**GOVERNMENT COLLEGE OF ENGINEERING ERODE**

****

B.E Electronics and Communication Engineering

PRODUCT SALES ANALYSIS

**Name of the Student: University Register no:**

SURENDHAR S 731121106049

Under the mentor of

**Dr.M.Poongothai**

**Department of Information Technology(IT)**

**Department of Electronics and Communication Engineering**

Government College of Engineering

Erode ,PO ,near Vasavi College,TamilNadu-638316,

Affiliated to Anna University ,Chennai.

**DEFINITION OF MACHINE LEARNING :**

Machine learning is a subfield of artificial intelligence (AI) that focuses on the development of algorithms and models that enable computers to learn and make predictions or decisions without being explicitly programmed. Machine learning systems use data and statistical techniques to improve their performance on a specific task over time. The primary goal of machine learning is to develop algorithms that can generalize from data, allowing them to make accurate predictions or decisions on new, unseen data.

**LINEAR REGRESSION : [Supervised]**

Linear regression is one of the fundamental techniques in machine learning and statistics used for modeling the relationship between a dependent variable (target) and one or more independent variables (features or predictors). It is a type of supervised learning algorithm that aims to find the linear relationship between the features and the target variable. Linear regression is widely used for tasks such as predicting house prices, estimating sales figures, and understanding the relationship between variables in various fields.

**CLUSTERING : [Unsupervised]**

Clustering is a type of unsupervised machine learning technique used to group or cluster similar data points together based on their characteristics, without any predefined labels or categories. The goal of clustering is to find underlying structures or patterns within the data, making it easier to understand and interpret complex datasets.

**K - MEANS CLUSTERING :**

K means is one of the most popular Unsupervised Machine Learning Algorithms Used for Solving Classification Problems in data science and is very important if you are aiming for a data scientist role. K Means segregates the unlabeled data into various groups, called clusters, based on having similar features and common patterns. This tutorial will teach you the definition and applications of clustering, focusing on the K means clustering algorithm and its implementation in Python. It will also tell you how to choose the optimum number of clusters for a dataset.

**THE IMPLEMENTATION AND WORKING OF THE K-MEANS ALGORITHM :**

* **Step 1:** Select the value of K to decide the number of clusters (n\_clusters) to be formed.
* **Step 2:** Select random K points that will act as cluster centroids (cluster\_centers).
* **Step 3:** Assign each data point, based on their distance from the randomly selected points (Centroid), to the nearest/closest centroid, which will form the predefined clusters.
* **Step 4:** Place a new centroid of each cluster.
* **Step 5:** Repeat step no.3, which reassigns each datapoint to the new closest centroid of each cluster.
* **Step 6:** If any reassignment occurs, then go to step 4; else, go to step 7.
* **Step 7:** Finish

**ABOUT DATASET**

**CONTEXT :**

This dataset contain various details of products sold at a store. These type of datasets are studied to find out the patterns in the selling structure and profit earned from them.

**CONTENT :**

* Order\_ID : A specific ID given to each product
* Order\_Priority : Priority of the product
* Order\_Quantity: No of product items sold.
* Ship\_Mode: Divided in two categories - Express Air and Regular Air
* Profit: Profit earned from the sale
* Customer\_Name: Name of the customer purchasing the products
* Region: Region to which the customer belongs
* Customer\_Segment: Divided as per the size of business
* Product\_Category: Divided according to the usage of the product
* Product\_Sub-Category: Divided according to the usage of the product
* Product\_Name: Name of the product
* Product\_Container: Type of container in which the product is shipped.

**PYTHON IMPLEMENTATION:**

* **Importing relevant libraries**

import numpy as np

import pandas as pd

import statsmodels.api as sm

import matplotlib.pyplot as plt

import seaborn as sns

sns.set()

from sklearn.cluster import KMeans

* **Loading the data**

data = pd.read\_csv('Countryclusters.csv')

data

**PRE PROCESSING IN DATA ANALYTICS :**

Preprocessing is a crucial step in data analytics and data science. It involves a series of operations that are performed on raw data to clean, transform, and prepare it for analysis. The quality of the preprocessing phase significantly impacts the accuracy and effectiveness of any data analytics or machine learning project. Here are some common preprocessing steps in data analytics:

* **Data Collection:**

This is the initial step where data is gathered from various sources, such as databases, files, APIs, or sensors.

* **Handling Missing Values:**

Identify and deal with missing data, which can involve imputation (filling in missing values) or removal of incomplete records.

* **Outlier Detection:**

Identify and handle outliers, which are data points significantly different from the rest and can skew results.

* **Data Encoding:**

Convert categorical variables into numerical representations (e.g., one-hot encoding or label encoding).

* **Feature Scaling:**

Standardize or normalize numerical features to have a common scale (e.g., using z-score normalization or min-max scaling).

* **Data Integration:**

Combine data from multiple sources or datasets if necessary.

* **Data Splitting:**

Divide the dataset into training, validation, and test sets for model training, evaluation, and testing.

* **Data Visualization:**

Generate visualizations to explore data and gain insights into its characteristics, relationships, and potential issues.

* **Data Normalization:**

Ensure that data follows a normal distribution, if required, to meet assumptions of certain statistical tests or machine learning algorithms.

* **Data Scaling:**

Scale features to have similar ranges to avoid bias in some machine learning algorithms.

* **Text Preprocessing:**

For natural language processing (NLP) tasks, this might include tokenization, stemming, lemmatization, and removing stop words.

**EQUALITY AND DISTANCE METRICS :**

These are two different types of measures used to assess and quantify relationships, similarities, or differences between data points. These metrics serve distinct purposes in data analysis:

**1. EQUALITY METRICS:**

Equality metrics, also known as similarity metrics, assess the degree of likeness or similarity between data points. These metrics provide a measure of how much two data points resemble each other. Some commonly used equality metrics in data analytics include:

* **Jaccard Similarity:** This metric quantifies the similarity between two sets by measuring the size of their intersection relative to their union. It is often used in recommendation systems and text analysis.
* **Cosine Similarity:** Measures the cosine of the angle between two non-zero vectors, indicating how aligned they are in high-dimensional space. It is commonly used in natural language processing and recommendation systems.
* **Correlation Coefficient:** Measures the linear relationship between two variables. A value of 1 indicates a perfect positive correlation, 0 indicates no correlation, and -1 indicates a perfect negative correlation.
* **Hamming Distance:** Measures the number of differing elements between two binary strings or categorical variables. It is used for comparing sequences and categorical data.
* **Spearman's Rank Correlation**: Assesses the strength and direction of the monotonic relationship between two ranked variables. It is suitable for ordinal data and non-linear relationships.

2. **DISTANCE METRICS:**

Distance metrics, also known as dissimilarity metrics, quantify the difference or distance between data points. These metrics provide a measure of how far apart two data points are from each other. Some commonly used distance metrics in data analytics include:

* **Euclidean Distance:** Measures the straight-line distance between two points in Euclidean space. It is widely used in clustering, anomaly detection, and dimensionality reduction.
* **Manhattan Distance (L1 Distance):** Represents the distance as the sum of the absolute differences along each dimension. It is also known as the City Block distance.
* **Mahalanobis Distance:** Takes into account the correlations between dimensions, making it suitable for cases where data is not spherical or where correlations matter. It is used in clustering and anomaly detection.
* **Chebyshev Distance (L∞ Distance): Measures** the maximum difference along any dimension. It is useful when you want to find the greatest dissimilarity in any dimension.
* **Wasserstein Distance (Earth Mover's Distance):** Measures the minimum cost of transforming one distribution into another. It is commonly used in image processing and histogram comparison.
* **Kullback-Leibler Divergence (KL Divergence):** Measures the difference between two probability distributions. It is often used in information theory and statistical analysis.

**EXAMPLE SOURCE CODE :**

# Import necessary libraries

import pandas as pd

import numpy as np

from sklearn.cluster import KMeans

import matplotlib.pyplot as plt

# Load your product sales data (replace 'sales\_data.csv' with your data file)

data = pd.read\_csv('sales\_data.csv')

# Assuming you have columns like 'ProductID' and 'Sales' in your dataset

X = data[['ProductID', 'Sales']]

# Determine the optimal number of clusters (K) using the Elbow method

wcss = []

for i in range(1, 11):

kmeans = KMeans(n\_clusters=i, init='k-means++', max\_iter=300, n\_init=10, random\_state=0)

kmeans.fit(X)

wcss.append(kmeans.inertia\_)

# Plot the Elbow method to find the optimal number of clusters

plt.plot(range(1, 11), wcss)

plt.title('Elbow Method')

plt.xlabel('Number of clusters')

plt.ylabel('WCSS (Within-Cluster Sum of Squares)')

plt.show()

# Apply K-Means clustering

kmeans = KMeans(n\_clusters=k, init='k-means++', max\_iter=300, n\_init=10, random\_state=0)

pred\_y = kmeans.fit\_predict(X)

# Add the cluster labels to the dataset

data['Cluster'] = pred\_y

# Print and analyze the clusters

for cluster in range(k):

cluster\_data = data[data['Cluster'] == cluster]

print(f"Cluster {cluster + 1}:")

print(cluster\_data)

# Visualization of clusters (assuming you have matplotlib installed)

for cluster in range(k):

cluster\_data = data[data['Cluster'] == cluster]

plt.scatter(cluster\_data['ProductID'], cluster\_data['Sales'], label=f'Cluster {cluster + 1}')

plt.scatter(kmeans.cluster\_centers\_[:, 0], kmeans.cluster\_centers\_[:, 1], s=300, c='red', label='Centroids')

plt.title('Product Sales Clusters')

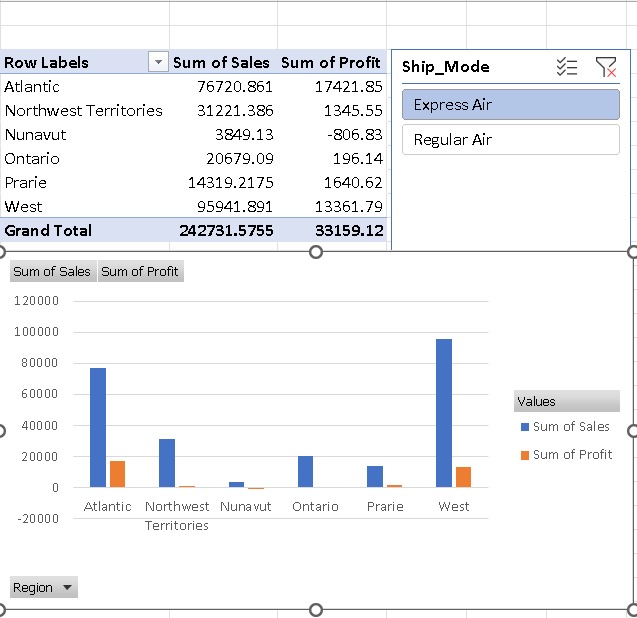
plt.xlabel('Product ID')

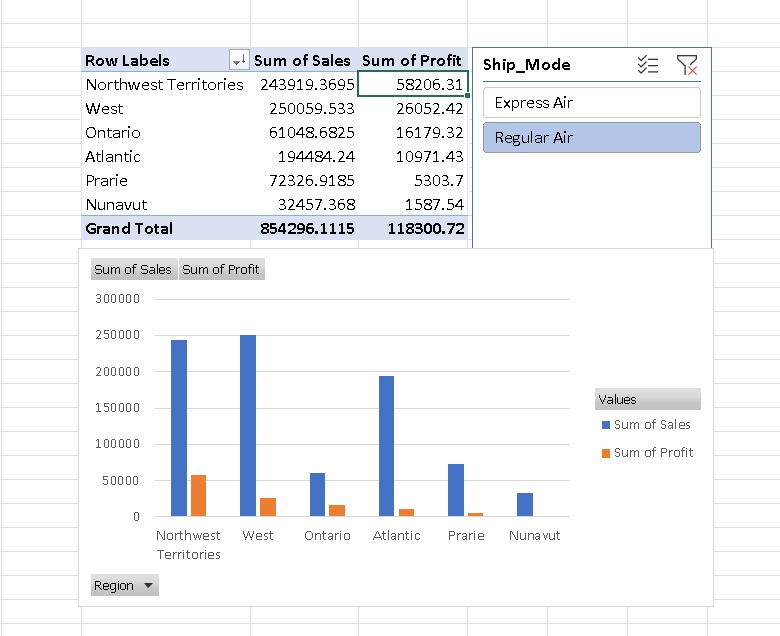
plt.ylabel('Sales')

plt.legend()

plt.show()

**REGION WISE PROFIT ANALYSIS :**





**CONCLUSION :**

* In conclusion, machine learning is not merely a tool but a transformative force that is reshaping the way we work, live, and interact with our world. Its future is brimming with possibilities, and our collective responsibility is to guide its development and application with wisdom and foresight.
* To summarize everything that has been stated so far, k-means clustering is a widely used unsupervised machine learning technique that enables the grouping of data into clusters based on similarity. It is a simple algorithm that can be applied to various domains and data types, including image and text data. k-means can be used for a variety of purposes. We can use it to perform dimensionality reduction also, where each transformed feature is the distance of the point from a cluster center.