**TEAM MEMBER**

**732521104030 :NAVEEN S**

***Phase 5 Submission Document***

***Project : Customer Segmentation using Data Science***

**Problem Definition:**

 The problem is to develop a machine learning model that predicts IMDb scores of movies available on Films based on features like genre, premiere date, runtime, and language. The objective is to create a model that accurately estimates the popularity of movies, helping users discover highly rated films that match their preferences. This project involves data preprocessing, feature engineering, model selection, training, and evaluation.

In this part you will need to understand the problem statement and create a document on what have you understood and how will you proceed ahead with solving the problem. Please think on a design and present in form of the document.

In this section you need to put your design into innovation to solve the problem. Create a doc around it and share the same for assessment.

In this section begin building your project by loading and preprocessing the dataset.

In this section continue building the project by performing different activities like feature engineering, model training, evaluation etc as per the instructions in the project.

In this section continue building the project by performing different activities like feature engineering, model training, evaluation etc as per the instructions in the project.

**Problem Definition:**

The problem is to implement data science techniques to segment customers based on their behavior, preferences, and demographic attributes. The goal is to enable businesses to personalize marketing strategies and enhance customer satisfaction. This project involves data collection, data preprocessing, feature engineering, clustering algorithms, visualization, and interpretation of results.

**Dataset Link:**[**https://www.kaggle.com/datasets/akram24/mall-customers**](https://www.kaggle.com/datasets/akram24/mall-customers)



**Design Thinking:**

**Data collection:**

Data collection is the process of collecting and analyzing information on relevant variables in a predetermined, methodical way so that one can respond to specific research questions, test hypotheses, and assess results.

**Data preprocessing**:  
Data preprocessing is an important step in the data mining process. It refers to the cleaning, transforming, and integrating of data in order to make it ready for analysis. The goal of data preprocessing is to improve the quality of the data and to make it more suitable for the specific data mining task.

**Feature Engineering**:

Feature Engineering is the process of creating new features or transforming existing features to improve the performance of a machine-learning model. It involves selecting relevant information from raw data and transforming it into a format that can be easily understood by a model.

**Clustering Algorithms**:

Clustering algorithms are procedures for partitioning data into groups or clusters such that the clusters are distinct, and members of each cluster belong together.

**Visualization**:

Visualization or visualisation may refer to: Visualization (graphics), the physical or imagining creation of images, diagrams, or animations to communicate a message. Data and information visualization, the practice of creating visual representations of complex data and information.

**Interpretation:**

Analyze and interpret the characteristics of each customer segment to derive actionable insights for marketing strategies.

**Program**:

import numpy as np linear algebra

import pandas as pd

import os

for dirname,. filenames in os.walk('C:\Users\Students\Documents\New folder\Mall Customers'):

for

filename in filenames:

print(os.path.join(dirname, filename))

import numpy as np

import pandas as pd import matplotlib.pyplot as plt

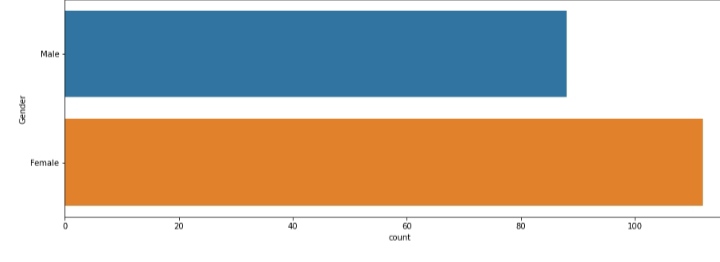
import seaborn as sns

df pd.read\_csv('/kaggle/input/mall-customers/Mall Customers.csv')

df.rename(columns=('Genre: 'Gender' }, inplace=True)

df.head()

**Output:**



**Data preprocessing:**

 Clean and preprocess the data, handle missing values, and convert categorical features into numerical representations.

**Feature Engineering**:

Create additional features that capture customer behavior and preferences, such as total spending, frequency of purchases, etc.

**Total spending**:

The sum of the price paid for one or more products or services multiplied by the amount of each item purchased.

**Frequency of purchases**:

Purchase frequency describes the number of times that your customers make a purchase from you within a specified period of time. This information is crucial in helping you to understand your customer retention rate, your customers' buying behaviors, and even the degree to which they're satisfied.

**Clustering Algorithms:**

Apply clustering algorithms like K-Means, DBSCAN, or hierarchical clustering to segment customers.

**K-Means :**

K-means is an unsupervised learning method for clustering data points. The algorithm iteratively divides data points into K clusters by minimizing the variance in each cluster.

**Hierarchical clustering:**

Hierarchical clustering is a popular method for grouping objects. It creates groups so that objects within a group are similar to each other and different from objects in other groups. Clusters are visually represented in a hierarchical tree called a dendrogram.

**Visualization:**

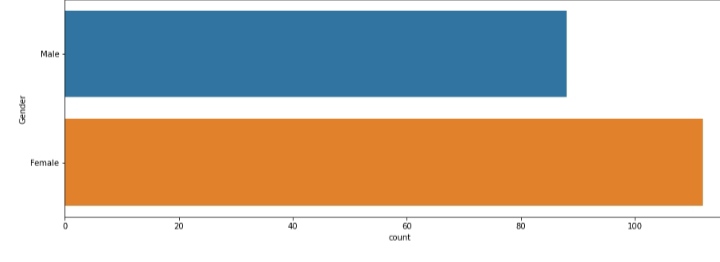
Visualize the customer segments using techniques like scatter plots, bar charts, and heatmaps.

**Plots :**

A scatter plot is a diagram where each value in the data set is represented by a dot. The Matplotlib module has a method for drawing scatter plots, it needs two arrays of the same length, one for the values of the x-axis, and one for the values of the y-axis: x = [5,7,8,7,2,17,2,9,4,11,12,9,6]

**Bar charts :**

The bar chart displays data using a number of bars, each representing a particular category. The height of each bar is proportional to a specific aggregation (for example the sum of the values in the category it represents). The categories could be something like an age group or a geographical location.



**Heatmaps:**

A heatmap is a graphical representation of data that uses a system of color coding to represent different values. Heatmaps are used in various forms of analytics but are most commonly used to show user behavior on specific web pages or webpage templates.

**Interpretation:**

Analyze and interpret the characteristics of each customer segment to derive actionable insights for marketing strategies.

import numpy as np linear algebra

import pandas as pd

import os

for dirname,. filenames in os.walk('C:\Users\Students\Documents\New folder\Mall Customers '):

for

filename in filenames:

print(os.path.join(dirname, filename))

import numpy as np

import pandas as pd import matplotlib.pyplot as plt

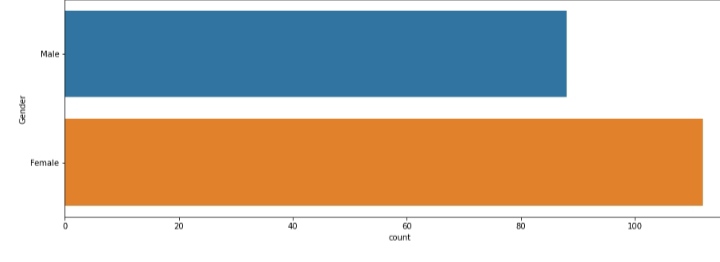
import seaborn as sns

df pd.read\_csv('/kaggle/input/mall-customers/Mall Customers.csv')

df.rename(columns=('Genre: 'Gender' }, inplace=True)

df.head()

**Output:**

****

plt.figure(1,figsize=(15,6))

n =0

forx **in** ['Age','Annual Income (k$)','Spending Score (1-100)']:

n +=1

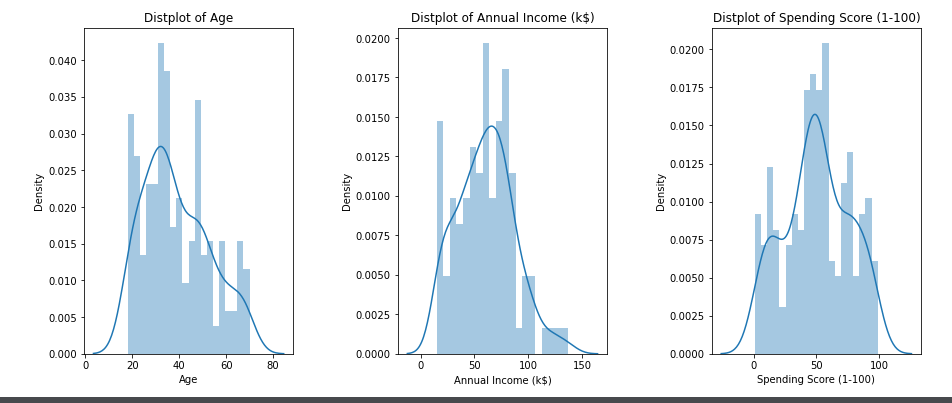
plt.subplot(1,3,n)

plt.subplots\_adjust(hspace=0.5,wspace=0.5)

sns.distplot(df[x],bins=20)

plt.title('Distplot of **{}**'.format(x))

plt.show()

**Output :**

plt.figure(1,figsize=(15,6))

n=0

forcolsin['Age','Annual Income (k$)','Spending Score (1-100)']:

n+=1

plt.subplot(1,3,n)

sns.set(style="whitegrid")

plt.subplots\_adjust(hspace=0.5,wspace=0.5)

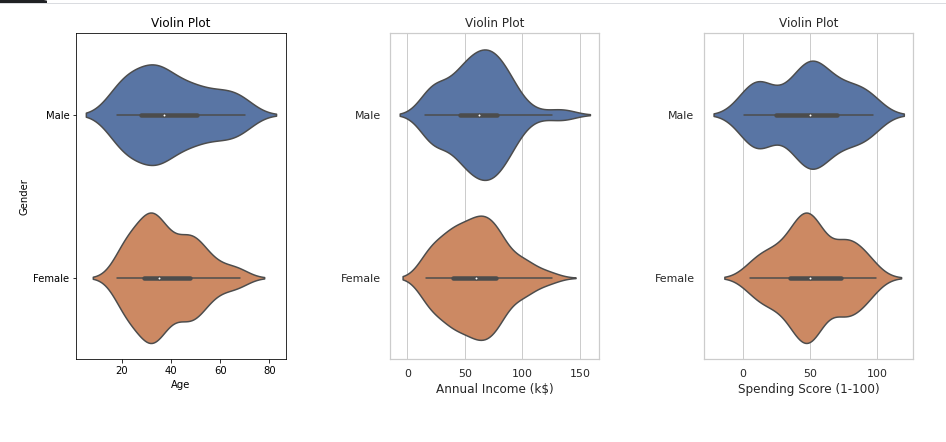
sns.violinplot(x=cols,y='Gender',data=df)

plt.ylabel('Gender'ifn==1else'')

plt.title('Violin Plot')

plt.show()

**Output:**

****

**Data Loading:**

The Load method provides a technique for filling a single DataTable with data, retrieved from an IDataReader instance. This method provides the same functionality, but allows you to load multiple result sets from an IDataReader into multiple tables within a DataSet.

**5 Different Ways to Load Data in Python**

As a beginner, you might only know a single way to load data (normally in CSV) which is to read it using pandas.read\_csv function. It is one of the most mature and strong functions, but other ways are a lot helpful and will definitely come in handy sometimes.

The ways that I am going to discuss are:

* Manual function
* loadtxt function
* genfromtxt function
* read\_csv function
* Pickle

The dataset that we are going to use to load data can be found [here](http://eforexcel.com/wp/downloads-18-sample-csv-files-data-sets-for-testing-sales/). It is named as 100-Sales-Records.

**Imports**  
We will use Numpy, Pandas, and Pickle packages so import them.

import numpy as np

import pandas as pd

import pickle

**1. Manual Function**

This is the most difficult, as you have to design a custom function, which can load data for you. You have to deal with Python’s normal filing concepts and using that you have to read a .csv file.

Let’s do that on 100 Sales Records file.

def load\_csv(filepath):

data = []

col = []

checkcol = False

with open(filepath) as f:

for val in f.readlines():

val = val.replace("\n","")

val = val.split(',')

if checkcol is False:

col = val

checkcol = True

else:

data.append(val)

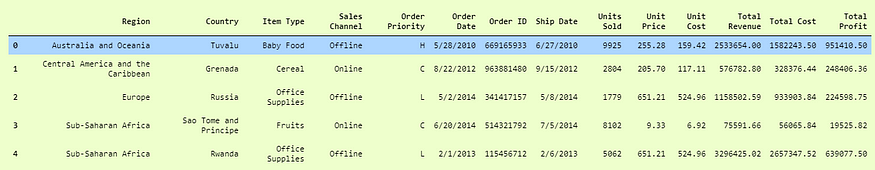
df = pd.DataFrame(data=data, columns=col)

return df

**Output**

myData = load\_csv('100 Sales Record.csv')

print(myData.head())



**2. Numpy.loadtxt function**

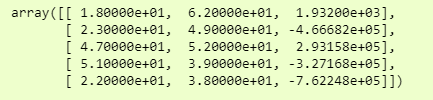
This is a built-in function in Numpy, a famous numerical library in Python. It is a really simple function to load the data. It is very useful for reading data which is of the same datatype.

When data is more complex, it is hard to read using this function, but when files are easy and simple, this function is really powerful.

df = np.loadtxt('convertcsv.csv', delimeter = ',')

Here we simply used the loadtxt function as passed in delimeter as ',' because this is a CSV file.

Now if we print df, we will see our data in pretty decent numpy arrays that are ready to use.

print(df[:5,:])

**3. Numpy.genfromtxt()**

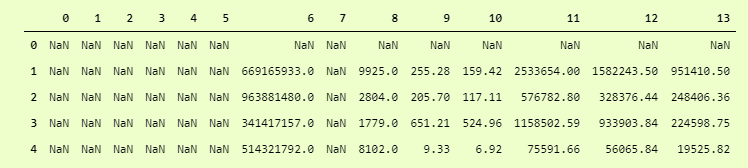
We will use the dataset, which is ‘100 Sales Records.csv’ which we used in our first example to demonstrate that we can have multiple data types in it.

Let’s jump to code.

data = np.genfromtxt('100 Sales Records.csv', delimiter=',')

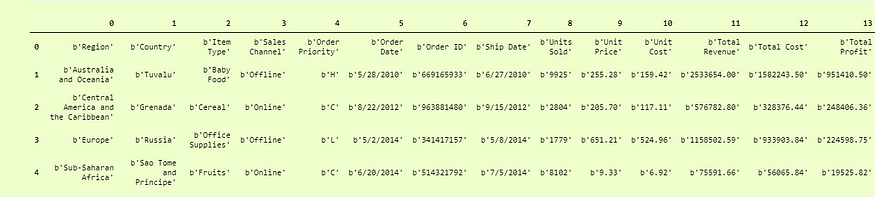
and to see it more clearly, we can just see it in a dataframe format, i.e.,

>>>pd.DataFrame(data)

Just add another dtype parameter and set dtype to None which means that it has to take care of datatypes of each column itself. Not to convert whole data to single dtype.

data = np.genfromtxt('100 Sales Records.csv', delimiter=',', dtype=None)

And then for output:

>>>pd.DataFrame(data).head()

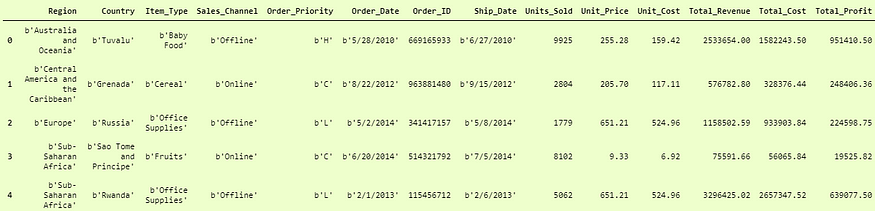
Quite better than the first one, but here our Columns titles are Rows, to make them column titles, we have to add another parameter which is names and set it to True so it will take the first row as the Column Titles.

i.e.

data = np.genfromtxt('100 Sales Records.csv', delimiter=',', dtype=None, names = True)

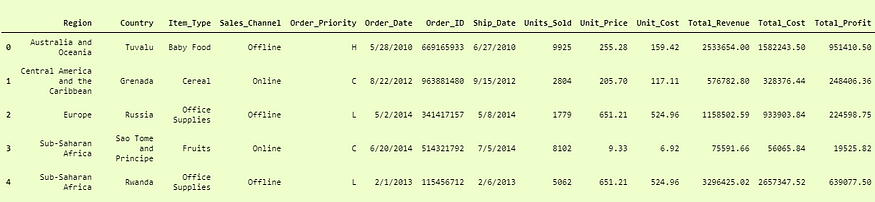
and we can print it as:

>>>pd.DataFrame(df3).head()

Now the last problem is that the columns which are of string data types are not the actual strings, but they are in bytes format. You can see that before every string, we have a b' so to encounter them, we have to decode them in utf-8 format.

df3 = np.genfromtxt('100 Sales Records.csv', delimiter=',', dtype=None, names=True, encoding='utf-8')

This will return our dataframe in the desired form.

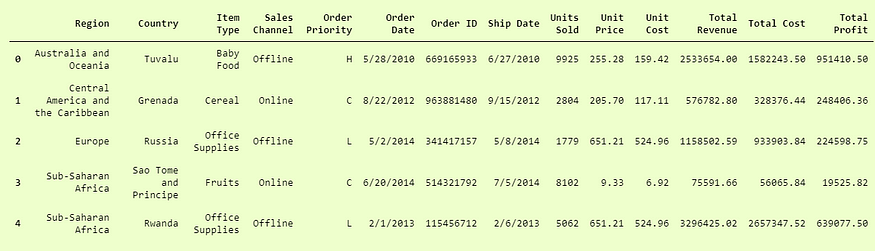
>>>pd.DataFrame(df3)

**4. Pandas.read\_csv()**

Pandas is a very popular data manipulation library, and it is very commonly used. One of it’s very important and mature functions is read\_csv() which can read any .csv file very easily and help us manipulate it. Let’s do it on our 100-Sales-Record dataset.

This function is very popular due to its ease of use. You can compare it with our previous codes, and you can check it.

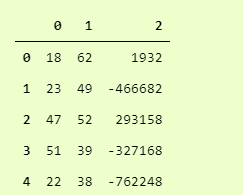
>>>pdDf = pd.read\_csv('100 Sales Record.csv')

>>>pdDf.head()

This was actually so simple and easy to use. Pandas.read\_csv definitely offers a lot of other parameters to tune in our data set, for example in our convertcsv.csv file, we had no column names so we can read it as:

>>>newdf = pd.read\_csv('convertcsv.csv', header=None)

>>>newdf.head()



**5. Pickle**

When your data is not in a good, human-readable format, you can use pickle to save it in a binary format. Then you can easily reload it using the pickle library.

We will take our 100-Sales-Record CSV file and first save it in a pickle format so we can read it.

with open('test.pkl','wb') as f:

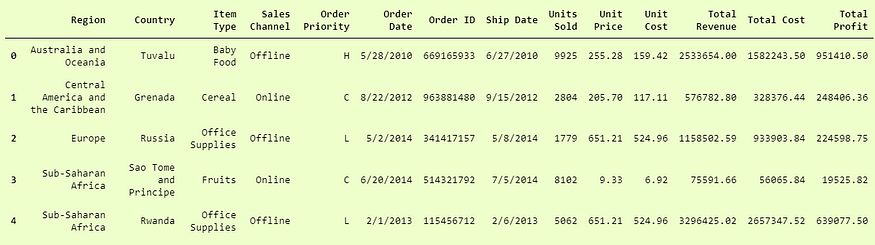
pickle.dump(pdDf, f)

This will create a new file test.pkl which has inside it our pdDf from Pandas heading.

Now to open it using pickle, we just have to use pickle.load function.

with open("test.pkl", "rb") as f:

d4 = pickle.load(f)

>>> d4.head()

**Data Preprocessing:**

Data preprocessing is converting raw data into legible and defined sets that allow businesses to conduct data mining, analyze the data, and process it for business activities. It's important for businesses to preprocess their data correctly, as they use various forms of input to collect raw data, which can affect its quality. Preprocessing data is an important step, as raw data can be inconsistent or incomplete in its formatting. Effectively preprocessing raw data can increase its accuracy, which can increase the quality of projects and improve its reliability.

**Importance of data preprocessing**

Preprocessing data is an important step for data analysis. The following are some benefits of preprocessing data:

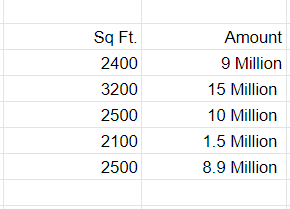
It improves accuracy and reliability. Preprocessing data removes missing or inconsistent data values resulting from human or computer error, which can improve the accuracy and quality of a dataset, making it more reliable.

It makes data consistent. When collecting data, it's possible to have data duplicates, and discarding them during preprocessing can ensure the data values for analysis are consistent, which helps produce accurate results.

It increases the data's algorithm readability. Preprocessing enhances the data's quality and makes it easier for machine learning algorithms to read, use, and interpret it.

**Feature engineering:**

Feature engineering is a machine learning technique that leverages data to create new variables that aren’t in the training set. It can produce new features for both supervised and unsupervised learning, with the goal of**simplifying and speeding up data transformations**while also**enhancing model accuracy**. Feature engineering is required when working with machine learning models. Regardless of the data or architecture, a terrible feature will have a direct impact on your model.



# Applying clustering algorithms :

* Choose the number of clusters k. The first step in k-means is to pick the number of clusters, k.
* Select k random points from the data as centroids. ...
* Assign all the points to the closest cluster centroid. ...
* Recompute the centroids of newly formed clusters. ...
* Repeat steps 3 and 4.

**Visualization :**

The physical or imagining information.creation of images, diagrams, or animations to communicate a message. Data and information visualization, the practice of creating visual representations of complex data

**(1) outcome visualization:** [envisioning the desired future end-point](https://www.betterup.com/blog/career-planning), and

**(2) process visualization:**envisioning every step toward that desired outcome. With all the senses engaged

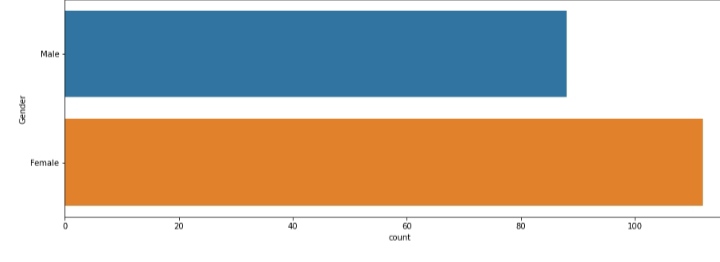
**Interpretation :**

* The basic definition of interpretation, according to the Webster dictionary is the “action of explaining the meaning of something; the way something is explained or understood.”
* In terms of language, the definition of interpretation should be broader: rendering a spoken or signed message into another spoken or signed language, preserving the register and meaning of the source language content.
* It is the spoken or signed language communication between users of different languages. A language interpreter or sign language interpreter must not only quickly and carefully interpret meaning, but also tone and intent of the original message into the target or interpreted language

**Feature Engineering**:

Create additional features that capture customer behavior and preferences, such as total spending, frequency of purchases, etc.

**Barchart:**

******

**Heatmaps** :

A heatmap is a graphical representation of data that uses a system of color coding to represent different values. Heatmaps are used in various forms of analytics but are most commonly used to show user behavior on specific web pages or webpage templates.

**Interpretation:**

Analyze and interpret the characteristics of each customer segment to derive actionable insights for marketing strategies.

**Program:**

import numpy as np linear algebra

import pandas as pd

import os

for dirname,. filenames in os.walk('C:\Users\Students\Documents\New folder\Mall Customers '):

for

filename in filenames:

print(os.path.join(dirname, filename))

import numpy as np

import pandas as pd import matplotlib.pyplot as plt

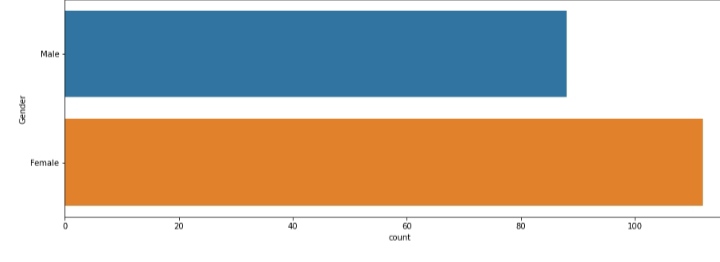
import seaborn as sns

df pd.read\_csv('/kaggle/input/mall-customers/Mall Customers.csv')

df.rename(columns=('Genre: 'Gender' }, inplace=True)

df.head()

**Output:**



plt.figure(1,figsize=(15,6))

n =0

for x **in** ['Age','Annual Income (k$)','Spending Score (1-100)']:

n +=1

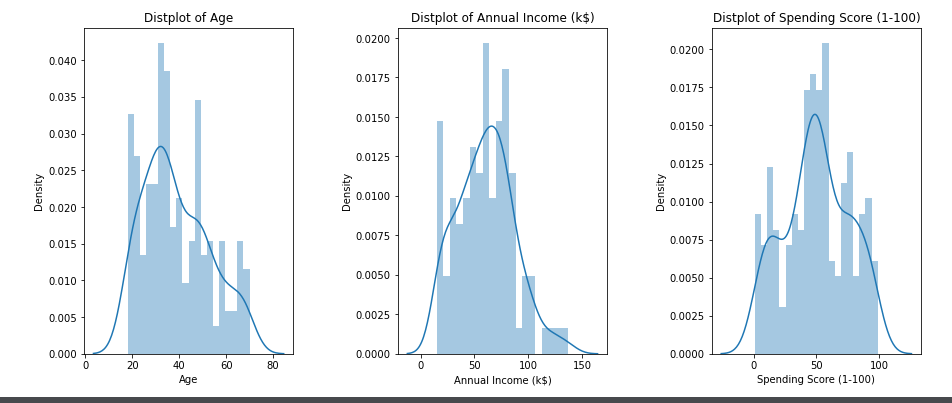
plt.subplot(1,3,n)

plt.subplots\_adjust(hspace=0.5,wspace=0.5)

sns.distplot(df[x],bins=20)

plt.title('Distplot of **{}**'.format(x))

plt.show()

**Output :**

plt.figure(1,figsize=(15,6))

n=0

forcolsin['Age','Annual Income (k$)','Spending Score (1-100)']:

n+=1

plt.subplot(1,3,n)

sns.set(style="whitegrid")

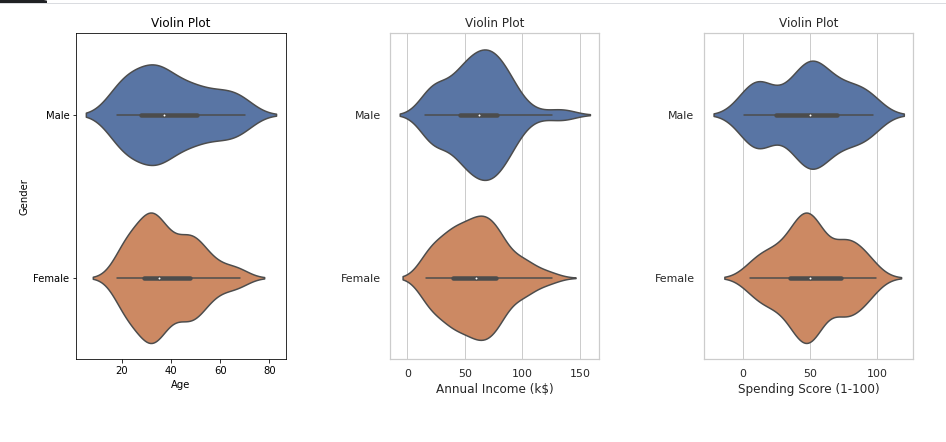
plt.subplots\_adjust(hspace=0.5,wspace=0.5)

sns.violinplot(x=cols,y='Gender',data=df)

plt.ylabel('Gender'ifn==1else'')

plt.title('Violin Plot')

plt.show()

**Output:**

**Conclusion:**

In conclusion, customer segmentation is a critical aspect of marketing strategy, and machine learning has become an increasingly popular tool for automating the process. By using machine learning algorithms to process vast amounts of customer data, companies can quickly identify new trends and patterns, target specific customer segments with tailored promotions, and make more informed marketing decisions. With its ability to process data in real time, eliminate the need for manual analysis, and continuously improve over time, machine learning is a powerful tool for customer segmentation.

The process of customer segmentation ensures that your brand is customer-centric and helps you serve them better.

Customer segmentation involves implementing data science methods to divide the customer base into smaller groups based on certain characteristics. It assists marketing managers in better understanding their customers' preferences and presenting them with better-targeted advertisements.

With customer segmentation, you can easily personalize your marketing, service, and sales efforts to the needs of specific groups. The result is a potential boost to customer loyalty and conversions.