**Abstract**

We live in a world where large and vast amount of data is

collected daily. Analysing such data is an important need. In the modern era of innovation, where there is a large competition to be better then everyone, the business strategy needs to be according to the modern conditions. The business done today runs on the basis of innovative ideas as there are large number of potential customers who are confounded to what to buy and what not to buy. The companies doing the business are also not able to diagnose the target potential customers. This is where the machine learning comes into picture, the various algorithms are applied to identify the hidden patterns in the data for better decision making. The concept of which customer segment to target is done using the customer segmentation process using the clustering technique. In this paper, the clustering algorithm used is K-means algorithm which is the partitioning algorithm, to segment the customers according to the similar characteristics. To determine the optimal clusters, elbow method is used.

|  |  |
| --- | --- |
| TABLE OF CONTENT |  |
| LIST OF ABBRIVATIONS | PAGE NO |
| CHAPTER 1: INTRODUCTION   * 1. Introduction   2. Project Aim & Agenda   3. Existing System & Its Limitations | 1 |
| CHAPTER 2: REQUREMENT ANALYSIS  2.1 Clustering Algorithm | 09 |
| CHAPTER 3: SPECIFICATIONS  3.2 Software Requirement Specifications  3.3 Hardware Requirement Specifications | 11 |
| CHAPTER 4: CODEING FOR COUSTMER SEGMENTATION    4.1 Coding | 15 |

**CHAPTER -1**

**1.1 Introduction**

Over the years, the competition amongst businesses is increased and the large historical data that is available has resulted in the widespread use of data mining techniques in extracting the meaningful and strategic information from the database of the organisation. Data mining is the process where methods are applied to extract data patterns in order to present it in the human readable format which can be used for the purpose of decision support. According to,[4] Clustering techniques consider data tuples as objects. They partition the data objects into groups or clusters, so that objects within a cluster are similar to one another and dissimilar to objects in other clusters. Customer Segmentation is the process of division of customer base into several groups called as customer segments such that each customer segment consists of customers who have similar characteristics. The segmentation is based on the similarity in different ways that are relevant to marketing such as gender, age, interests, and miscellaneous spending habits. The customer segmentation has the importance as it includes, the ability to modify the programs of market so that it is suitable to each of the customer segment, support in business decision; identification of products associated with each customer segment and to mange the demand and supply of that product; identifying and targeting the potential customer base, and predicting customer defection, providing directions in finding the solutions. The thrust of this paper is to identify customer segments using the data mining approach, using the partitioning algorithm called as K-means clustering algorithm. The elbow method determines the optimal clusters.

**Motivation**

Over the years, as there is very strong competition in the business world, the organizations have to enhance their profits and business by satisfying the demands of their customers and attract new customers according to their needs. The identification of customers and satisfying the demands of each customer is a very complex and tedious task. This is because customers may be different according to their demands, tastes, preferences and so on. Instead of “one-size-fits-all” approach, customer segmentation clusters the customers into groups sharing the same properties or behavioural characteristics. According to, customer segmentation is a strategy of dividing the market into homogenous groups. The data used in customer segmentation technique that divides the customers into groups depends on various factors like, data geographical conditions, economic conditions, demographical conditions as well as behavioural patterns. The customer segmentation technique allows the business to make better use of their marketing budgets, gain a competitive edge over their rival companies, demonstrating the better knowledge of the needs of the customer.

It also helps an organization in, increasing their marketing efficiency, determining new market opportunities, making better brand strategy, identifying customers retention.

* 1. **Project Aim & Agenda**

Customer Segmentation is an unsupervised method of targeting the customers in order to increase sales and market goods in a better way

This project deals with real-time data where we have to segment the customers in the form f clusters using the K-Means algorithm

The data set consists of important variables like Age, Gender, annual income, etc.

With the help of the algorithms, we can easily visualize the data and can get a segmentation of each customer so that we can target the customers in the better way

Customer Segmentation is the subdivision of a market into discrete customer groups that share similar characteristics. Customer Segmentation can be a powerful means to identify unsatisfied customer needs. Using the above data companies can then outperform the competition by developing uniquely appealing products and services.

**Advantages of Customer Segmentation**

1. Help identify least and most profitable customers, thus helping the business to concentrate marketing activities on those most likely to buy your products or services
2. Helps build loyal relationships with customers by developing and offering them the products and services they want
3. Helps improve customer service
4. Helps maximize the use of your resources
5. Helps improve or tweak products to meet customer requirements
6. Helps increase profit by keeping costs down
7. Determine appropriate product pricing.
8. Develop customized marketing campaigns.
9. Design an optimal distribution strategy.

10. Choose specific product features for deployment.

11. Prioritize new product development efforts.

**Types of Customer Groups**

To begin with, you need to understand whether your business is B2B or B2C. For the latter, the easiest way is to segment your customers by location, the lifestyle of consumers, attitudes, and behavior.

For B2B, you may segment customers according to the industry sector, public or private, size and location, how they operate, their buying patterns, and so on.

**How To Segment Customers**

For that, you will need variables like the individual’s geographical location, age, gender, or if it is an organization, its sector, size, and buying patterns.

You can use vertical as well as horizontal segmentation. In the former, you may select particular industries or professions to whom your product or service is most likely to appeal. Under horizontal segmentation, you may select variables such as job title across a range of organizations. Many outfits, to find out about their customers, also carry out market research. Another more sophisticated way to do the job is to use mathematical models to analyze large amounts of data to group customers with similar data sets into particular segments.

Ultimately, the idea behind the exercise is to segment customers into groups based on how much total future value they are predicted to bring to your company, to direct each group in a way most likely to maximize that lifetime value.

**Some of the segmentation tactics that many firms adopt include:**

* Customer Interests

At the time of registration or purchase of a product, ask the customer about their interests and preferences. This is one way of clubbing same-interest customers into one group. If Group A is price-sensitive, send promotions-related material their way.

* Loyalty Dividends

No doubt, repeat customers should be rewarded, but the first step is to identify them and club them together. A starting point could be your business’s website which may have recorded a customer’s order history.

You then have two segments of repeat customers. Those that buy from your site without a promotion, and those that buy with one. To target even more specific segments, identify repeat customers who buy from your site because they like the products, like the customer service.

* Devices Used

Identify how your customers are logging into your site. Are they using mobiles or desktops? If mobiles, then whether Android or iPhone or some other? Put out offers accordingly to each of the groups. Eg: Those using iOS devices can be classified as high-end customers, while those using Blackberry can be segmented as “ high value” clients.

**1.3 Existing System**

Existing system tells about the developed project for how many modules are completed.

Before we get into the process, I will give you a brief on what kind of steps we will get.

* Gather the data
* Create Recency Frequency Monetary (RFM) table
* Manage skewness and scale each variable
* Explore the data
* Cluster the data
* Interpret the result

**CHAPTER - 2**

**REQUIREMENT ANALYSIS**

Need of Customer Segmentation. Types of Segmentation. Customer Segmentation using RFM Analysis. Identify Potential Customer Segments using RFM in Python.

* Calculate the Recency, Frequency, Monetary values for each customer.
* Add segment bin values to RFM table using quartile.
* Sort the customer RFM score in ascending order

**2.1 Using K-Means clustering algorithm**

1. Specify number of clusters *K*.
2. Initialize centroids by first shuffling the dataset and then randomly selecting *K*data points for the centroids without replacement.
3. Keep iterating until there is no change to the centroids. assignment of data points to clusters isn’t changing.

K-means algorithm in one of the most popular centroid based algorithm. Suppose data set, D, contains n objects in space. Partitioning methods distribute the objects in D into k clusters, C1,...,Ck , that is, Ci ⊂ D and Ci ∩Cj = ∅ for (1 ≤ i, j ≤ k). A centroid-based partitioning technique uses the centroid of a cluster, Ci , to represent that cluster. Conceptually, the centroid of a cluster is its center point. The difference between an object p ∈ Ci and ci , the representative of the cluster, is measured by dist(p,ci), where dist(x,y) is the Euclidean distance between two points x and y.

Algorithm: The k-means algorithm for partitioning, where each cluster’s center is represented by the mean value of the objects in the cluster. Input: k: the number of clusters, D: a data set containing n objects. Output: A set of k clusters. Method: (1) arbitrarily choose k objects from D as the initial cluster centers; (2) repeat (3) (re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster; (4) update the cluster means, that is, calculate the mean value of the objects for each cluster; (5) until no change.

**CHAPTER -3**

**SPECFICATIONS**

**SYSTEM SPECIFICATIONS**:

**3.1 Hardware Requirements:**

System : intel Core i3

Hard Disk : 200 MB RAM : 4GB

**3.2 Software Requirements:**

Operating System : Windows 7 Coding

Language : Python

Data Base : DBMS

# 

# Environment and Tools:

**1.Scikit-Learn**

Scikit-learn is probably the most useful **library for machine learning in Python**. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

# 2.Seaborn

Seaborn is **a data visualization library built on top of matplotlib** and closely integrated with pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.

**3.Numpy**

NumPy is **the fundamental package for scientific computing in Python**. NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Typically, such operations are executed more efficiently and with less code than is possible using Python's built-in sequences.

**4.Pandas**

Pandas stands for **“Python Data Analysis Library ”**. What's cool about Pandas is that it takes data (like a CSV or TSV file, or a SQL database) and creates a Python object with rows and columns called data frame that looks very similar to table in a statistical software.

**5.Matplotlib**

Matplotlib is **a plotting library for the Python programming language** and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.

**K- Means Clustering Algorithm**

1. Specify number of clusters *K*.
2. Initialize centroids by first shuffling the dataset and then randomly selecting *K*data points for the centroids without replacement.
3. Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn’t changing.

K Means Clustering where K=3



**CHAPTER -4**

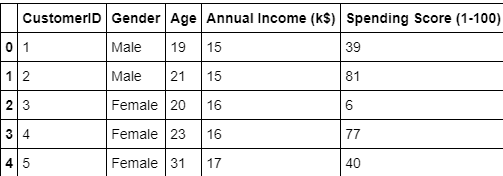
**CODEING FOR COUSTMER SEGMENTATION**

**4.1 Code :**

Without much ado, let’s get started with the code. The complete project on github.

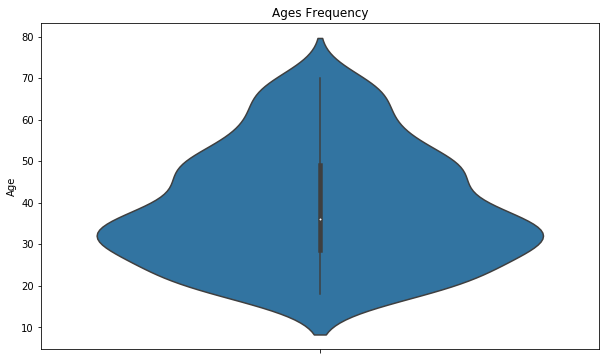
I started with loading all the libraries and dependencies. The columns in the dataset are customer id, gender, age, income and spending score.

|  |  |
| --- | --- |
|  | import numpy as np |
|  | import pandas as pd |
|  | import matplotlib.pyplot as plt |
|  | import seaborn as sns |
|  |  |
|  | df = pd.read\_csv("../input/customer-segmentation-tutorial-in-python/Mall\_Customers.csv") |
|  | df.head() |



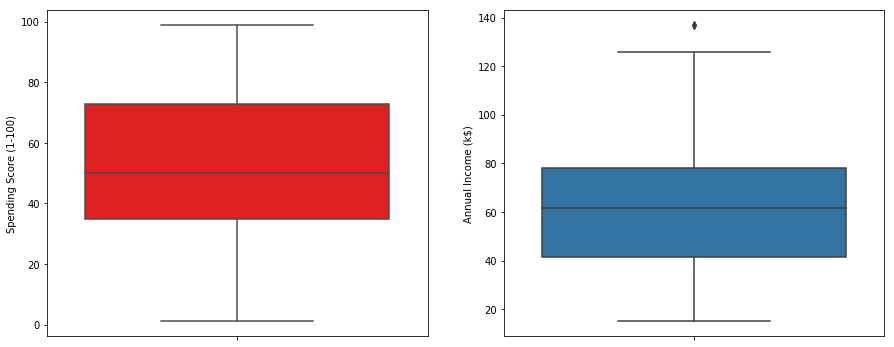
I dropped the id column as that does not seem relevant to the context. Also I plotted the age frequency of customers.

|  |  |
| --- | --- |
|  | df.drop(["CustomerID"], axis = 1, inplace=True) |
|  |  |
|  | plt.figure(figsize=(10,6)) |
|  | plt.title("Ages Frequency") |
|  | sns.axes\_style("dark") |
|  | sns.violinplot(y=df["Age"]) |
|  | plt.show() |



Next I made a box plot of spending score and annual income to better visualize the distribution range. The range of spending score is clearly more than the annual income range.

|  |  |
| --- | --- |
|  | plt.figure(figsize=(15,6)) |
|  | plt.subplot(1,2,1) |
|  | sns.boxplot(y=df["Spending Score (1-100)"], color="red") |
|  | plt.subplot(1,2,2) |
|  | sns.boxplot(y=df["Annual Income (k$)"]) |
|  | plt.show() |

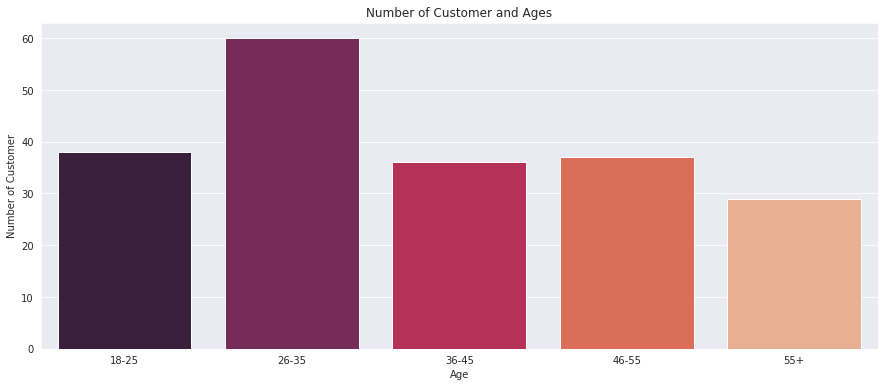


I made a bar plot to check the distribution of male and female population in the dataset. The female population clearly outweighs the male counterpart.

|  |  |
| --- | --- |
|  | genders = df.Gender.value\_counts() |
|  | sns.set\_style("darkgrid") |
|  | plt.figure(figsize=(10,4)) |
|  | sns.barplot(x=genders.index, y=genders.values) |
|  | plt.show()  1_0xcFGU42JWMRNNEJpnTOhA.png |

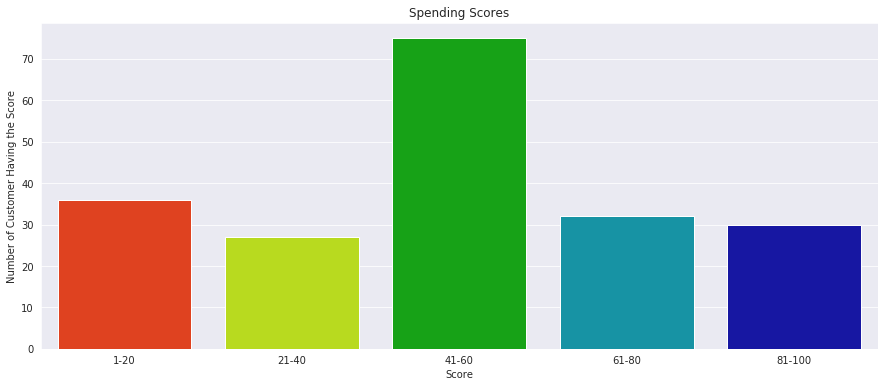
Next I made a bar plot to check the distribution of number of customers in each age group. Clearly the 26–35 age group outweighs every other age group.

|  |  |
| --- | --- |
|  | age18\_25 = df.Age[(df.Age <= 25) & (df.Age >= 18)] |
|  | age26\_35 = df.Age[(df.Age <= 35) & (df.Age >= 26)] |
|  | age36\_45 = df.Age[(df.Age <= 45) & (df.Age >= 36)] |
|  | age46\_55 = df.Age[(df.Age <= 55) & (df.Age >= 46)] |
|  | age55above = df.Age[df.Age >= 56] |
|  |  |
|  | x = ["18-25","26-35","36-45","46-55","55+"] |
|  | y = [len(age18\_25.values),len(age26\_35.values),len(age36\_45.values),len(age46\_55.values),len(age55above.values)] |
|  |  |
|  | plt.figure(figsize=(15,6)) |
|  | sns.barplot(x=x, y=y, palette="rocket") |
|  | plt.title("Number of Customer and Ages") |
|  | plt.xlabel("Age") |
|  | plt.ylabel("Number of Customer") |
|  | plt.show() |



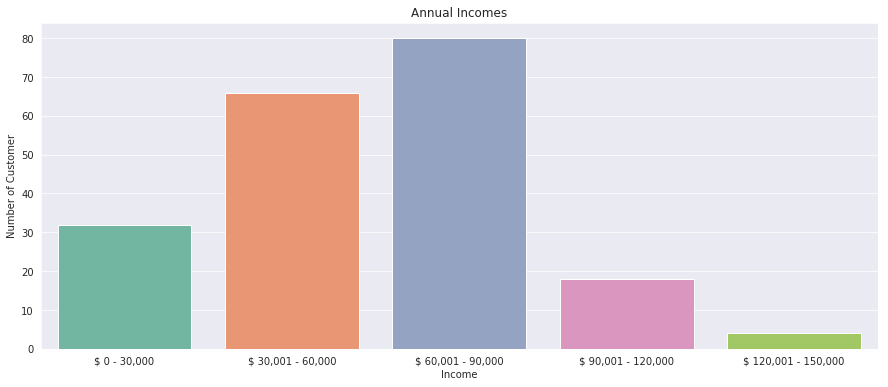
I continued with making a bar plot to visualize the number of customers according to their spending scores. The majority of the customers have spending score in the range 41–60.

|  |  |
| --- | --- |
|  | ss1\_20 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 1) & (df["Spending Score (1-100)"] <= 20)] |
|  | ss21\_40 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 21) & (df["Spending Score (1-100)"] <= 40)] |
|  | ss41\_60 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 41) & (df["Spending Score (1-100)"] <= 60)] |
|  | ss61\_80 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 61) & (df["Spending Score (1-100)"] <= 80)] |
|  | ss81\_100 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 81) & (df["Spending Score (1-100)"] <= 100)] |
|  |  |
|  | ssx = ["1-20", "21-40", "41-60", "61-80", "81-100"] |
|  | ssy = [len(ss1\_20.values), len(ss21\_40.values), len(ss41\_60.values), len(ss61\_80.values), len(ss81\_100.values)] |
|  |  |
|  | plt.figure(figsize=(15,6)) |
|  | sns.barplot(x=ssx, y=ssy, palette="nipy\_spectral\_r") |
|  | plt.title("Spending Scores") |
|  | plt.xlabel("Score") |
|  | plt.ylabel("Number of Customer Having the Score") |
|  | plt.show() |

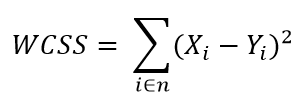


Also I made a bar plot to visualize the number of customers according to their annual income. The majority of the customers have annual income in the range 60000 and 90000.

|  |  |
| --- | --- |
|  | ai0\_30 = df["Annual Income (k$)"][(df["Annual Income (k$)"] >= 0) & (df["Annual Income (k$)"] <= 30)] |
|  | ai31\_60 = df["Annual Income (k$)"][(df["Annual Income (k$)"] >= 31) & (df["Annual Income (k$)"] <= 60)] |
|  | ai61\_90 = df["Annual Income (k$)"][(df["Annual Income (k$)"] >= 61) & (df["Annual Income (k$)"] <= 90)] |
|  | ai91\_120 = df["Annual Income (k$)"][(df["Annual Income (k$)"] >= 91) & (df["Annual Income (k$)"] <= 120)] |
|  | ai121\_150 = df["Annual Income (k$)"][(df["Annual Income (k$)"] >= 121) & (df["Annual Income (k$)"] <= 150)] |
|  |  |
|  | aix = ["$ 0 - 30,000", "$ 30,001 - 60,000", "$ 60,001 - 90,000", "$ 90,001 - 120,000", "$ 120,001 - 150,000"] |
|  | aiy = [len(ai0\_30.values), len(ai31\_60.values), len(ai61\_90.values), len(ai91\_120.values), len(ai121\_150.values)] |
|  |  |
|  | plt.figure(figsize=(15,6)) |
|  | sns.barplot(x=aix, y=aiy, palette="Set2") |
|  | plt.title("Annual Incomes") |
|  | plt.xlabel("Income") |
|  | plt.ylabel("Number of Customer") |
|  | plt.show() |

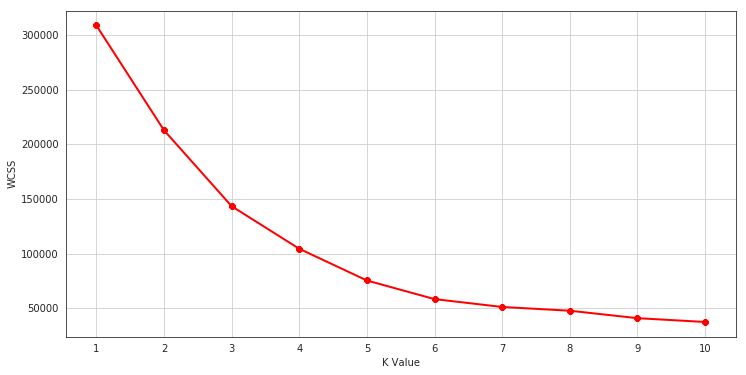


Next I plotted Within Cluster Sum Of Squares (WCSS) against the the number of clusters (K Value) to figure out the optimal number of clusters value. WCSS measures sum of distances of observations from their cluster centroids which is given by the below formula.



where *Yi* is centroid for observation *Xi*. The main goal is to maximize number of clusters and in limiting case each data point becomes its own cluster centroid.

|  |  |
| --- | --- |
|  | from sklearn.cluster import KMeans |
|  | wcss = [] |
|  | for k in range(1,11): |
|  | kmeans = KMeans(n\_clusters=k, init="k-means++") |
|  | kmeans.fit(df.iloc[:,1:]) |
|  | wcss.append(kmeans.inertia\_) |
|  | plt.figure(figsize=(12,6)) |
|  | plt.grid() |
|  | plt.plot(range(1,11),wcss, linewidth=2, color="red", marker ="8") |
|  | plt.xlabel("K Value") |
|  | plt.xticks(np.arange(1,11,1)) |
|  | plt.ylabel("WCSS") |
|  | plt.show() |



**The Elbow Method**

Calculate the Within Cluster Sum of Squared Errors (WSS) for different values of k, and choose the k for which WSS first starts to diminish. In the plot of WSS-versus k, this is visible as an elbow.

The optimal K value is found to be 5 using the elbow method.

Finally I made a 3D plot to visualize the spending score of the customers with their annual income. The data points are separated into 5 classes which are represented in different colours as shown in the 3D plot.

|  |  |
| --- | --- |
|  |  |
|  | km = KMeans(n\_clusters=5) |
|  | clusters = km.fit\_predict(df.iloc[:,1:]) |
|  | df["label"] = clusters |
|  |  |
|  | from mpl\_toolkits.mplot3d import Axes3D |
|  | import matplotlib.pyplot as plt |
|  | import numpy as np |
|  | import pandas as pd |
|  |  |
|  | fig = plt.figure(figsize=(20,10)) |
|  | ax = fig.add\_subplot(111, projection='3d') |
|  | ax.scatter(df.Age[df.label == 0], df["Annual Income (k$)"][df.label == 0], df["Spending Score (1-100)"][df.label == 0], c='blue', s=60) |
|  | ax.scatter(df.Age[df.label == 1], df["Annual Income (k$)"][df.label == 1], df["Spending Score (1-100)"][df.label == 1], c='red', s=60) |
|  | ax.scatter(df.Age[df.label == 2], df["Annual Income (k$)"][df.label == 2], df["Spending Score (1-100)"][df.label == 2], c='green', s=60) |
|  | ax.scatter(df.Age[df.label == 3], df["Annual Income (k$)"][df.label == 3], df["Spending Score (1-100)"][df.label == 3], c='orange', s=60) |
|  | ax.scatter(df.Age[df.label == 4], df["Annual Income (k$)"][df.label == 4], df["Spending Score (1-100)"][df.label == 4], c='purple', s=60) |
|  | ax.view\_init(30, 185) |
|  | plt.xlabel("Age") |
|  | plt.ylabel("Annual Income (k$)") |
|  | ax.set\_zlabel('Spending Score (1-100)') |
|  | plt.show()  1_RFiP_B-AfYYFfw0KaiB1lA.png |

### Conclusions

K means clustering is one of the most popular clustering algorithms and usually the first thing practitioners apply when solving clustering tasks to get an idea of the structure of the dataset. The goal of K means is to group data points into distinct non-overlapping subgroups. One of the major application of K means clustering is segmentation of customers to get a better understanding of them which in turn could be used to increase the revenue of the company.