bread  Milk  Apple
16   NaN  Chips  Bread  Milk  Apple
17   NaN  Chips    NaN  Milk  Apple
18  Wine  Chips  Bread  Milk  Apple
19  Wine  Chips  Bread  Milk  Apple
20  Wine  Chips  Bread  Milk  Apple
21   NaN  Chips    NaN   NaN    NaN
[['Wine', 'Chips', 'Bread', 'Milk', 'Apple'], ['Wine', 'nan', 'Bread', 'Milk', 'nan'], ['nan', 'Chips', 'Bread', 'Milk', 'nan'], ['nan', 'Chips', 'nan', 'nan', 'Apple'], ['Wine', 'Chips', 'Bread', 'Milk', 'Apple'], ['Wine', 'Chips', 'nan', 'Milk', 'nan'], ['Wine', 'Chips', 'Bread', 'nan', 'Apple'], ['Wine', 'Chips', 'nan', 'Milk', 'Apple'], ['Wine', 'nan', 'Bread', 'nan', 'Apple'], ['Wine', 'nan', 'Bread', 'Milk', 'nan'], ['nan', 'Chips', 'Bread', 'nan', 'Apple'], ['Wine', 'nan', 'nan', 'Milk', 'Apple'], ['Wine', 'Chips', 'Bread', 'Milk', 'nan'], ['Wine', 'nan', 'Bread', 'Milk', 'Apple'], ['Wine', 'nan', 'Bread', 'Milk', 'Apple'], ['Wine', 'Chips', 'Bread', 'Milk', 'Apple'], ['nan', 'Chips', 'Bread', 'Milk', 'Apple'], ['nan', 'Chips', 'nan', 'Milk', 'Apple'], ['Wine', 'Chips', 'Bread', 'Milk', 'Apple'], ['Wine', 'Chips', 'Bread', 'Milk', 'Apple'], ['Wine', 'Chips', 'Bread', 'Milk', 'Apple']]
3
[RelationRecord(items=frozenset({'Wine', 'Apple', 'Bread', 'Chips'}),
support=0.3333333333333333,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'Wine', 'Chips'}),
items_add=frozenset({'Apple', 'Bread'}), confidence=0.7, lift=1.225)]),
RelationRecord(items=frozenset({'Wine', 'Apple', 'Chips', 'Milk'}),
support=0.3333333333333333,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'Wine', 'Chips'}),
items_add=frozenset({'Apple', 'Milk'}), confidence=0.7, lift=1.225)]),
RelationRecord(items=frozenset({'Bread', 'Chips', 'Wine', 'Apple', 'Milk'}),
support=0.2857142857142857,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'Wine', 'Apple', 'Chips'}),
items_add=frozenset({'Bread', 'Milk'}), confidence=0.75, lift=1.2115384615384615),
OrderedStatistic(items_base=frozenset({'Wine', 'Bread', 'Chips'}),
items_add=frozenset({'Apple', 'Milk'}), confidence=0.75, lift=1.3125)])]
RelationRecord(items=frozenset({'Wine', 'Apple', 'Bread', 'Chips'}),
support=0.3333333333333333,
```

ordered_statistics=[OrderedStatistic(items_base=frozenset({'Wine', 'Chips'}), items_add=frozenset({'Apple', 'Bread'}), confidence=0.7, lift=1.225)])

RelationRecord(items=frozenset({'Wine', 'Apple', 'Chips', 'Milk'}), support=0.3333333333333333, ordered_statistics=[OrderedStatistic(items_base=frozenset({'Wine', 'Chips'}), items_add=frozenset({'Apple', 'Milk'}), confidence=0.7, lift=1.225)])

RelationRecord(items=frozenset({'Bread', 'Chips', 'Wine', 'Apple', 'Milk'}), support=0.2857142857142857, ordered_statistics=[OrderedStatistic(items_base=frozenset({'Wine', 'Apple', 'Chips'}), items_add=frozenset({'Bread', 'Milk'}), confidence=0.75, lift=1.2115384615384615), OrderedStatistic(items_base=frozenset({'Wine', 'Bread', 'Chips'}), items_add=frozenset({'Apple', 'Milk'}), confidence=0.75, lift=1.3125)])

Practical 6	
Aim: To install Power BI and perform data visualization on the dataset.	
Name: Labhesh Joshi	Roll No: KCTBCS030
Performance date: __/__/2022	Sign:

Theory:

Power BI

Power BI is an interactive data visualization software product developed by Microsoft with a primary focus on business intelligence. It is part of the Microsoft Power Platform.

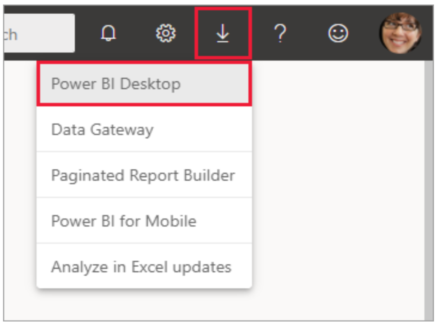
Power BI is used for analyzing and visualizing raw data to present actionable information. It combines business analytics, data visualization, and best practices that help an organization to make data-driven decisions.

Steps in downloading Power Bi

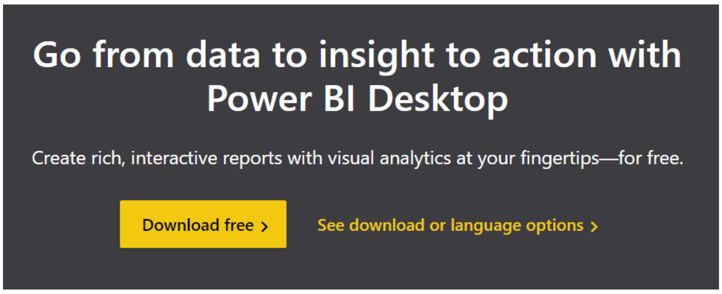
Install as an app from the Microsoft Store

There are a few ways to access the most recent version of Power BI Desktop from the Microsoft Store.

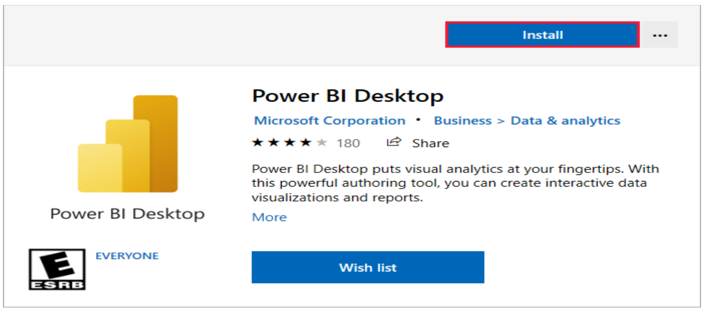
1. Use one of the following options to open the Power BI Desktop page of the Microsoft Store:

- Open a browser and go directly to the Power BI Desktop page of the Microsoft Store.
- From the Power BI service, in the upper right corner, select the Download icon and then choose Power BI Desktop.
    - 

- Go to the Power BI Desktop product page, and then select Download Free.
    - **https://powerbi.microsoft.com/en-us/desktop/**



2. After you've landed on the Power BI Desktop page of the Microsoft Store, select Install.

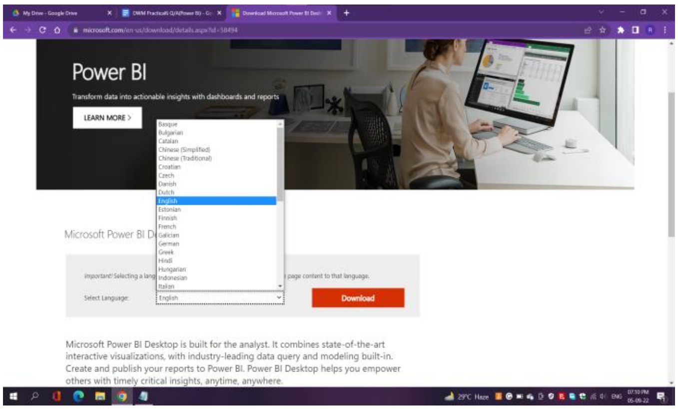


3. It directly gets installed and opens to be used.

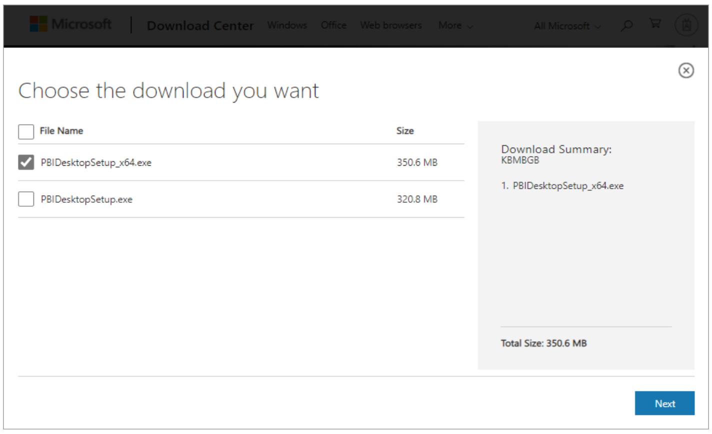
Download Power BI Desktop directly

1. To download the Power BI Desktop executable from the Download Center, select Download from the Download Center page.

https://www.microsoft.com/en-us/download/details.aspx?id=58494

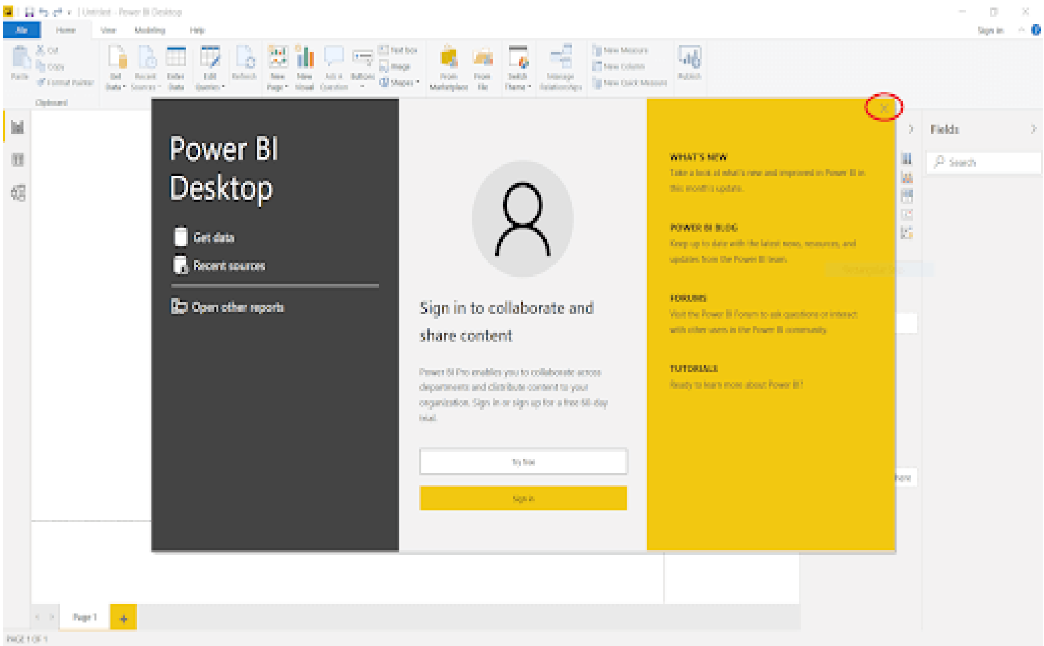


2. Select the language as English and click on download.
3. Then, specify a 32-bit or 64-bit installation file to download.



4. Click next, after it gets downloaded launch the installation package.
5. Once you open the downloadable file on your system, you will notice the following dialog box.
6. Once you click the Next button, you will be asked to click the Next button to continue or the Cancel button to exit in the dialog box.
7. The license agreement dialog box is displayed once you click the Next button. Now the Next button will be enabled after you click the checkbox.
8. Click the Next button to open the Destination folder. The folder allows you to either leave the default C location or use the Change button to alter your desired location for installing Power BI Desktop Application in your device.

9. Please, click the Next button to give you the following alternatives.
10. Are you ready to Install? If, yes click the Install button (or) Do you want to review or change any of the installation settings? If yes, click Back Button. Next, click the Install button for installation.
11. Please, wait until the installation is finished.
12. Click the Finish button to initialize the process.
13. Please, Wait for a few seconds to start Power BI Desktop.
14. Here you can see the installed Power BI Desktop Page. Let me Close the start page.

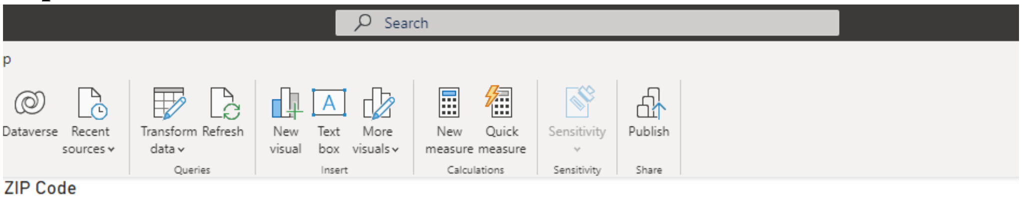


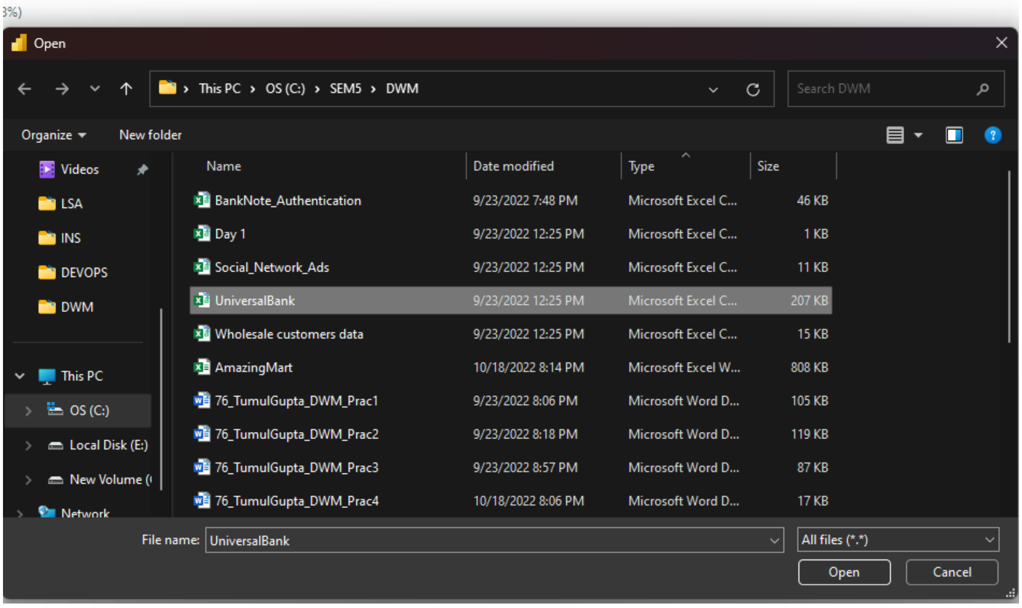
Advantages of Power Bi

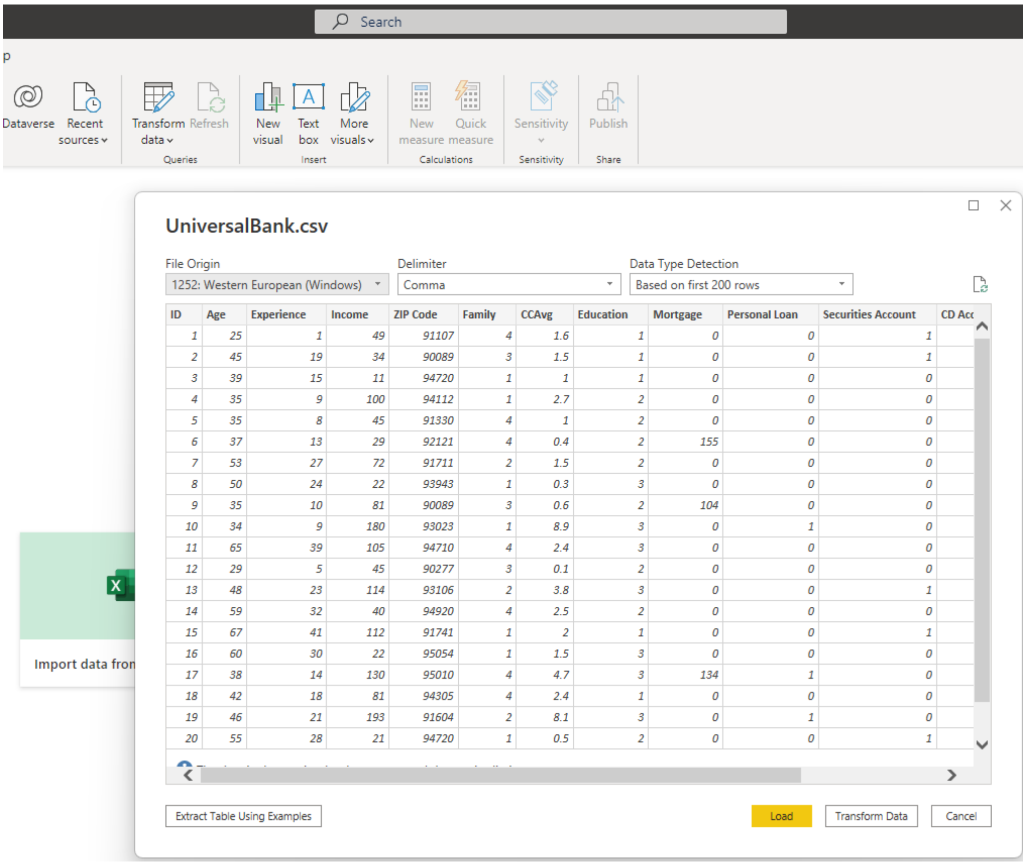
Key features and benefits of Microsoft Power BI include:

1. Power BI can provide business intelligence for all
2. Power BI brings data to life (interactivity)
3. Power BI is secure
4. Power BI easily connects to many data sources
5. Power BI has artificial intelligence capabilities
6. Power BI is constantly improved
7. Power BI apps – an excellent means of sharing content

1. Import the Universal bank dataset.

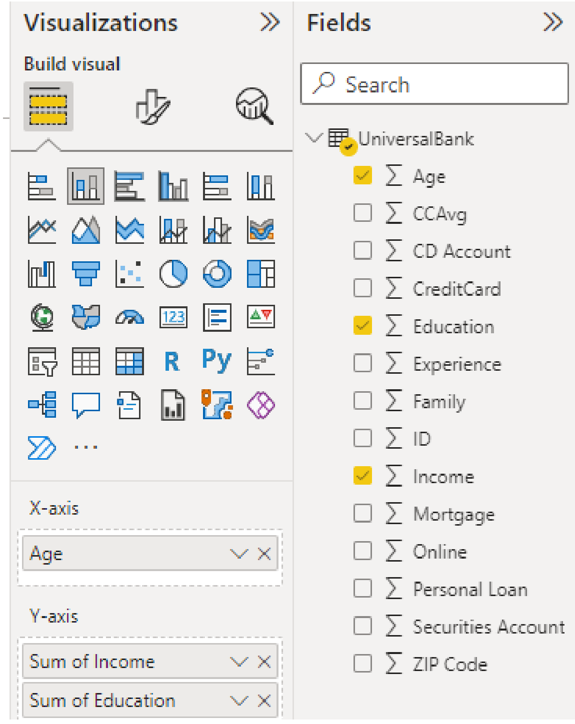


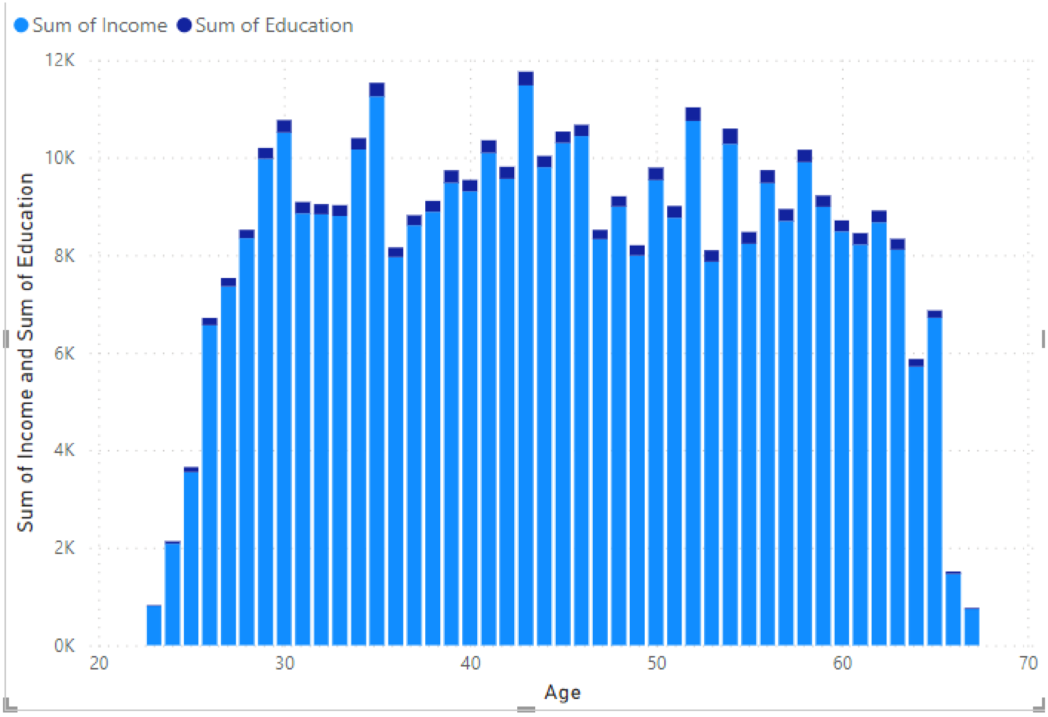




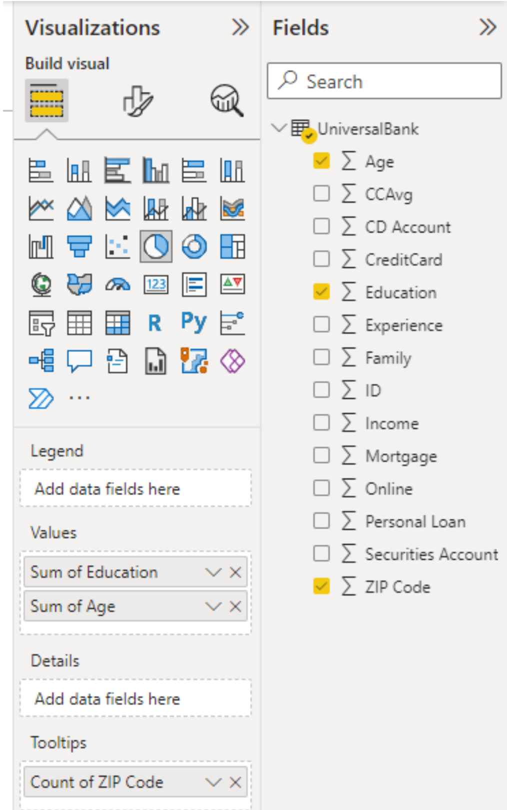
ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account
1	25	1	49	91107	4	1.6	1	0	0	1
2	45	19	34	90089	3	1.5	1	0	0	1
3	39	15	11	94720	1	1	1	0	0	0
4	35	9	100	94112	1	2.7	2	0	0	0
5	35	8	45	91330	4	1	2	0	0	0
6	37	13	29	92121	4	0.4	2	155	0	0
7	53	27	72	91711	2	1.5	2	0	0	0
8	50	24	22	93943	1	0.3	3	0	0	0
9	35	10	81	90089	3	0.6	2	104	0	0
10	34	9	180	93023	1	8.9	3	0	1	0
11	65	39	105	94710	4	2.4	3	0	0	0
12	29	5	45	90277	3	0.1	2	0	0	0
13	48	23	114	93106	2	3.8	3	0	0	1
14	59	32	40	94920	4	2.5	2	0	0	0
15	67	41	112	91741	1	2	1	0	0	1
16	60	30	22	95054	1	1.5	3	0	0	0
17	38	14	130	95010	4	4.7	3	134	1	0
18	42	18	81	94305	4	2.4	1	0	0	0
19	46	21	193	91604	2	8.1	3	0	1	0
20	55	28	21	94720	1	0.5	2	0	0	1

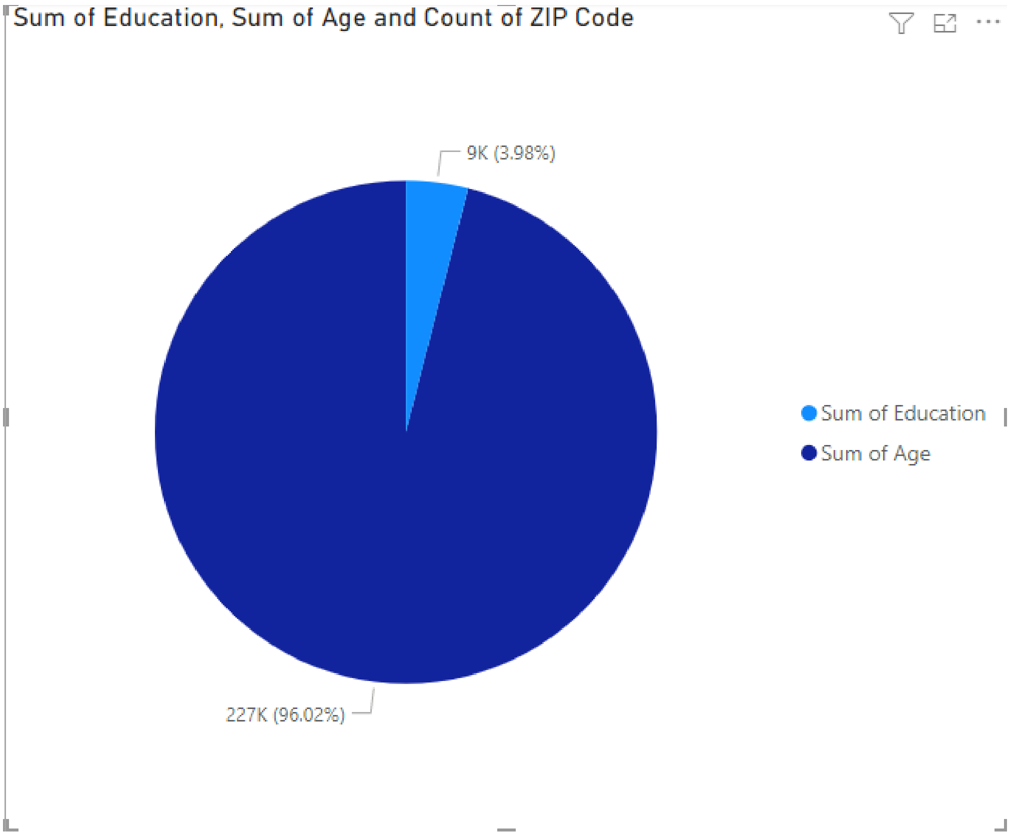
2. **Plot a stacked column chart of Income and education w.r.t. age.**



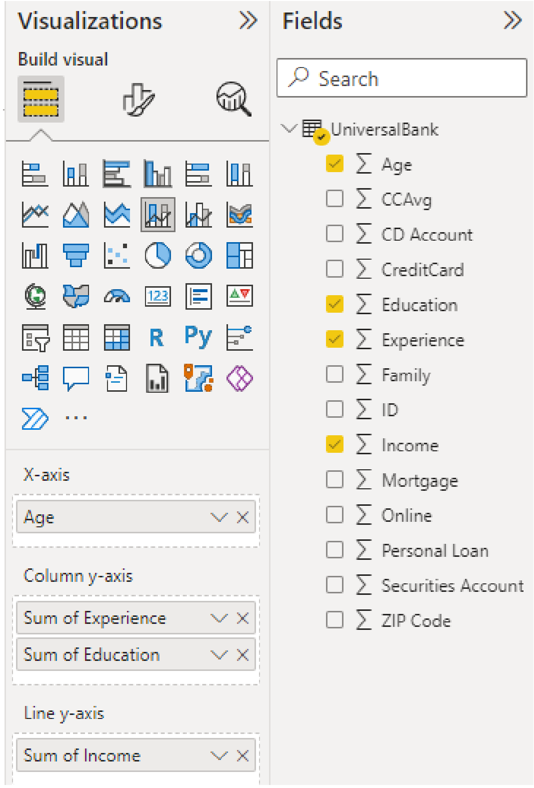


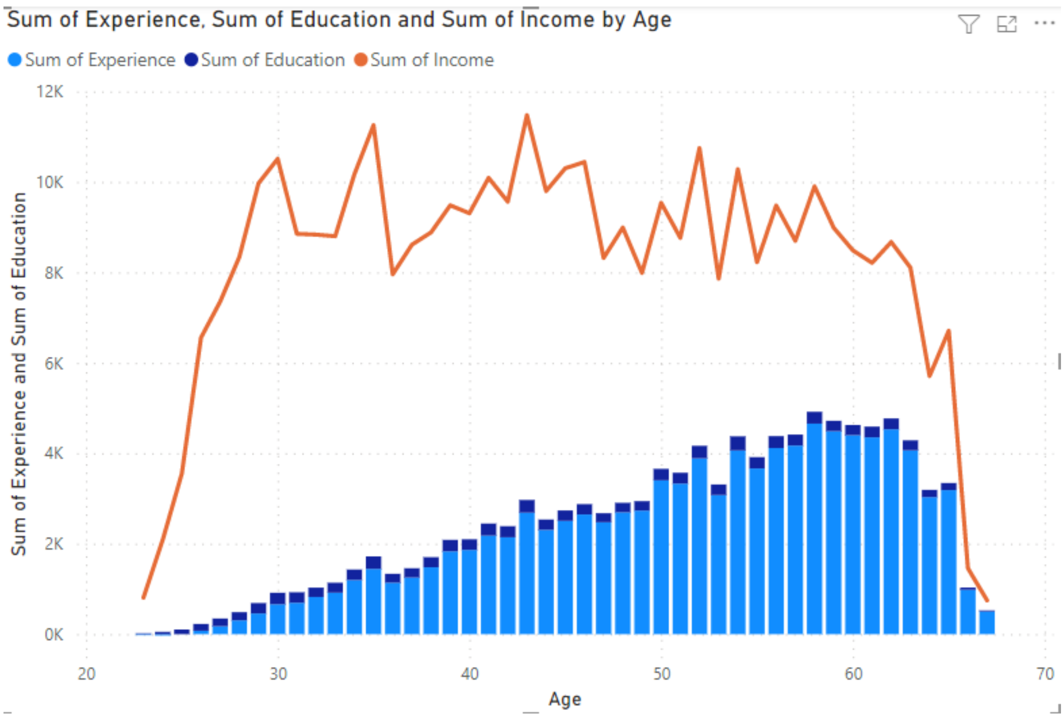
3. **Plot the pie chart of education qualification with age. Provide the zip code details using the tool tip.**



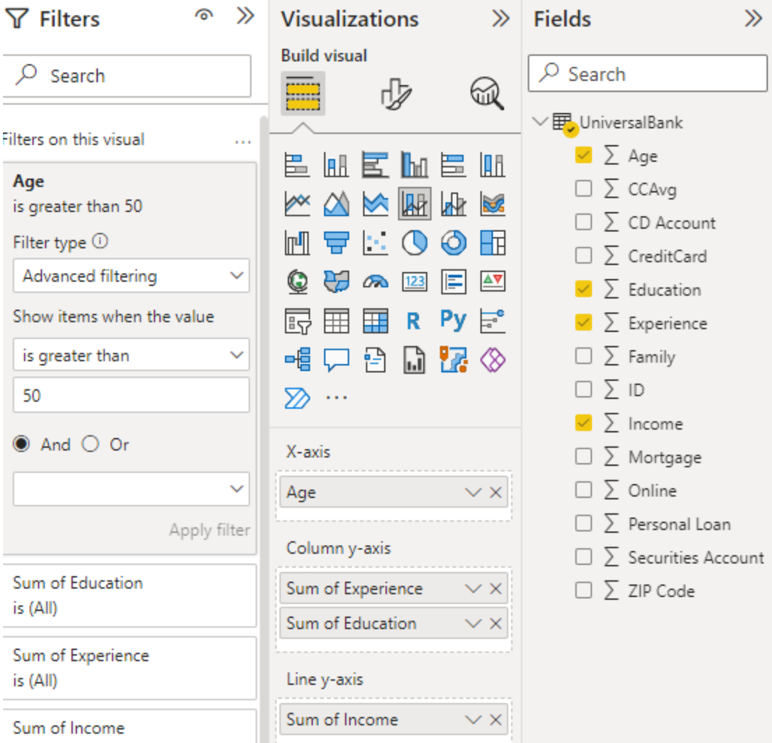


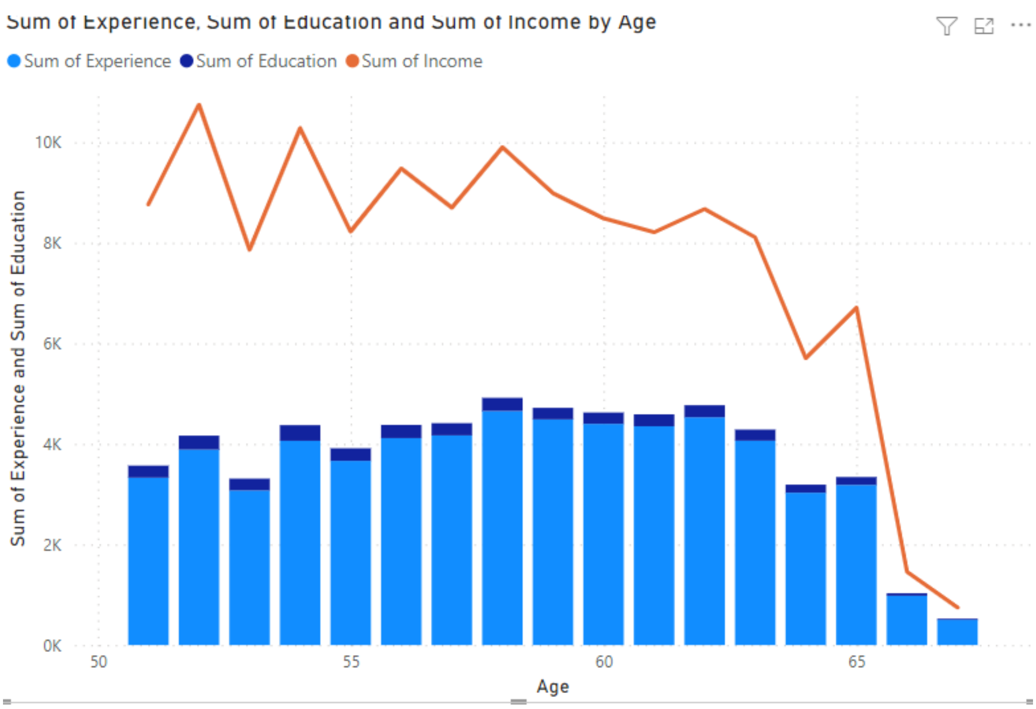
4. **Plot the line and column stacked chart showing age and experience and education in column chart and income on line chart.**



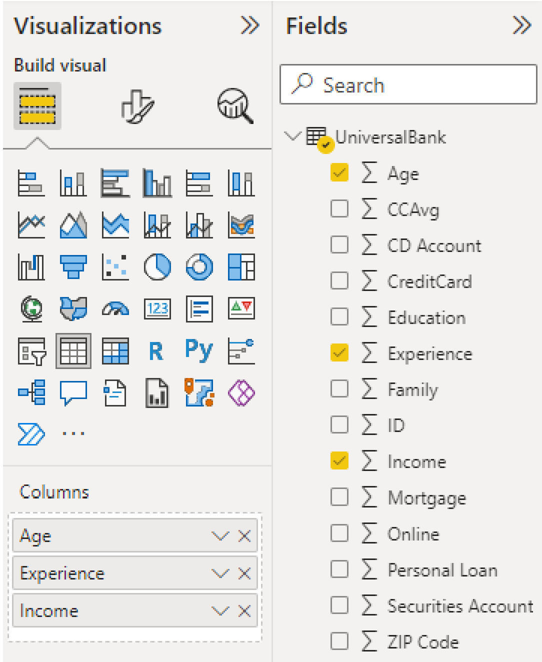


5. **Modify the above column stacked chart for age group greater than 50.**





6. **View age, experience, income data in tabular format using table.**



Age	Experience	Income
67	43	41
67	43	79
67	43	105
66	42	35
66	42	39
66	42	53
66	42	95
67	42	21
67	42	32
67	42	51
67	42	75
65	41	40
65	41	42
65	41	45
65	41	51
65	41	55

<table>
<tr><td colspan="2" align="center">Practical 7</td></tr>
<tr><td colspan="2"><u>Aim</u>: To perform the Extraction Transformation and Loading (ETL) process to construct the database in Power BI.</td></tr>
<tr><td>Name: Labhesh Joshi</td><td>Roll No: KCTBCS030</td></tr>
<tr><td>Performance date: __/__/2022</td><td>Sign:</td></tr>
</table>

Theory:

Data Warehouse

A Data Warehouse (DW) is a relational database that is designed for query and analysis rather than transaction processing.

It includes historical data derived from transaction data from single and multiple sources.

It provides integrated, enterprise-wide, historical data and focuses on providing support for decision-makers for data modelling and analysis.

A Data Warehouse is not used for daily operations and transaction processing but used for making decisions.

It can be viewed as a data system with the following attributes:

- It is a database designed for investigative tasks, using data from various applications.
- It supports a relatively small number of clients with relatively long interactions.
- It includes current and historical data to provide a historical perspective of information.
- Its usage is read-intensive.
- It contains a few large tables.

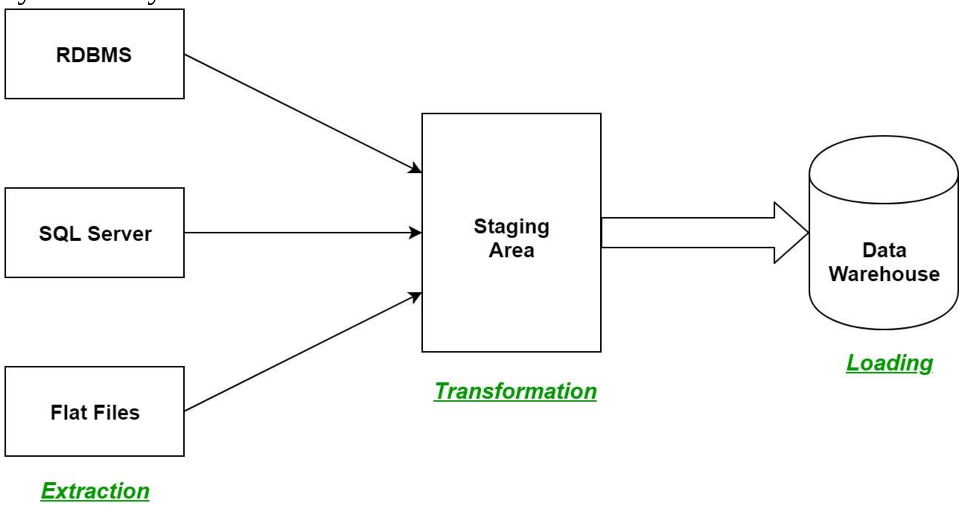
ETL Process

ETL consists of three separate phases:

1. **Extraction:**
    - Data from various source systems is extracted which can be in various formats like relational databases, No SQL, XML, and flat files into the staging area. This is the first step of the ETL process.
    - It is important to extract the data from various source systems and store it into the staging area first and not directly into the data warehouse because the extracted data is in various formats and can be corrupted also.
    - Hence loading it directly into the data warehouse may damage it and rollback will be much more difficult. Therefore, this is one of the most important steps of ETL process.
2. **Transformation:**
    - The second step of the ETL process is transformation.
    - In this step, a set of rules or functions are applied on the extracted data to convert it into a single standard format.

- It may involve following processes/tasks:
    a. Filtering – loading only certain attributes into the data warehouse.
    b. Cleaning – filling up the NULL values with some default values, mapping U.S.A, United States, and America into USA, etc.
    c. Joining – joining multiple attributes into one.
    d. Splitting – splitting a single attribute into multiple attributes.
    e. Sorting – sorting tuples on the basis of some attribute (generally key-attribute).

3. **Loading:**
    - The third and final step of the ETL process is loading.
    - In this step, the transformed data is finally loaded into the data warehouse.
    - Sometimes the data is updated by loading into the data warehouse very frequently and sometimes it is done after longer but regular intervals.
    - The rate and period of loading solely depends on the requirements and varies from system to system.



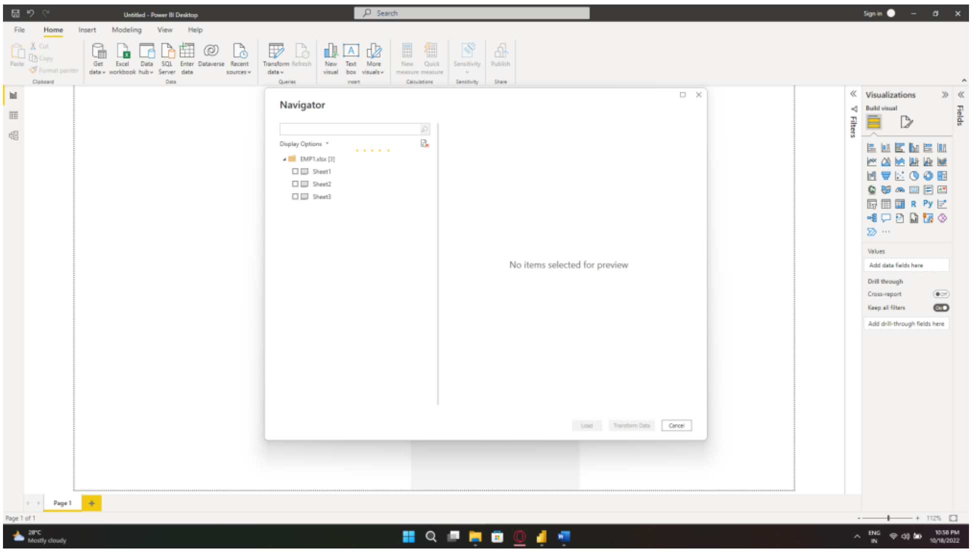
Advantages of Data Warehouse:

- Delivers Enhanced Business Intelligence.
- Ensures Data Quality and Consistency.
- Saves Time and Money.
- Tracks Historically Intelligent Data.
- Generates high ROI

Disadvantages of Data Warehouse:

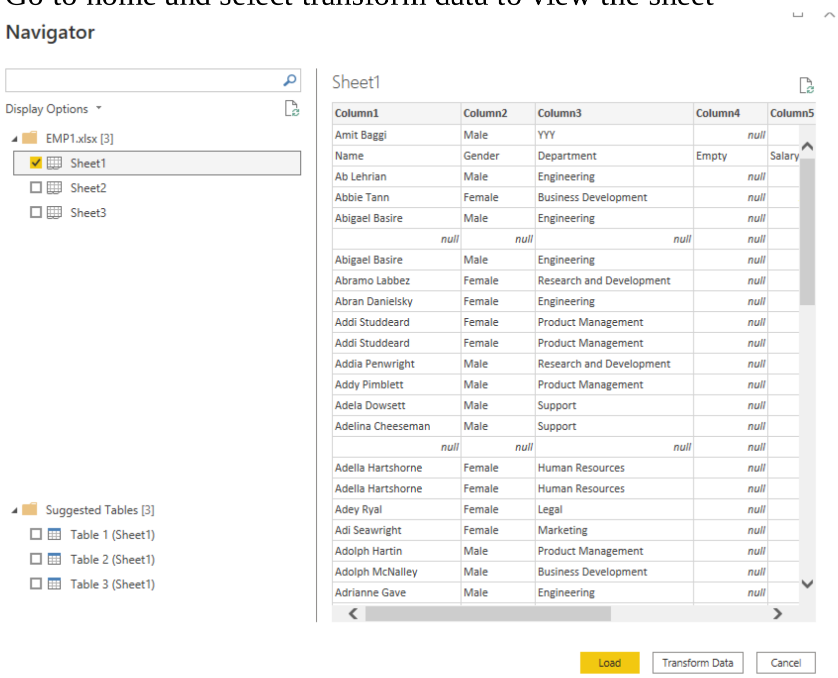
- Extra Report Work.
- Inflexibility and homogenization of data.
- Ownership Concerns.
- Demands for large amounts of resources.
- Hidden issues consume time.

Perform the Extraction Transformation and Loading (ETL) process to construct the database.



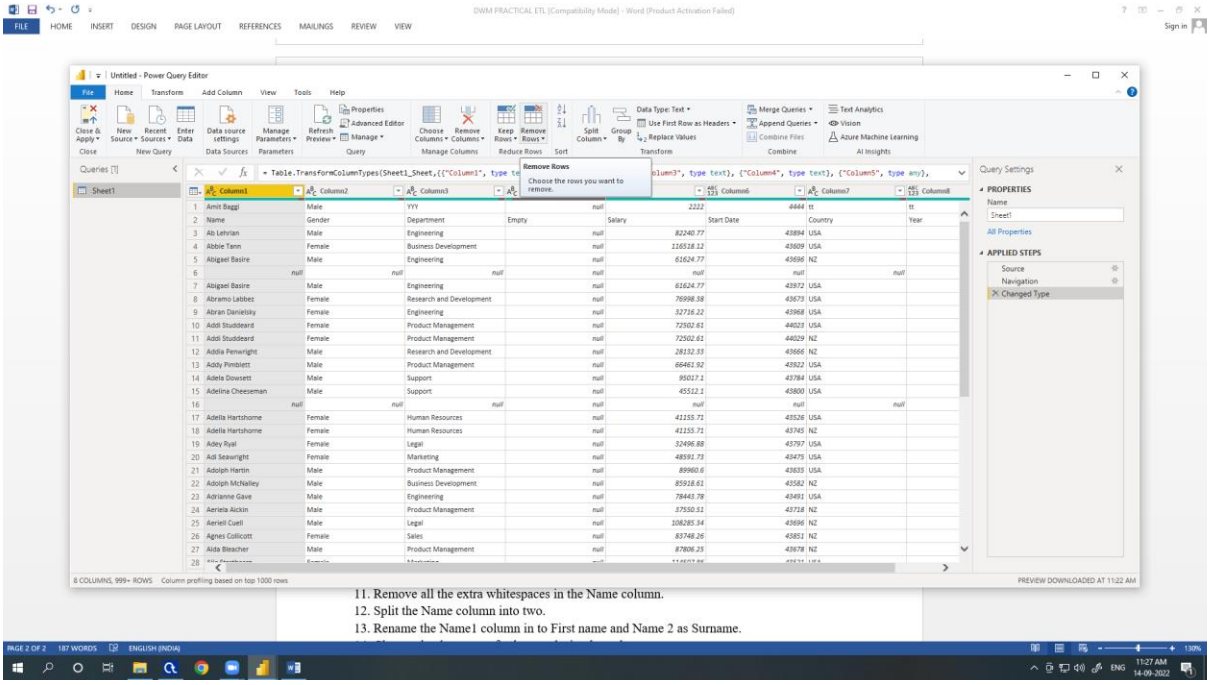
1. Load the Emp1.xlsx

Click Get Data
>Select Excel workbook
>Select “EMP1.xlsx”
>Select “Sheet1”
>Click on load
>Go to home and select transform data to view the sheet

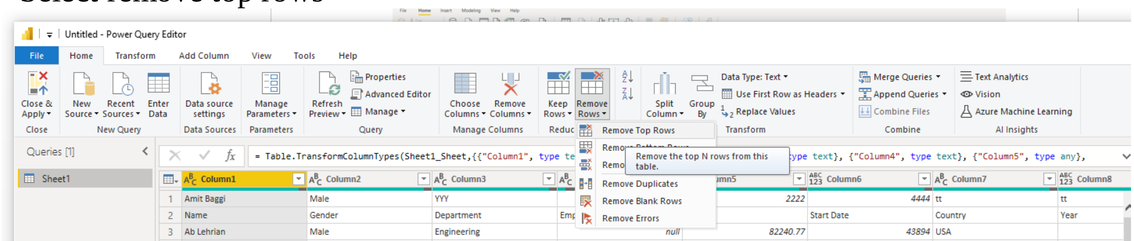


2. Remove the first row

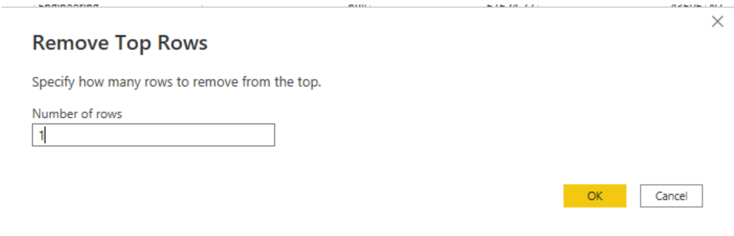
>Go to remove rows option



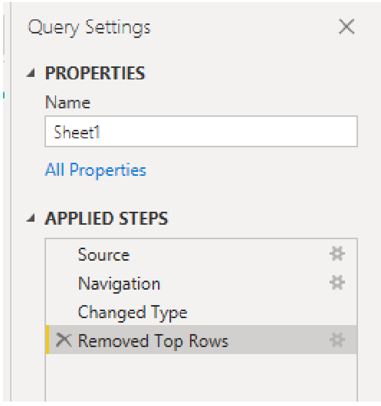
>Select remove top rows



>Enter number of rows to be removed

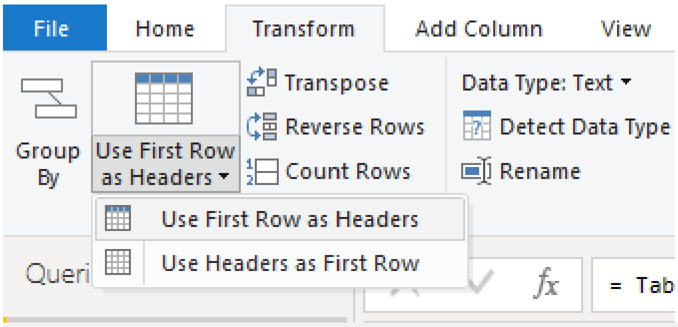


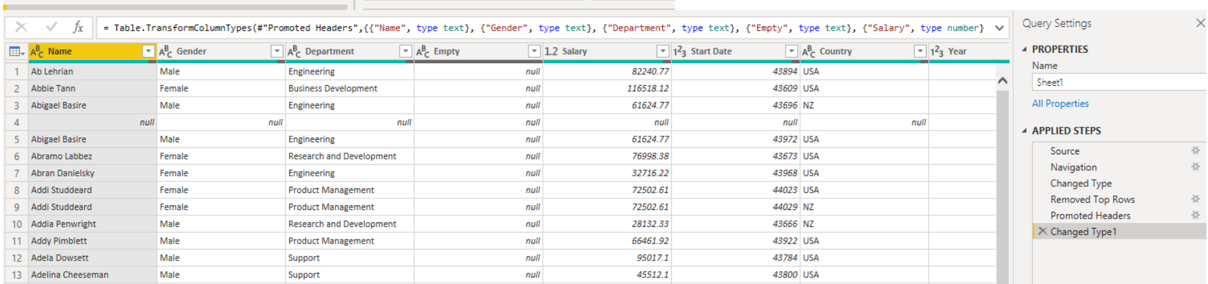
>Performed function will be displayed in “Applied steps” at the right corner



3. Promote the first row as header.

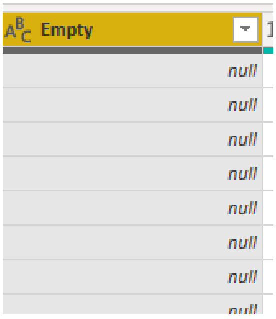
>Go to "Transform" tab and click on "Use first row as header"



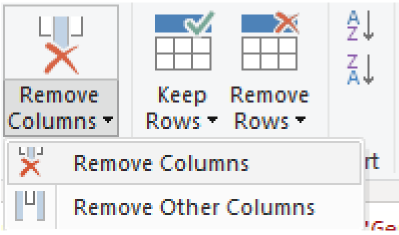


4. Remove the null column.

>Select the empty column

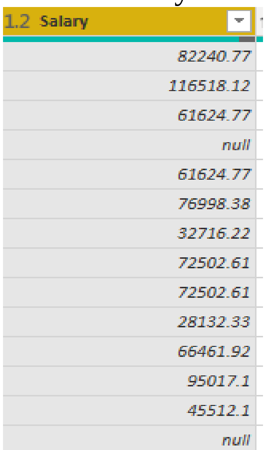


>Select "remove columns" from the tab

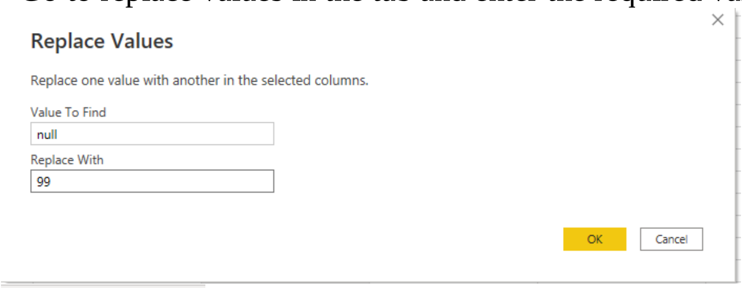


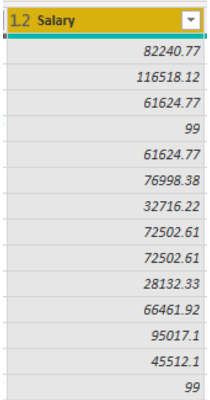
5. Replace the missing values in salary column with 99.

>Select Salary column



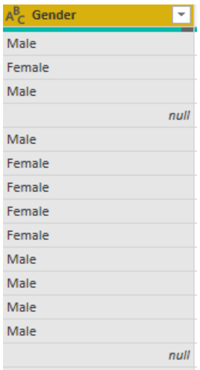
>Go to replace values in the tab and enter the required values



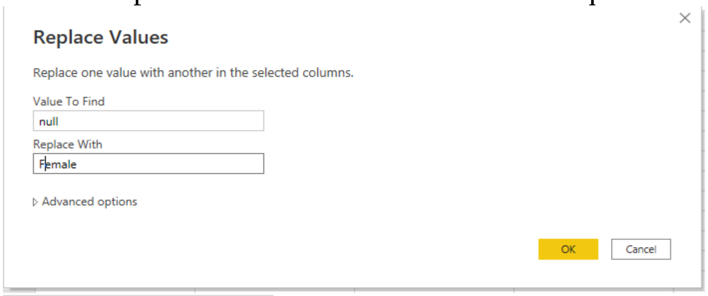


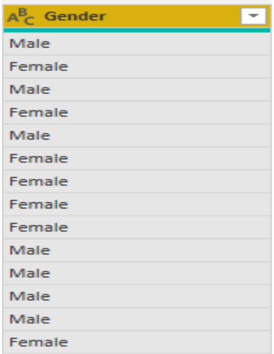
6. Replace the empty spaces in Gender column with Female.

>Select “Gender” column



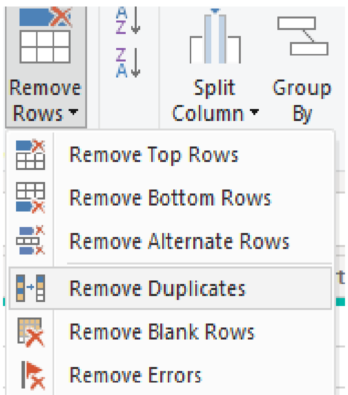
>Select replace values in tab and enter the required values





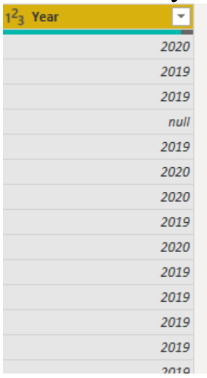
7. Remove duplicate data in name column.

>Select the “name” column. Go to “Remove rows” option in tab and select “remove duplicates”

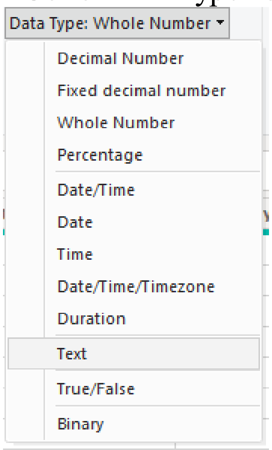


8. Change the data type of Year column as text.

>Select the “year” column



>Go to “Data type” option in tab and select the “text” option

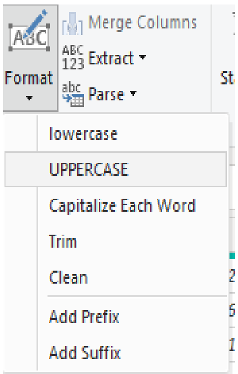


9. Convert the data in the Name column to Upper case.

>Select “name” column

	Name
1	Ab Lehrian
2	Abbie Tann
3	Abigael Basire
4	null
5	Abramo Labbez
6	Abran Danielsky
7	Addi Studdeard
8	Addia Penwright
9	Addy Pimblett
10	Adela Dowsett
11	Adelina Cheeseman

>Select “format” option from “Transform” tab and click on Uppercase option

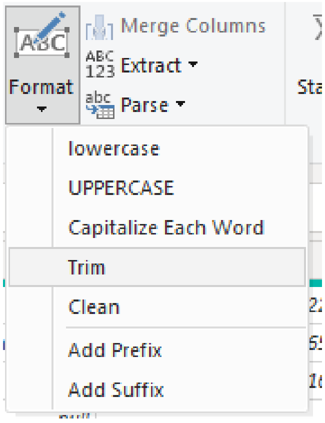


Name
AB LEHRIAN
ABBIE TANN
ABIGAEL BASIRE
null
ABRAMO LABBEZ
ABRAN DANIELSKY
ADDI STUDDEARD
ADDIA PENWRIGHT
ADDY PIMBLETT
ADELA DOWSETT
ADELINA CHEESEMAN

10. Remove all the extra whitespaces in the Name column.

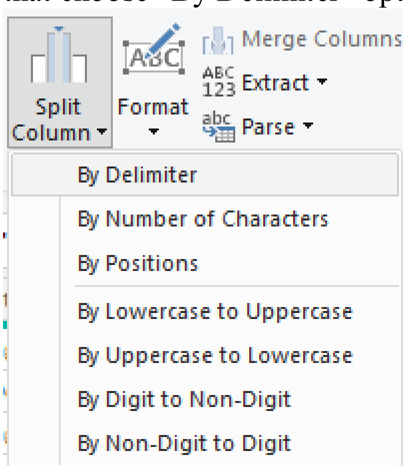
Name
AB LEHRIAN
ABBIE TANN
ABIGAEL BASIRE
null
ABRAMO LABBEZ
ABRAN DANIELSKY
ADDI STUDDEARD
ADDIA PENWRIGHT
ADDY PIMBLETT
ADELA DOWSETT
ADELINA CHEESEMAN

>Select “trim” option in “format” available in “transform” tab

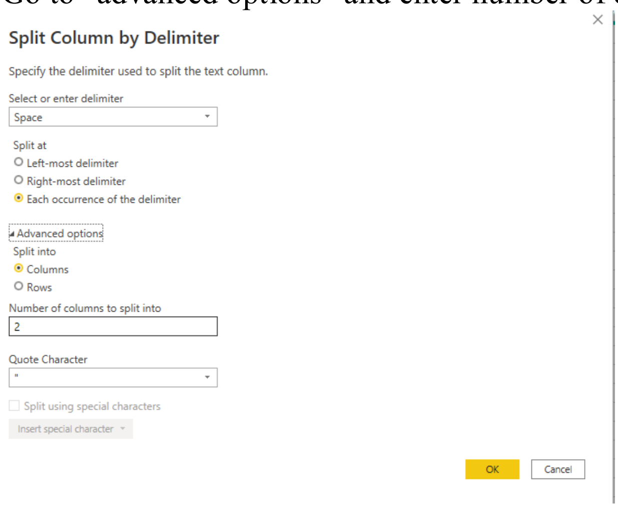


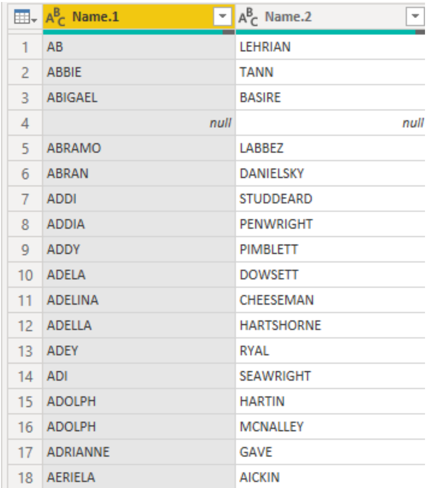
11. Split the Name column into two.

>Select “name” column. Go into “transform” tab and click on “split column” option after that choose “By Delimiter” option



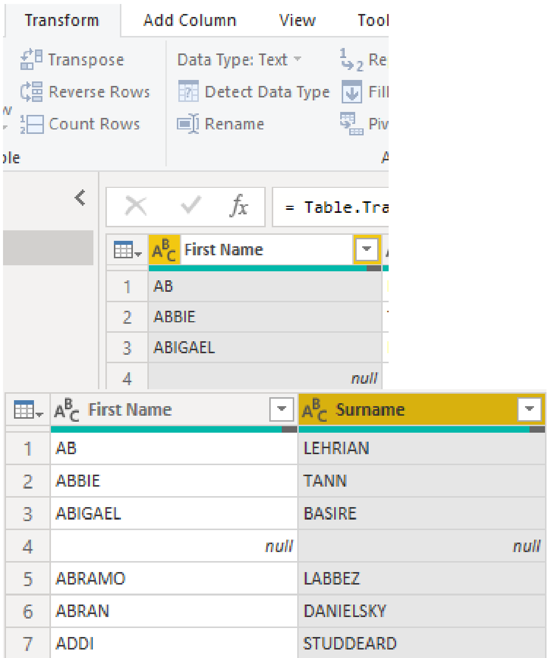
>Go to “advanced options” and enter number of columns to split into





	Name.1	Name.2
1	AB	LEHRIAN
2	ABBIE	TANN
3	ABIGAEL	BASIRE
4	null	null
5	ABRAMO	LABBEZ
6	ABRAN	DANIELSKY
7	ADDI	STUDDEARD
8	ADDIA	PENWRIGHT
9	ADDY	PIMBLETT
10	ADELA	DOWSETT
11	ADELINA	CHEESEMAN
12	ADELLA	HARTSHORNE
13	ADEY	RYAL
14	ADI	SEAWRIGHT
15	ADOLPH	HARTIN
16	ADOLPH	MCNALLEY
17	ADRIANNE	GAVE
18	AERIELA	AICKIN

12. Rename the Name1 column in to First name and Name 2 as Surname.
>Select “name.1” column. Go to “rename” option in “transform” tab and enter the value. Perform the same for “name.2” column

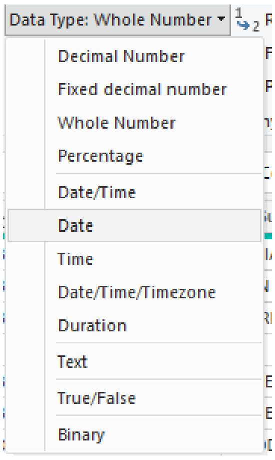


	First Name	Surname
1	AB	LEHRIAN
2	ABBIE	TANN
3	ABIGAEL	BASIRE
4	null	null
5	ABRAMO	LABBEZ
6	ABRAN	DANIELSKY
7	ADDI	STUDDEARD

13. Change the data type of salary as decimal number.

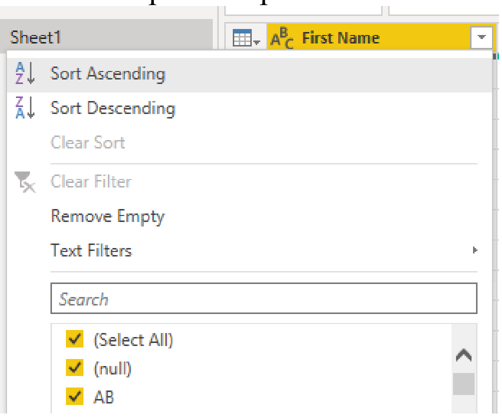
14. Change the data type of Start date column as date.

>Select “Start date” column. Go to transform tab option “data type” and select “data” option

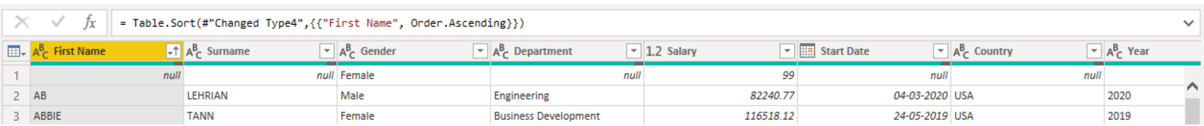


15. Sort the data in the ascending order of Name

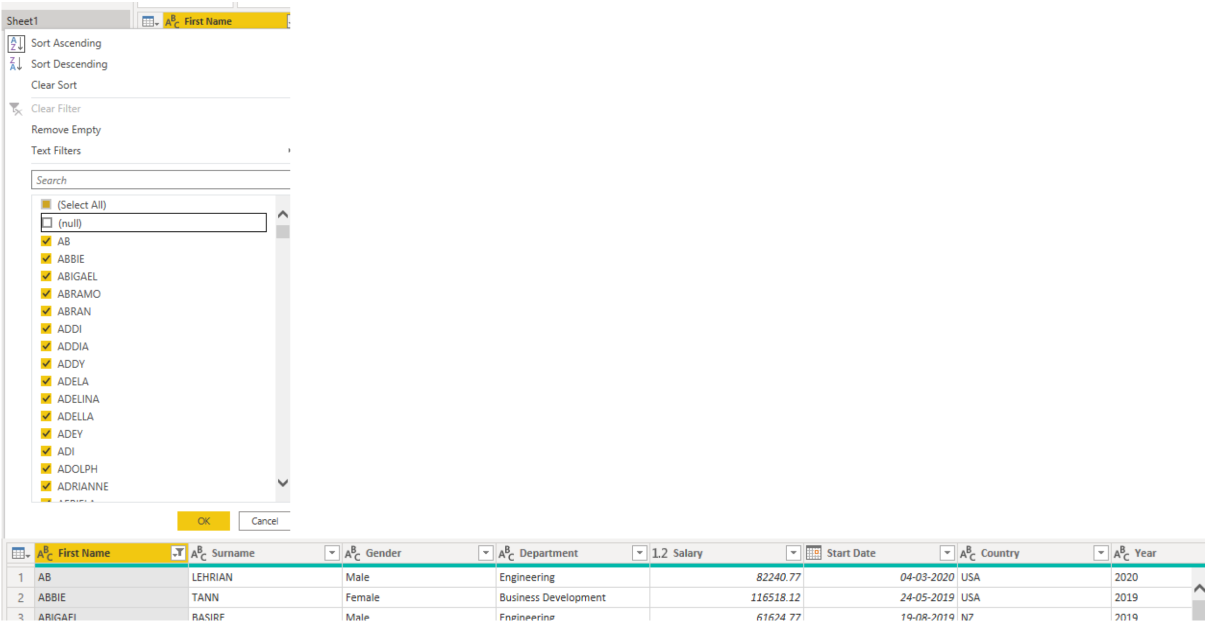
>Select dropdown option beside "name" column and click on "sort ascending" option



16. Remove the rows with null in name column.

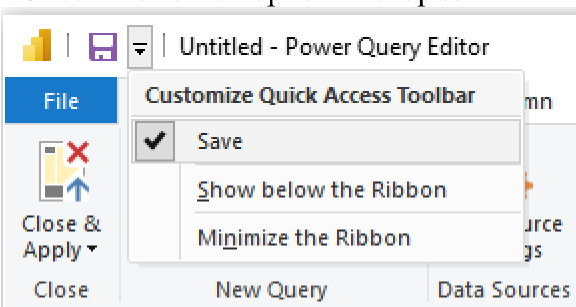


>Select "first name" column and go to dropdown option besides it and uncheck "null" option



17. Save the file.

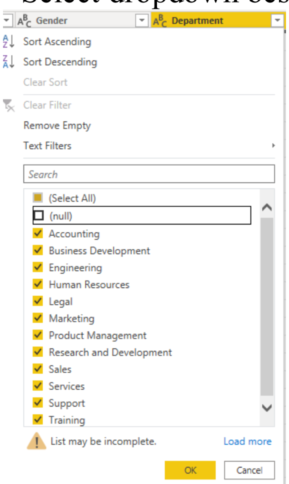
>Check the "save" option in dropdown



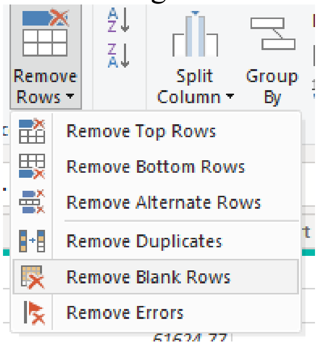
18. Remove the staff with department as null.

Two ways to do:

>Select dropdown besides "Department" column and uncheck the "null" option



>In the tab go to “remove rows” option and select “remove blank rows”



Practical 8	
Aim: To use the data in Microsoft Excel and create the Pivot table and Pivot Chart.	
Name: Labhesh Joshi	Roll No: KCTBCS030
Performance date: __/__/2022	Sign:

Theory:

OLAP Cube

Online Analytical Processing (OLAP) is a category of software that allows users to analyze information from multiple database systems at the same time. It is a technology that enables analysts to extract and view business data from different points of view.

Analysts frequently need to group, aggregate and join data. These OLAP operations in data mining are resource intensive. With OLAP data can be pre-calculated and pre-aggregated, making analysis faster.
OLAP databases are divided into one or more cubes. The cubes are designed in such a way that creating and viewing reports become easy. OLAP stands for Online Analytical Processing.

Different operations done on OLAP cube.

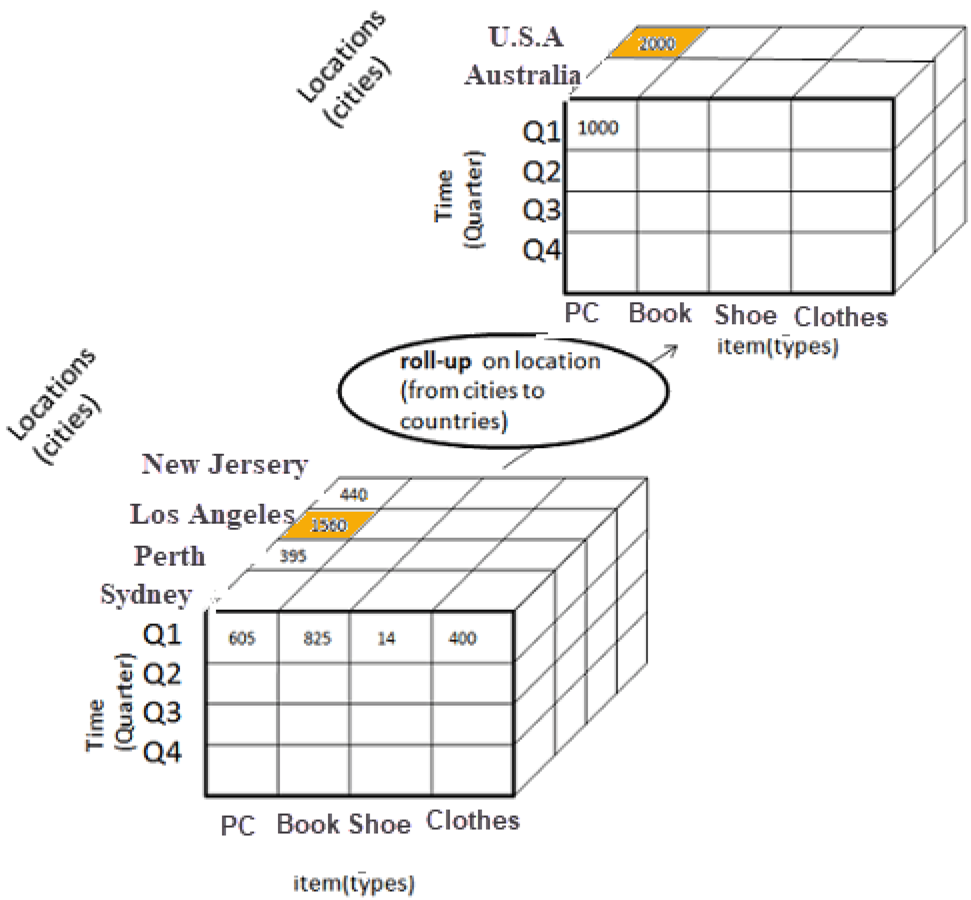
Four types of analytical OLAP operations are:

1. Roll-up
2. Drill-down
3. Slice and dice
4. Pivot (rotate)

Roll-up is also known as “consolidation” or “aggregation.” The Roll-up operation can be performed in 2 ways

1. Reducing dimensions
2. Climbing up concept hierarchy. Concept hierarchy is a system of grouping things based on their order or level.

Consider the following diagram



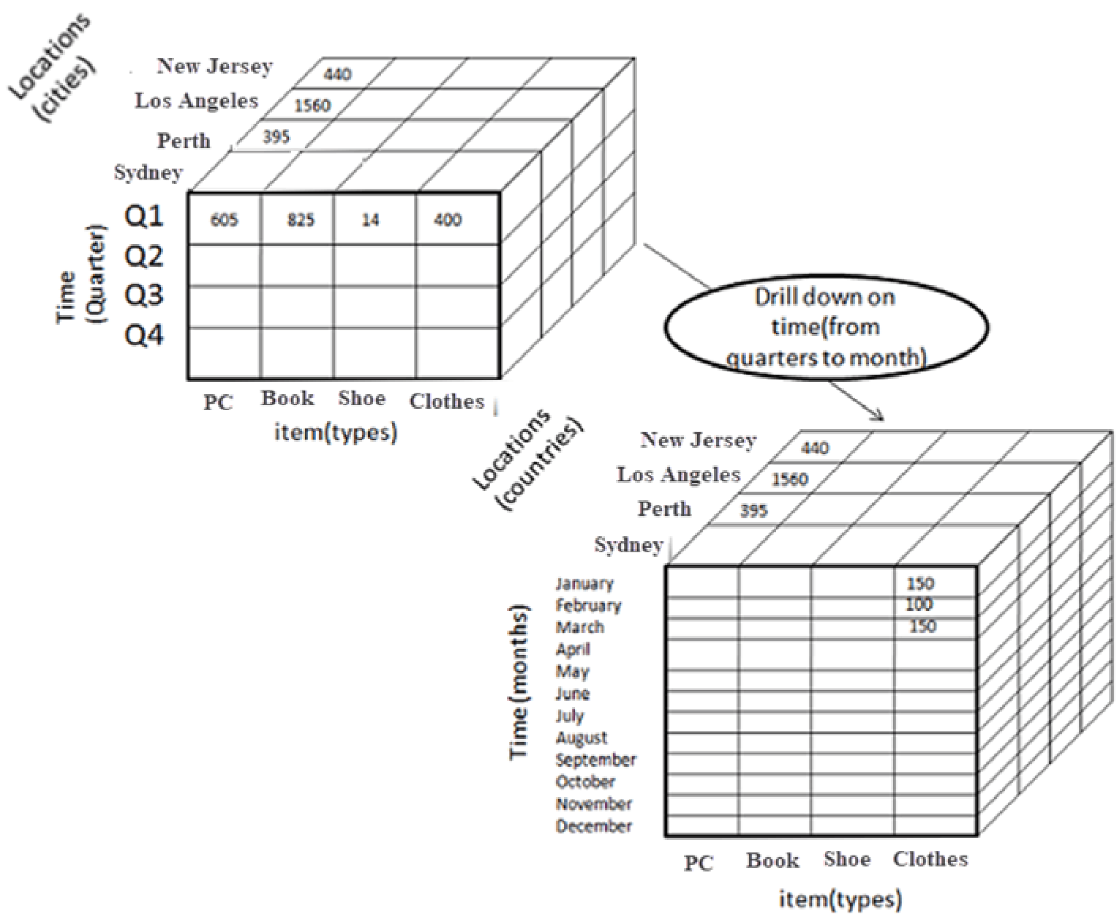
In this example, cities New jersey and Lost Angles and rolled up into country USA

- The sales figure of New Jersey and Los Angeles are 440 and 1560 respectively. They become 2000 after roll-up
- In this aggregation process, data is location hierarchy moves up from city to the country.
- In the roll-up process at least one or more dimensions need to be removed. In this example, Cities dimension is removed.

2) Drill-down

In drill-down data is fragmented into smaller parts. It is the opposite of the rollup process. It can be done via

- Moving down the concept hierarchy
- Increasing a dimension



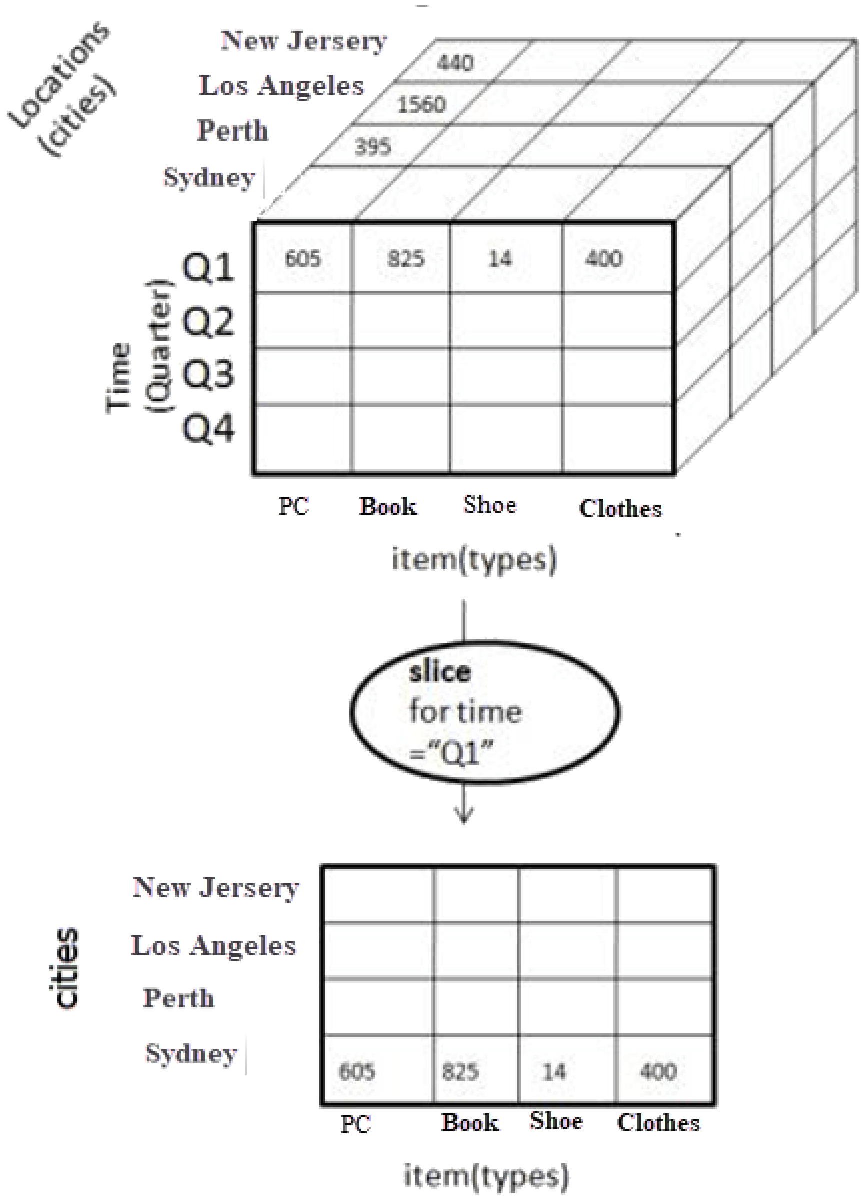
Consider the diagram above

- Quater Q1 is drilled down to months January, February, and March. Corresponding sales are also registers.
- In this example, dimension months are added

3) Slice:

Here, one dimension is selected, and a new sub-cube is created.

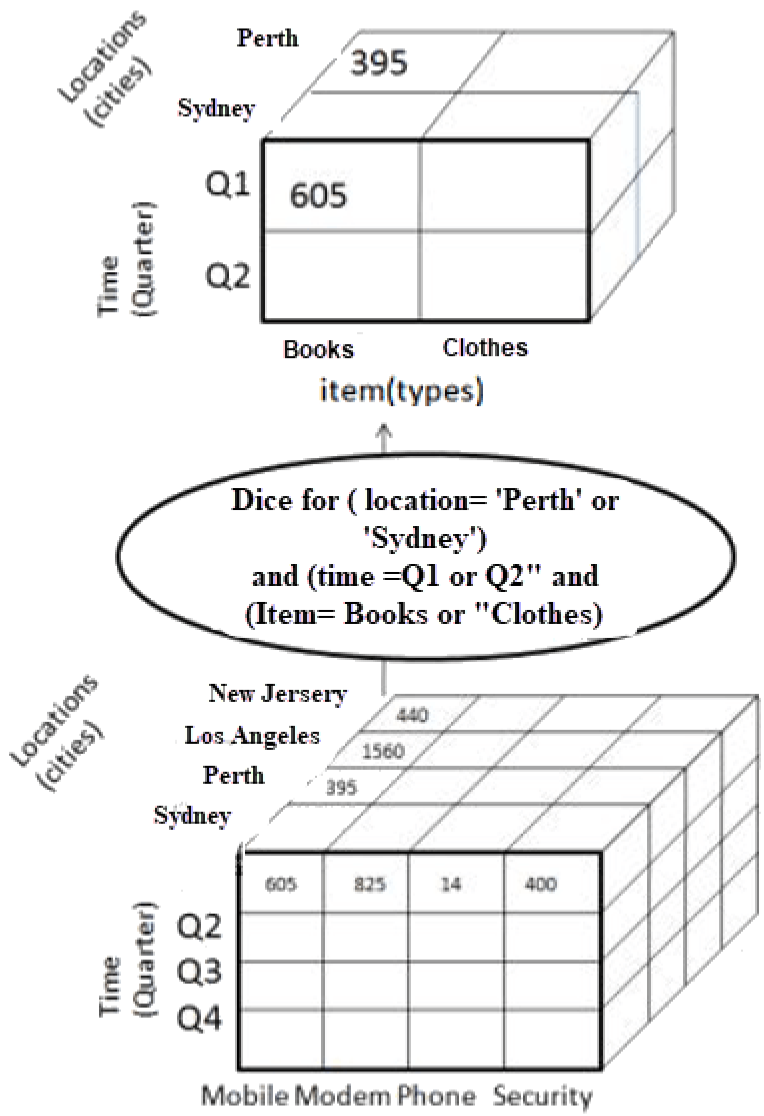
Following diagram explain how slice operation performed:



- Dimension Time is Sliced with Q1 as the filter.
- A new cube is created altogether.

Dice:

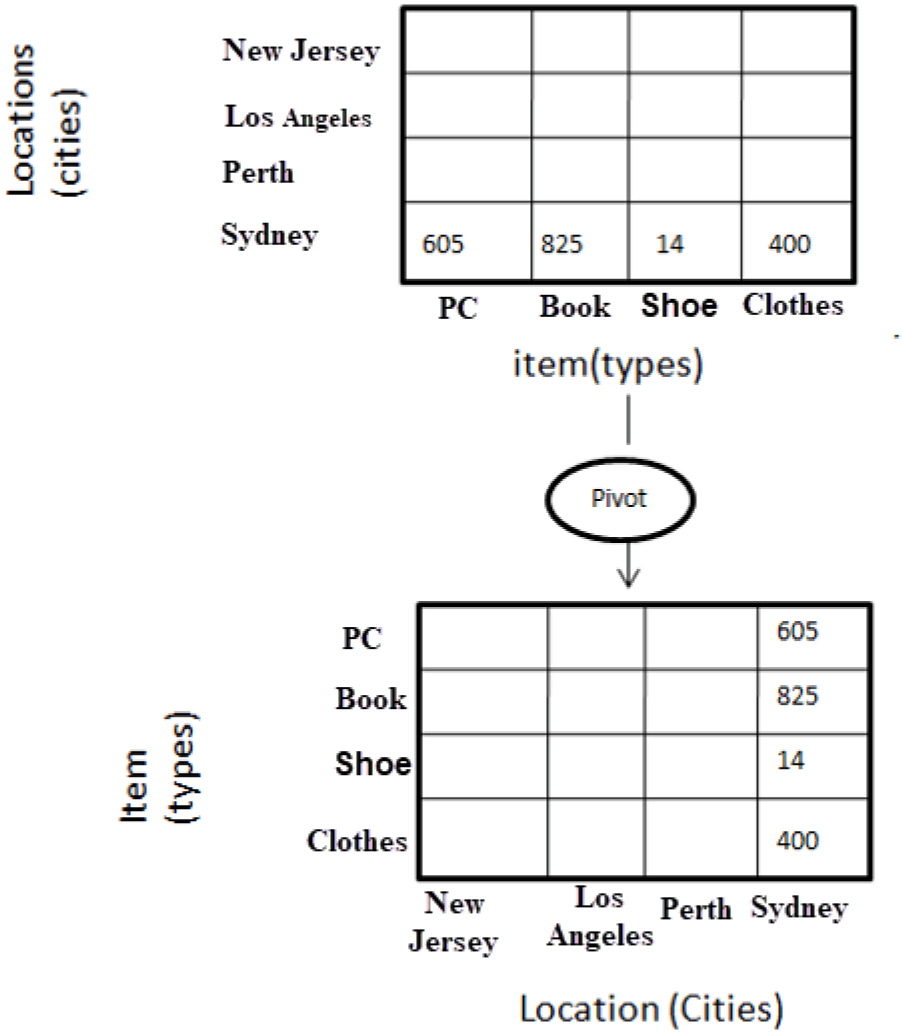
- This operation is similar to a slice. The difference in dice is you select 2 or more dimensions that result in the creation of a sub-cube.



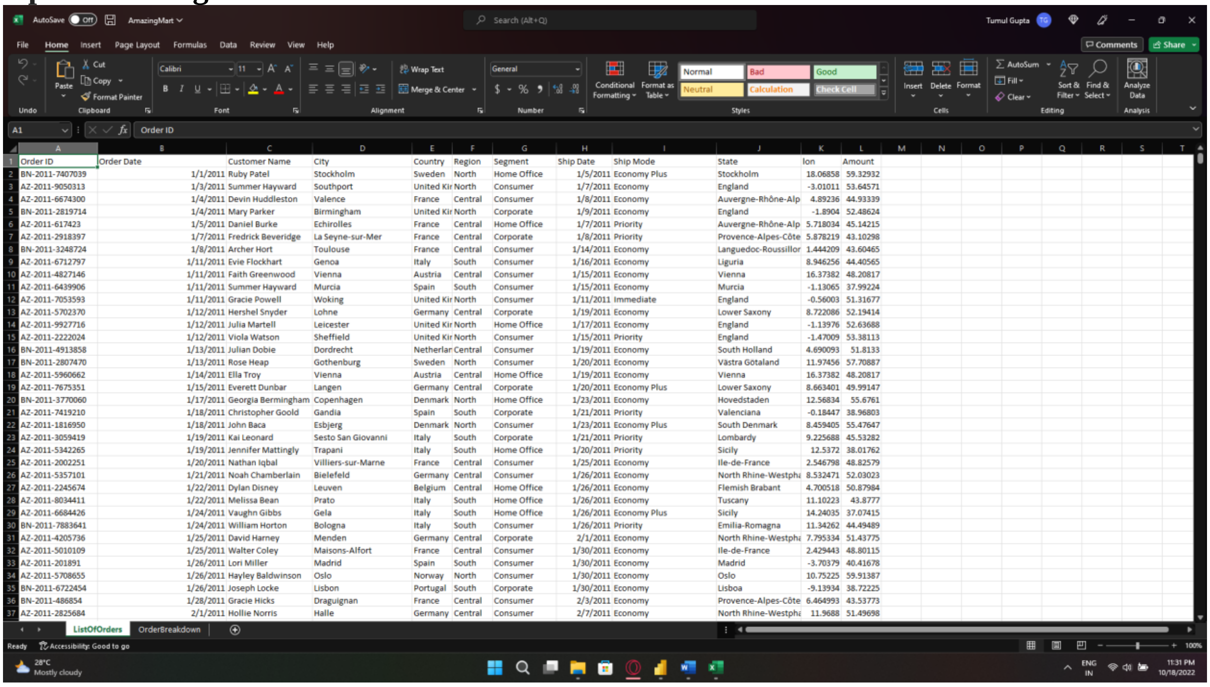
4) Pivot

In Pivot, you rotate the data axes to provide a substitute presentation of data.

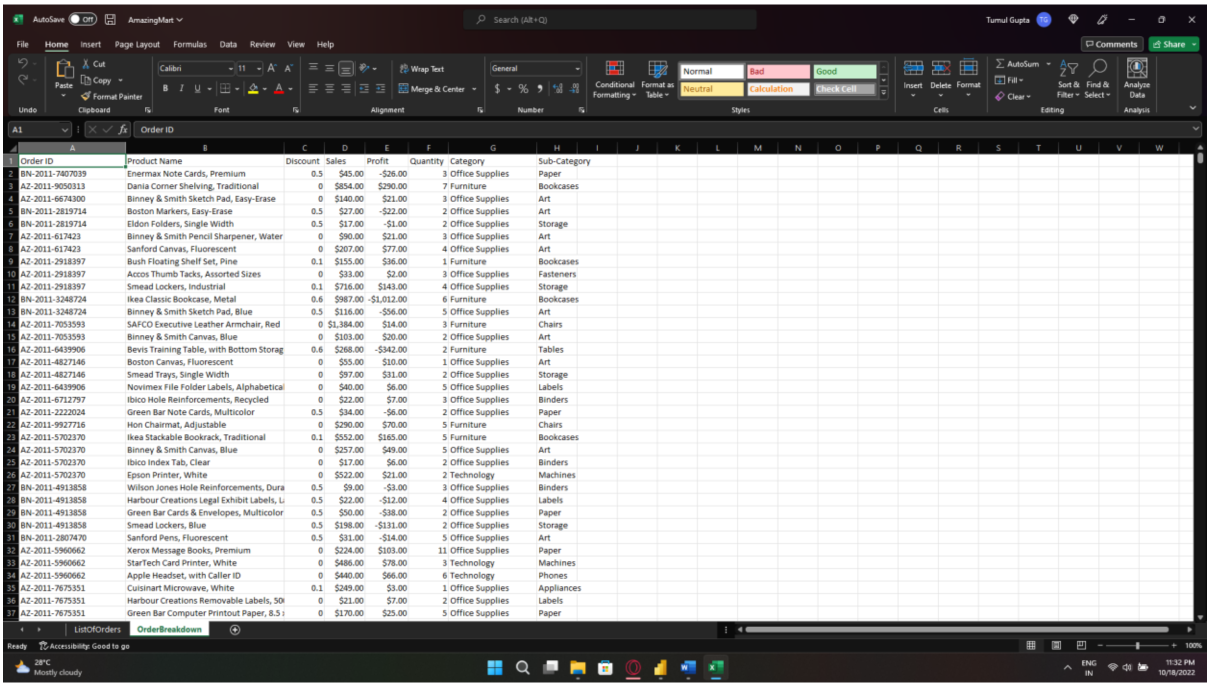
In the following example, the pivot is based on item types.



1. **Open AmazingMart.xlsx**



2. **Select OrderBreakdown Sheet**

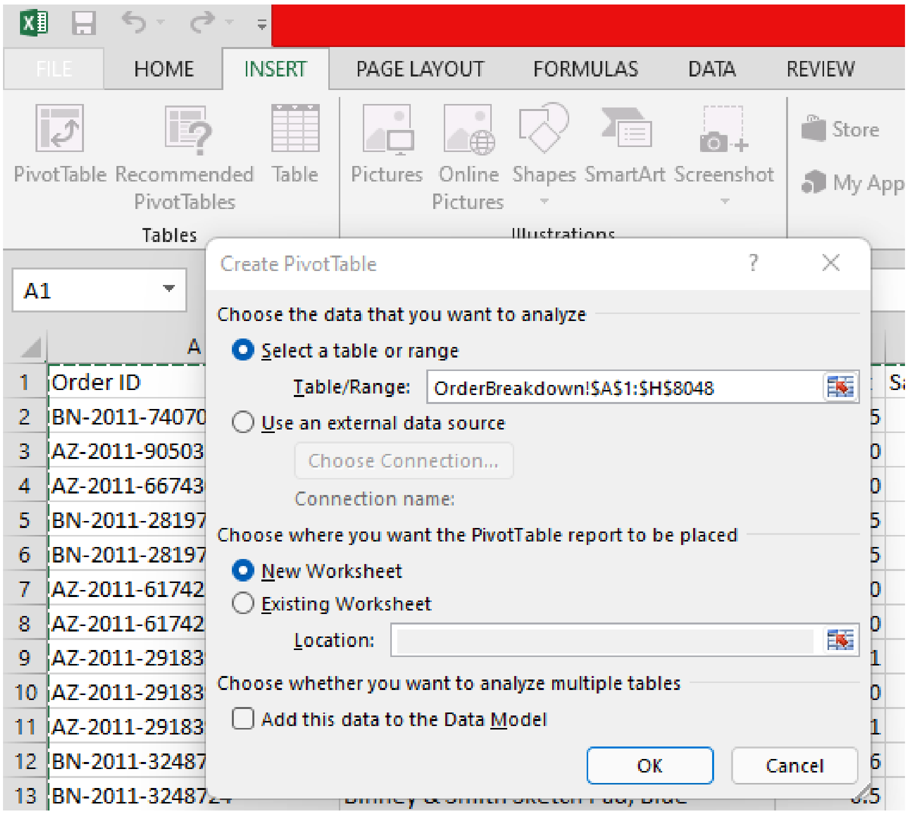


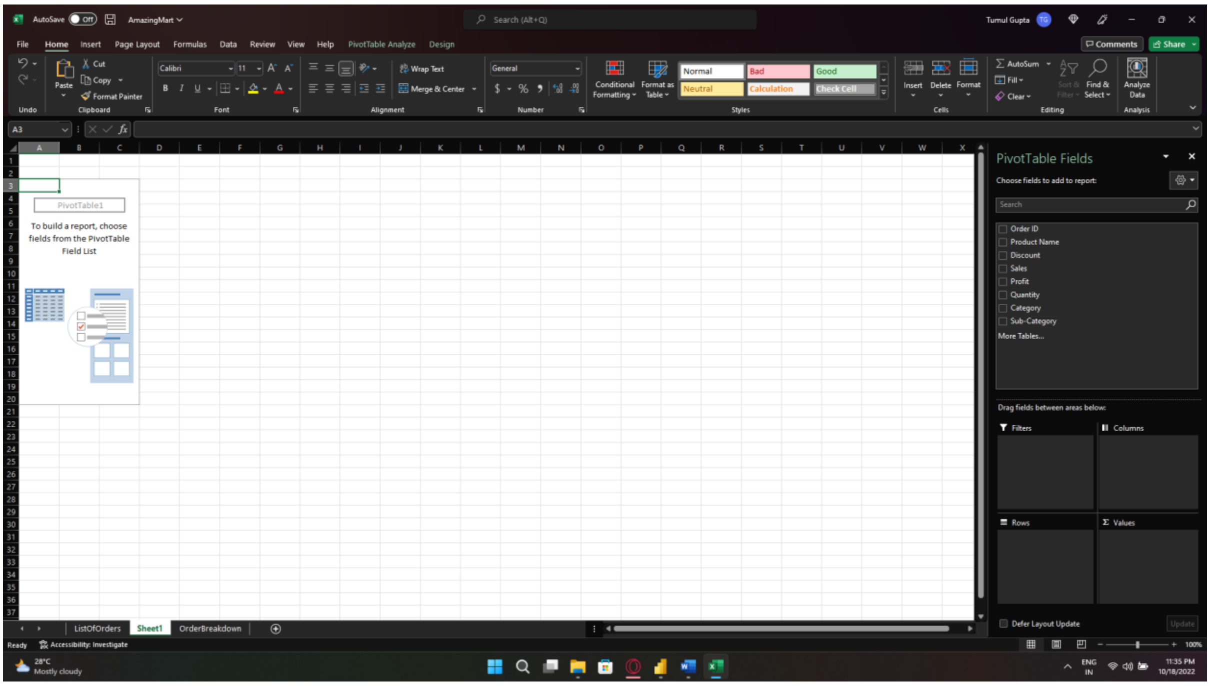
3. Select a Pivot Table in new worksheet

>Click on insert

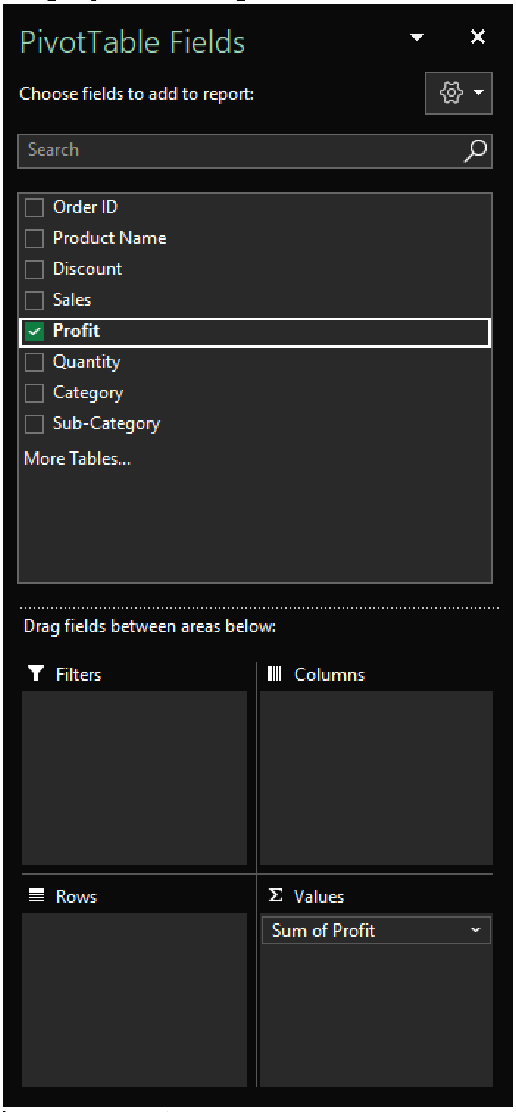
>Click on pivot tables

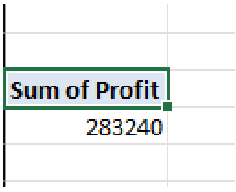
>Click OK



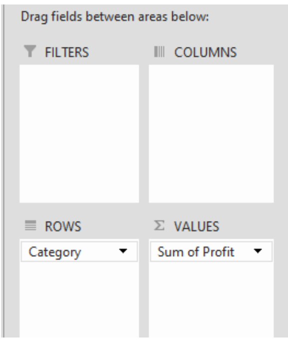


4. **Display the total profit incurred.**



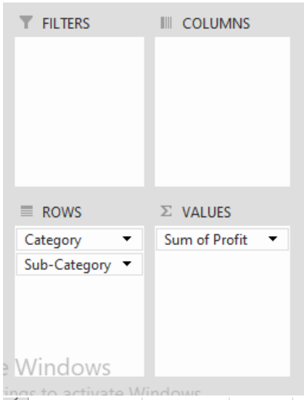


5. **Get the total Profit of each category of items.**



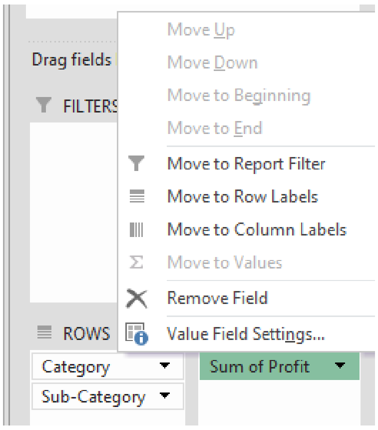
Row Labels	Sum of Profit
Furniture	49734
Office Supplies	124952
Technology	108554
Grand Total	**283240**

6. **Display the total Profit distribution as per every subcategory inside category of items.**

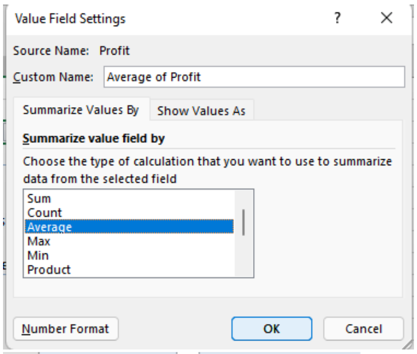


Row Labels	Sum of Profit
⊟Furniture	**49734**
Bookcases	43655
Chairs	15489
Furnishings	11321
Tables	-20731
⊟Office Supplies	**124952**
Appliances	37906
Art	23491
Binders	14703
Envelopes	6463
Fasteners	3420
Labels	2636
Paper	7485
Storage	21995
Supplies	6853
⊟Technology	**108554**
Accessories	26830
Copiers	42775
Machines	11318
Phones	27631
Grand Total	**283240**

7. **Display the AVERAGE Profit distribution as per every subcategory inside category of items.**

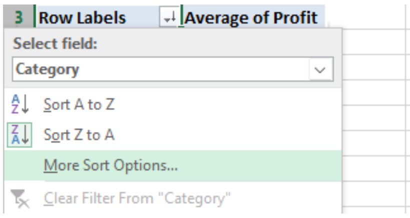


>Go to value Field Setting
>Select Average
>Click OK

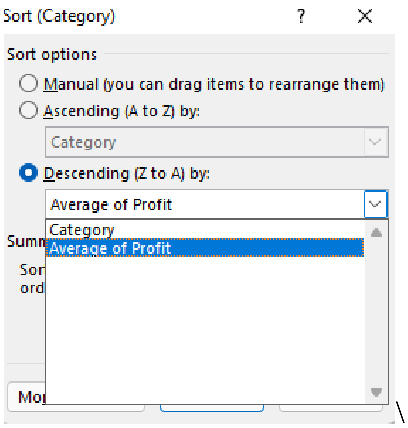


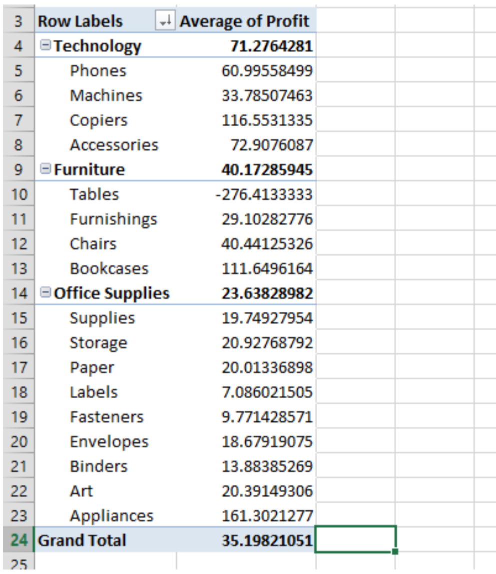
	Row Labels	Average of Profit
3	Row Labels	Average of Profit
4	**Furniture**	**40.17285945**
5	Bookcases	111.6496164
6	Chairs	40.44125326
7	Furnishings	29.10282776
8	Tables	-276.4133333
9	**Office Supplies**	**23.63828982**
10	Appliances	161.3021277
11	Art	20.39149306
12	Binders	13.88385269
13	Envelopes	18.67919075
14	Fasteners	9.771428571
15	Labels	7.086021505
16	Paper	20.01336898
17	Storage	20.92768792
18	Supplies	19.74927954
19	**Technology**	**71.2764281**
20	Accessories	72.9076087
21	Copiers	116.5531335
22	Machines	33.78507463
23	Phones	60.99558499
24	**Grand Total**	**35.19821051**

8. Sort the Profit in descending order.



>Go to more settings
>Select Average of Profit
>Click OK

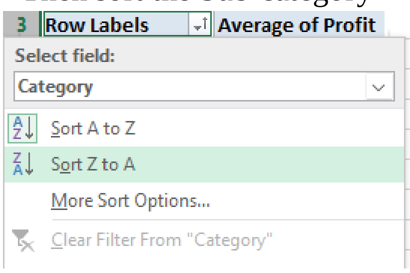


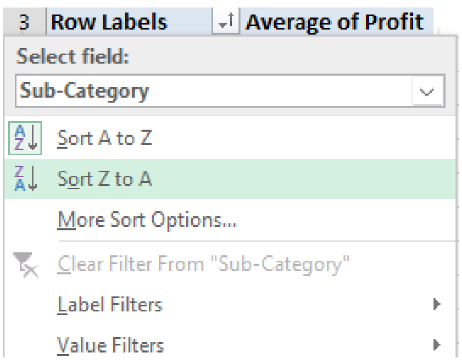


	Row Labels	Average of Profit
3	Row Labels	Average of Profit
4	**Technology**	**71.2764281**
5	Phones	60.99558499
6	Machines	33.78507463
7	Copiers	116.5531335
8	Accessories	72.9076087
9	**Furniture**	**40.17285945**
10	Tables	-276.4133333
11	Furnishings	29.10282776
12	Chairs	40.44125326
13	Bookcases	111.6496164
14	**Office Supplies**	**23.63828982**
15	Supplies	19.74927954
16	Storage	20.92768792
17	Paper	20.01336898
18	Labels	7.086021505
19	Fasteners	9.771428571
20	Envelopes	18.67919075
21	Binders	13.88385269
22	Art	20.39149306
23	Appliances	161.3021277
24	**Grand Total**	**35.19821051**
25		

9. Sort the Category and subcategory in the descending order.

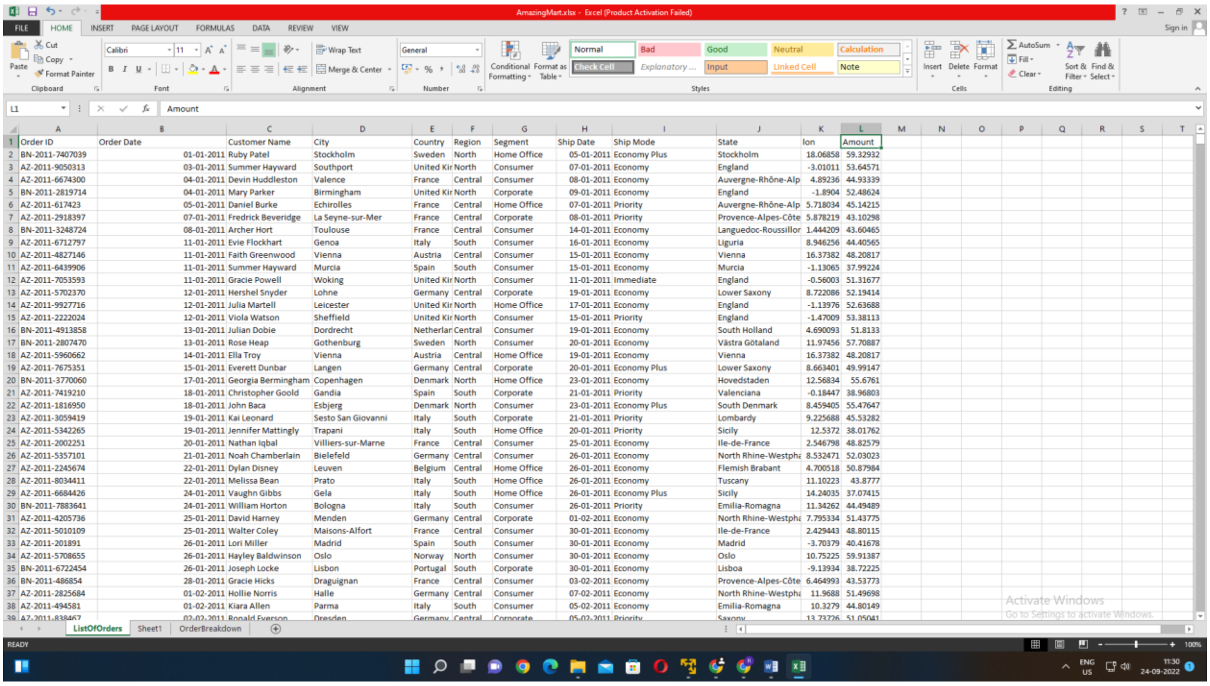
>First sort the Category
>Then sort the Sub-category





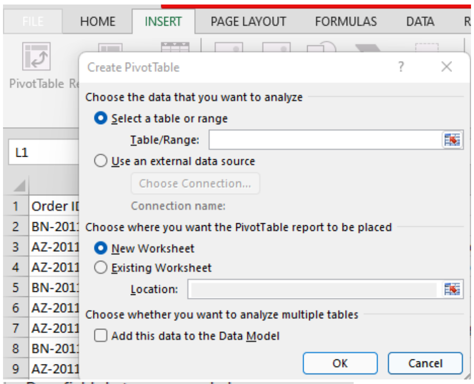
	Row Labels	Average of Profit
3	Row Labels	Average of Profit
4	Technology	71.2764281
5	Phones	60.99558499
6	Machines	33.78507463
7	Copiers	116.5531335
8	Accessories	72.9076087
9	Office Supplies	23.63828982
10	Supplies	19.74927954
11	Storage	20.92768792
12	Paper	20.01336898
13	Labels	7.086021505
14	Fasteners	9.771428571
15	Envelopes	18.67919075
16	Binders	13.88385269
17	Art	20.39149306
18	Appliances	161.3021277
19	Furniture	40.17285945
20	Tables	-276.4133333
21	Furnishings	29.10282776
22	Chairs	40.44125326
23	Bookcases	111.6496164
24	Grand Total	35.19821051

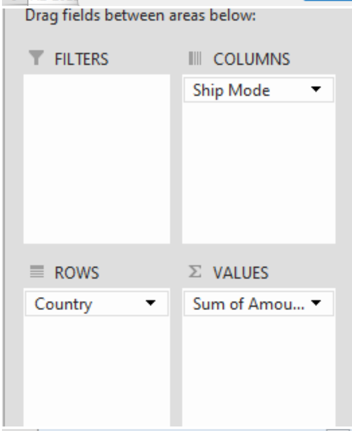
10. Use ListOfOrder sheet



11. Insert 2 dimensional Pivot table such that countries are available along rows, shipping mode is available in columns along with total amount as values.

>Create Pivot table of Of ListOfOrders

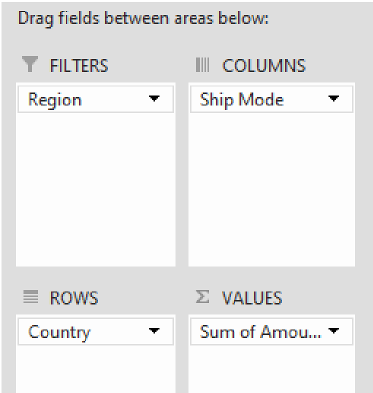


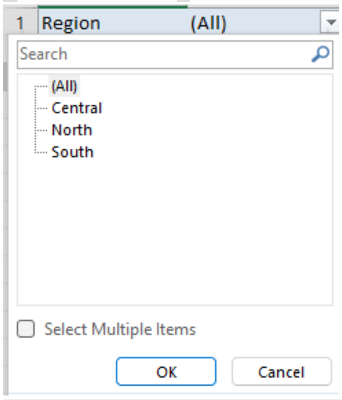


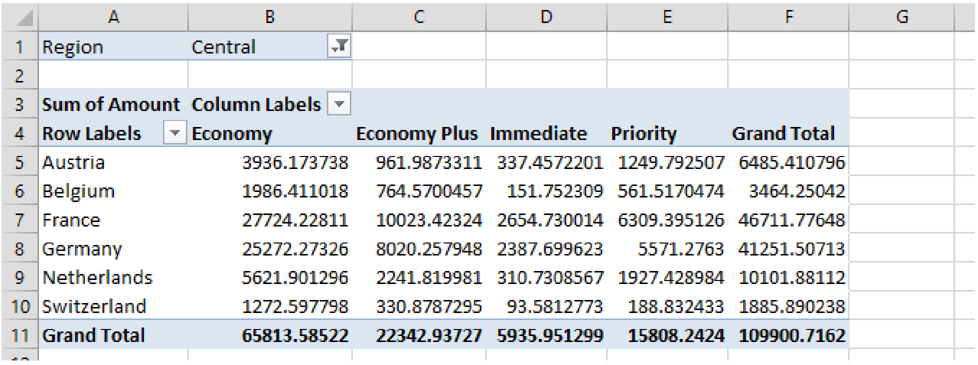
Sum of Amount	Column Labels				
Row Labels	Economy	Economy Plus	Immediate	Priority	Grand Total
Austria	3936.173738	961.9873311	337.4572201	1249.792507	6485.410796
Belgium	1986.411018	764.5700457	151.752309	561.5170474	3464.25042
Denmark	1055.123929	334.6433816		222.4433508	1612.210662
Finland	1144.794338	422.018942	301.2694861	180.5452025	2048.627969
France	27724.22811	10023.42324	2654.730014	6309.395126	46711.77648
Germany	25272.27326	8020.257948	2387.699623	5571.2763	41251.50713
Ireland	1327.743833	530.5922258	158.5965023	637.1494841	2654.082045
Italy	12547.09283	4308.825175	857.0167917	3382.053913	21094.98871
Netherlands	5621.901296	2241.819981	310.7308567	1927.428984	10101.88112
Norway	1255.498113	416.2788899	58.1599119	477.9747228	2207.911638
Portugal	1110.720565	156.349317		200.8842683	1467.95415
Spain	9510.68146	3192.716654	1010.230328	2467.67563	16181.30407
Sweden	3475.018471	1238.158621	355.9558046	822.3994349	5891.532331
Switzerland	1272.597798	330.8787295	93.5812773	188.832433	1885.890238
United Kingdom	21394.73793	7852.09679	1889.986165	5748.165372	36884.98625
Grand Total	**118634.9967**	**40794.61727**	**10567.16629**	**29947.53377**	**199944.314**

12. Apply the filter for central region.

>Add Regions to Filter
>Click on ALL
>Select Central







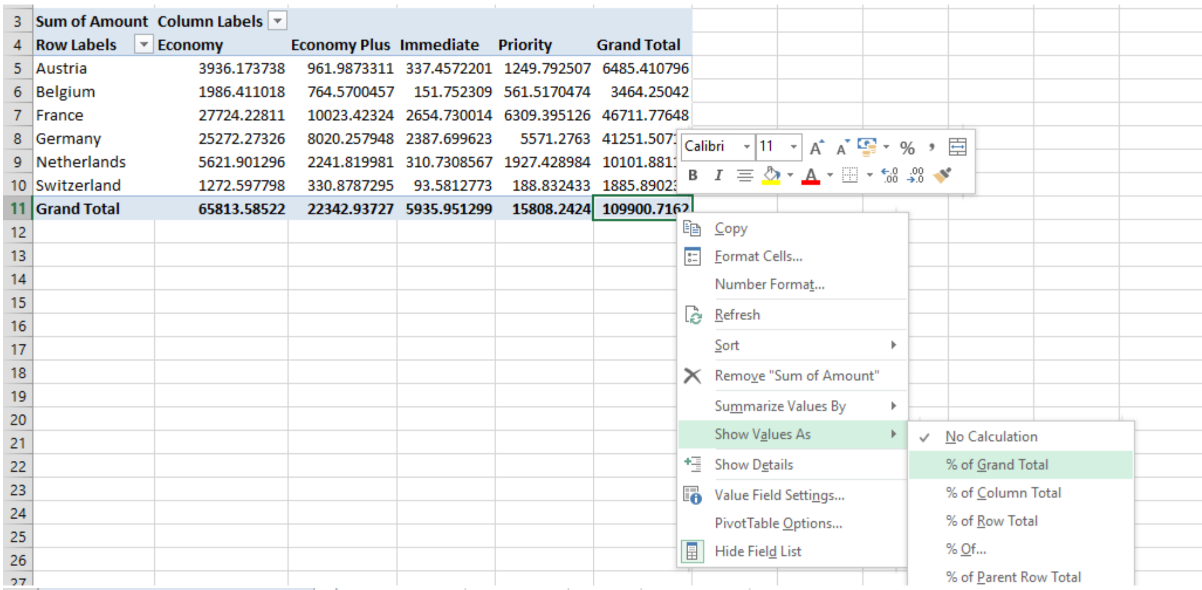
	A	B	C	D	E	F	G
1	Region	Central					
2							
3	**Sum of Amount**	**Column Labels**					
4	**Row Labels**	**Economy**	**Economy Plus**	**Immediate**	**Priority**	**Grand Total**	
5	Austria	3936.173738	961.9873311	337.4572201	1249.792507	6485.410796	
6	Belgium	1986.411018	764.5700457	151.752309	561.5170474	3464.25042	
7	France	27724.22811	10023.42324	2654.730014	6309.395126	46711.77648	
8	Germany	25272.27326	8020.257948	2387.699623	5571.2763	41251.50713	
9	Netherlands	5621.901296	2241.819981	310.7308567	1927.428984	10101.88112	
10	Switzerland	1272.597798	330.8787295	93.5812773	188.832433	1885.890238	
11	**Grand Total**	**65813.58522**	**22342.93727**	**5935.951299**	**15808.2424**	**109900.7162**	

13. Display the data in the form of percentage of Grand total.

>Right click on total cell

>Select Show value as

>Click % of Grand Total



Sum of Amount	Column Labels				
Row Labels	Economy	Economy Plus	Immediate	Priority	Grand Total
Austria	3936.173738	961.9873311	337.4572201	1249.792507	6485.410796
Belgium	1986.411018	764.5700457	151.752309	561.5170474	3464.25042
France	27724.22811	10023.42324	2654.730014	6309.395126	46711.77648
Germany	25272.27326	8020.257948	2387.699623	5571.2763	41251.507
Netherlands	5621.901296	2241.819981	310.7308567	1927.428984	10101.881
Switzerland	1272.597798	330.8787295	93.5812773	188.832433	1885.8902
Grand Total	**65813.58522**	**22342.93727**	**5935.951299**	**15808.2424**	**109900.7162**

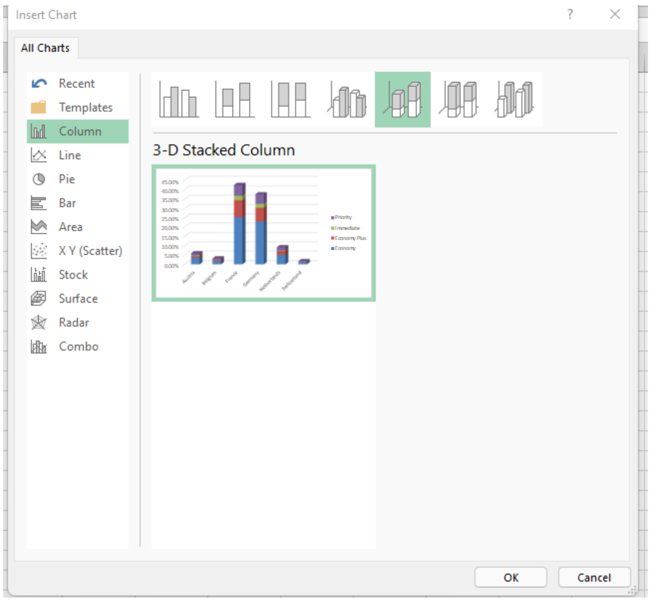
Region	Central				
Sum of Amount	**Column Labels**				
Row Labels	**Economy**	**Economy Plus**	**Immediate**	**Priority**	**Grand Total**
Austria	3.58%	0.88%	0.31%	1.14%	5.90%
Belgium	1.81%	0.70%	0.14%	0.51%	3.15%
France	25.23%	9.12%	2.42%	5.74%	42.50%
Germany	23.00%	7.30%	2.17%	5.07%	37.54%
Netherlands	5.12%	2.04%	0.28%	1.75%	9.19%
Switzerland	1.16%	0.30%	0.09%	0.17%	1.72%
Grand Total	**59.88%**	**20.33%**	**5.40%**	**14.38%**	**100.00%**

14. Insert a 3D stacked pyramid Pivot chart for the above table

>Glick on Pivot chart

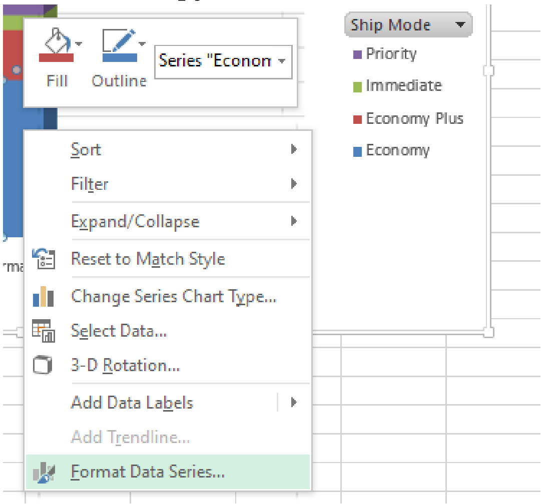
>Select Stacked 3-D

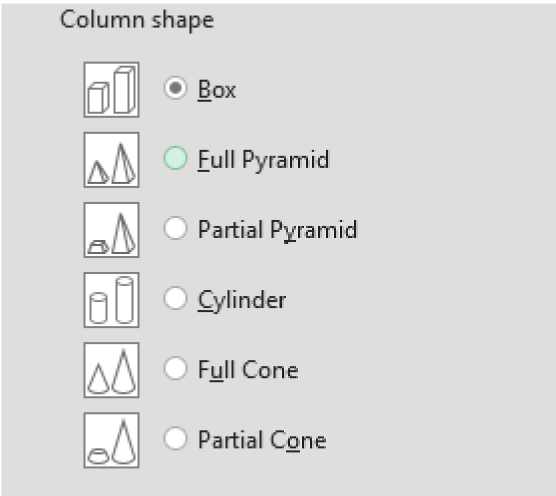
>Click OK

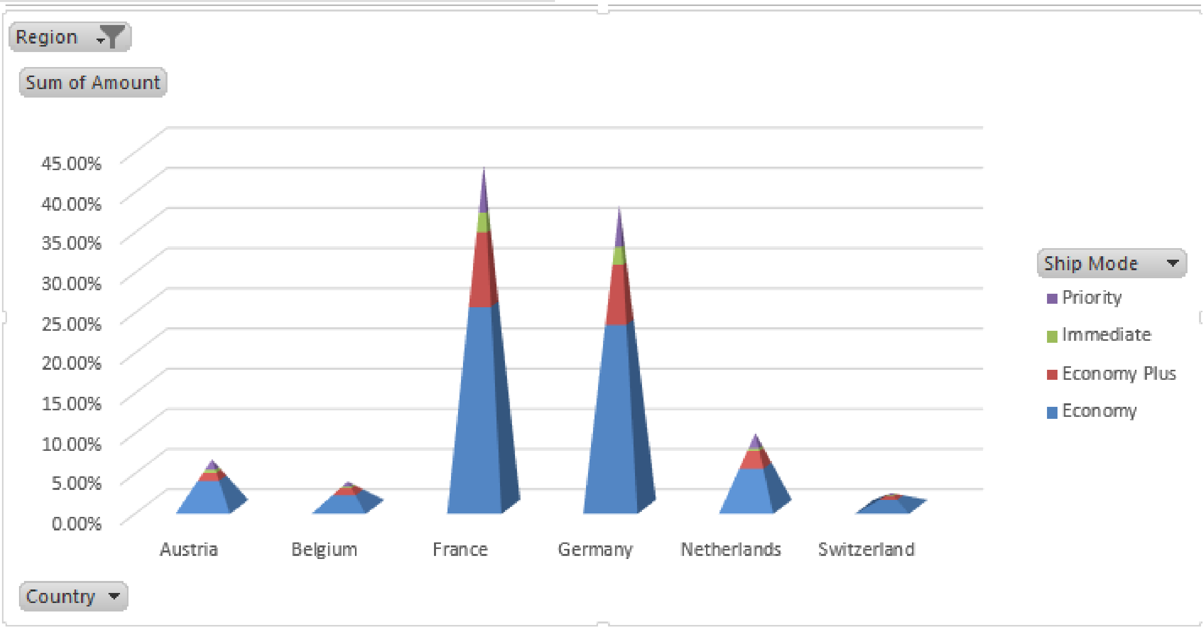


>Click on the chart and select Format data series

>Then select pyramid

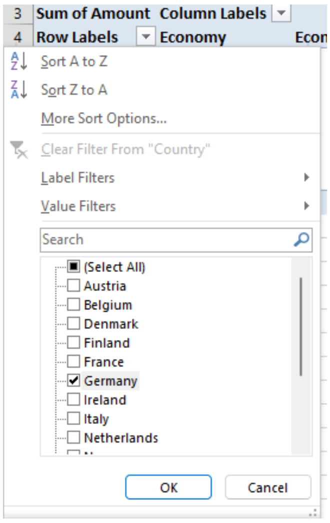


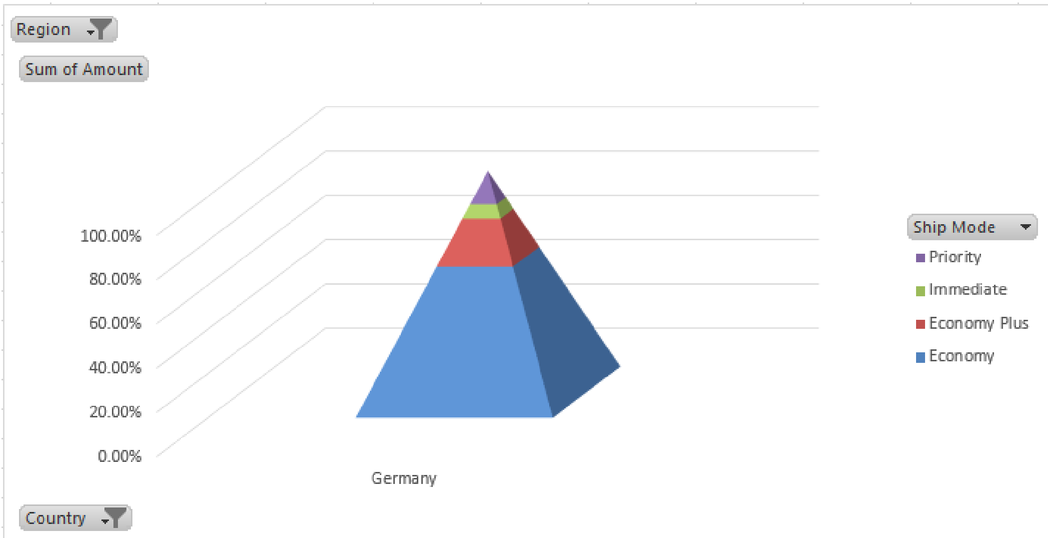




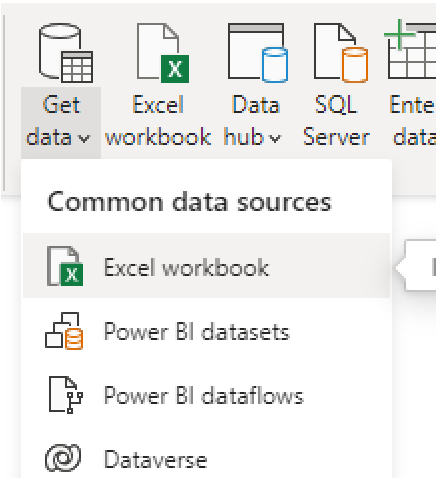
15. Display the data of only Germany.

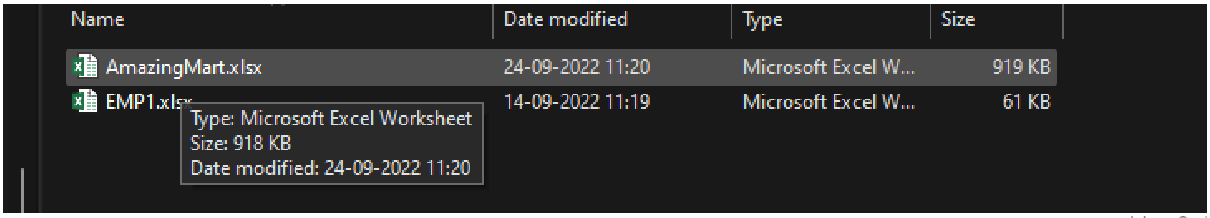
>Go to row labels and select only Germany

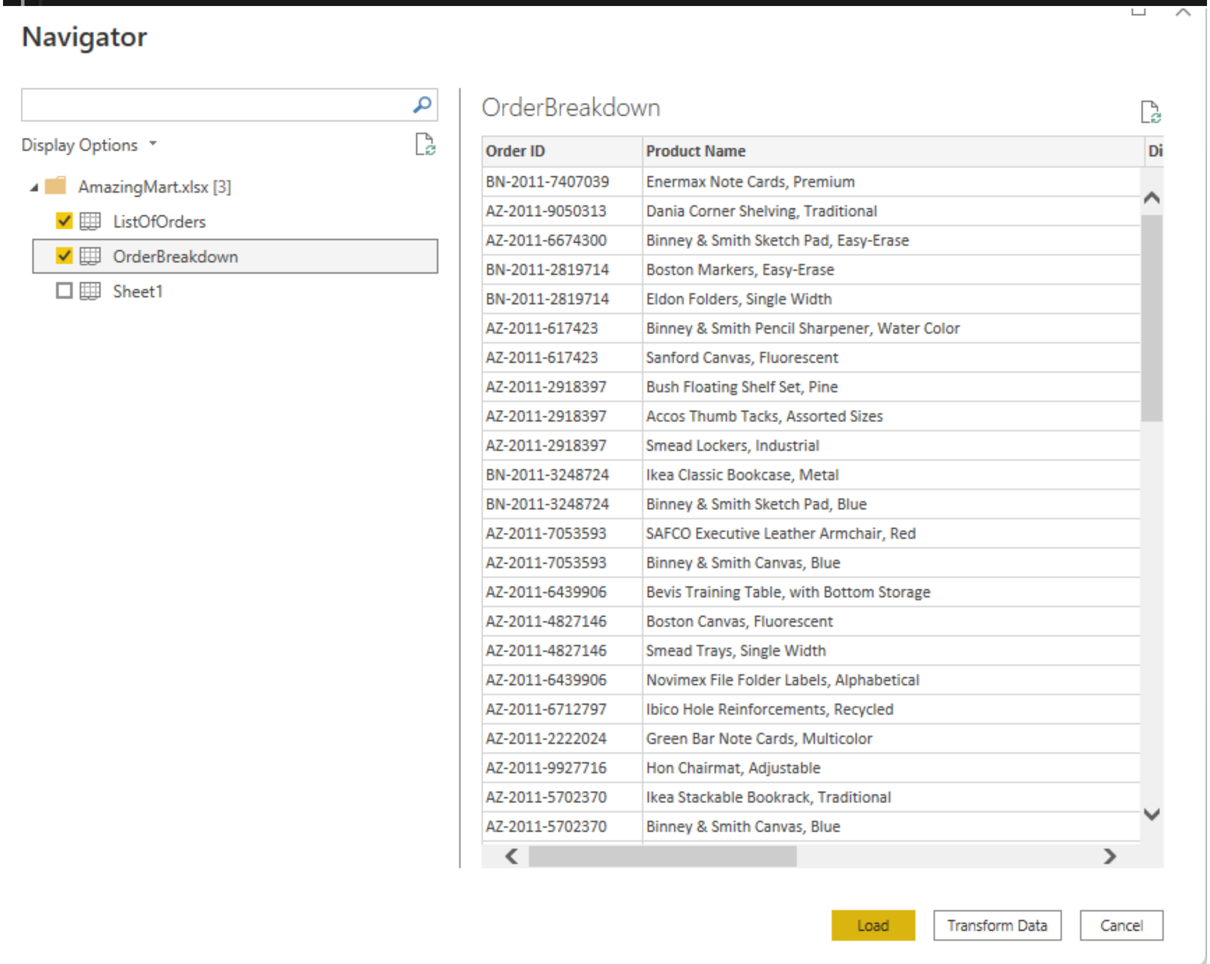


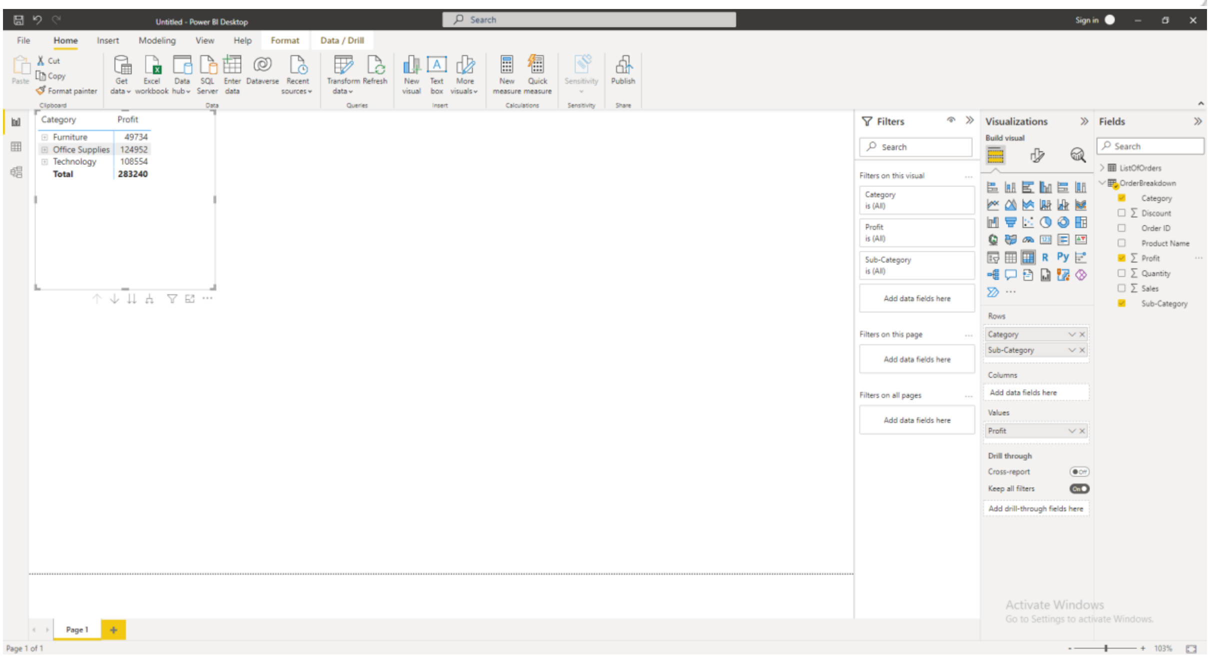


16. Create pivot table in Power Bi.




Name
Date modified
Type
Size
AmazingMart.xlsx
24-09-2022 11:20
Microsoft Excel W...
919 KB
EMP1.xlsx
14-09-2022 11:19
Microsoft Excel W...
61 KB
Type: Microsoft Excel Worksheet
Size: 918 KB
Date modified: 24-09-2022 11:20


Navigator
Display Options
AmazingMart.xlsx [3]
ListOfOrders
OrderBreakdown
Sheet1
OrderBreakdown
Order ID
Product Name
BN-2011-7407039 Enermax Note Cards, Premium
AZ-2011-9050313 Dania Corner Shelving, Traditional
AZ-2011-6674300 Binney & Smith Sketch Pad, Easy-Erase
BN-2011-2819714 Boston Markers, Easy-Erase
BN-2011-2819714 Eldon Folders, Single Width
AZ-2011-617423 Binney & Smith Pencil Sharpener, Water Color
AZ-2011-617423 Sanford Canvas, Fluorescent
AZ-2011-2918397 Bush Floating Shelf Set, Pine
AZ-2011-2918397 Accos Thumb Tacks, Assorted Sizes
AZ-2011-2918397 Smead Lockers, Industrial
BN-2011-3248724 Ikea Classic Bookcase, Metal
BN-2011-3248724 Binney & Smith Sketch Pad, Blue
AZ-2011-7053593 SAFCO Executive Leather Armchair, Red
AZ-2011-7053593 Binney & Smith Canvas, Blue
AZ-2011-6439906 Bevis Training Table, with Bottom Storage
AZ-2011-4827146 Boston Canvas, Fluorescent
AZ-2011-4827146 Smead Trays, Single Width
AZ-2011-6439906 Novimex File Folder Labels, Alphabetical
AZ-2011-6712797 Ibico Hole Reinforcements, Recycled
AZ-2011-2222024 Green Bar Note Cards, Multicolor
AZ-2011-9927716 Hon Chairmat, Adjustable
AZ-2011-5702370 Ikea Stackable Bookrack, Traditional
AZ-2011-5702370 Binney & Smith Canvas, Blue
Load
Transform Data
Cancel


Untitled - Power BI Desktop
File
Home
Insert
Modeling
View
Help
Format
Data / Drill
Category
Profit
Furniture 49734
Office Supplies 124952
Technology 108554
Total 283240
Filters
Visualizations
Fields
ListOfOrders
OrderBreakdown
Category
Discount
Order ID
Product Name
Profit
Quantity
Sales
Sub-Category
Page 1