Project to

IBM NAAN MUTHALVAN

APPLIED DATA SCIENCE

Submitted by

Selvin Raajkumar C (911721106315)

Vinoth Raajan K (911721106039)

Sriram N (911721106317)

Praveen Kumar (911721106310)

**MOUNT ZION COLLEGE OF ENGINEERING AND TECHNOLOGY**

**NAAC Accredited Institution (with A+ Grade)**

**Pudukkottai-622507, Tamilnadu.**



**Customer segmentation using data science**

**Project Title:** Customer segmentation

**Phase 4**: The customer segmentation model by:

* Feature engineering
* Applying clustering algorithms
* Visualization
* Interpretation.

**Problem Statement: Customer Segmentation using Data Science**

**Background:** In today's competitive business landscape, understanding and effectively catering to the diverse needs of customers is crucial for sustainable growth. Customer segmentation is a powerful strategy that involves dividing a customer base into distinct groups based on similar characteristics, behaviors, or preferences. By doing so, businesses can tailor their marketing, product development, and customer service efforts to better meet the specific needs of each segment.

**Objective**: The goal of this project is to leverage data science techniques to perform customer segmentation for a given business. The identified segments should provide actionable insights that enable the company to enhance its marketing strategies, improve customer satisfaction, and drive overall business success.

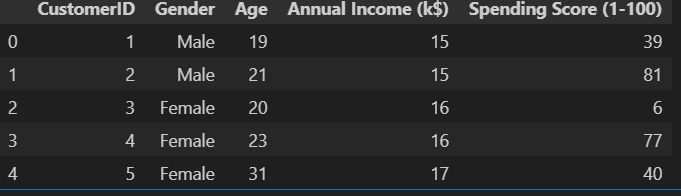
**Dataset**: The dataset for this project will include relevant customer data such as

Data sets based on Customer‘s information

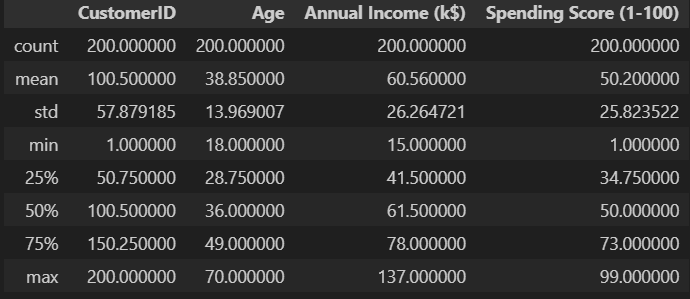
df=pd.read\_csv('/kaggle/input/mall-customers/Mall\_Customers.csv')

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

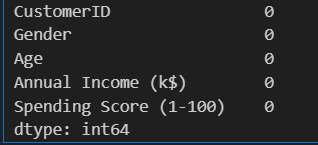
df.head()



df.describe()



df.isnull().sum()



df.drop(['CustomerID'],axis=1,inplace=True)

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

n = 0

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

n +=1

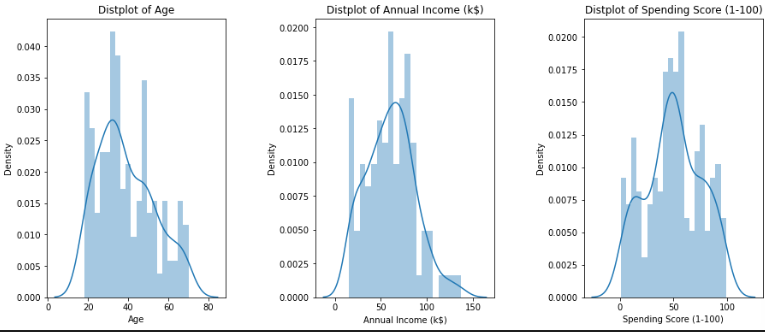
plt.subplot(1,3,n)

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

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

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

plt.show()



plt.figure(figsize=(15,5))

sns.countplot(y='Gender',data=df)

plt.show()



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

n = 0

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

n +=1

plt.subplot(1,3,n)

sns.set(style="whitegrid")

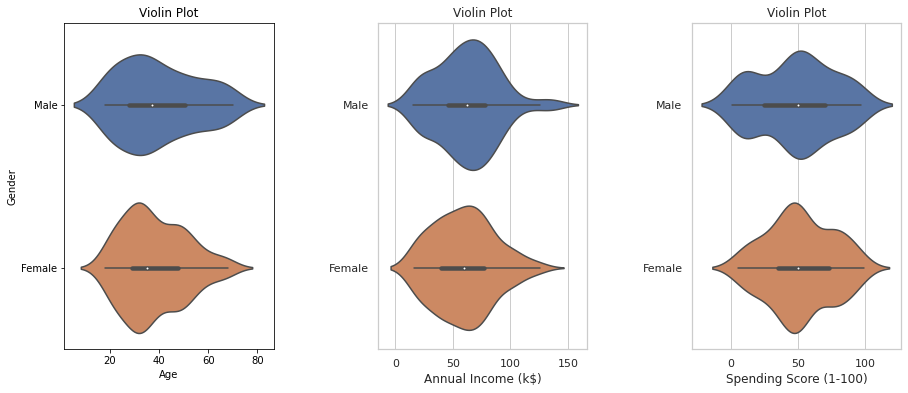
plt.subplots\_adjust(hspace=0.5,wspace=0.5)

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

plt.ylabel('Gender' if n== 1 else '')

plt.title('Violin Plot')

plt.show()



age\_18\_25 = df.Age[(df.Age >=18) & (df.Age <= 25)]

age\_26\_35 = df.Age[(df.Age >=26) & (df.Age <= 35)]

age\_36\_45 = df.Age[(df.Age >=36) & (df.Age <= 45)]

age\_46\_55 = df.Age[(df.Age >=46) & (df.Age <= 55)]

age\_55\_above = df.Age[(df.Age >= 56)]

age\_x =["18-25","26-35","36-45","46-55","55+"]

age\_y = [len(age\_18\_25.values),len(age\_26\_35.values),len(age\_36\_45),len(age\_46\_55),len(age\_55\_above)]

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

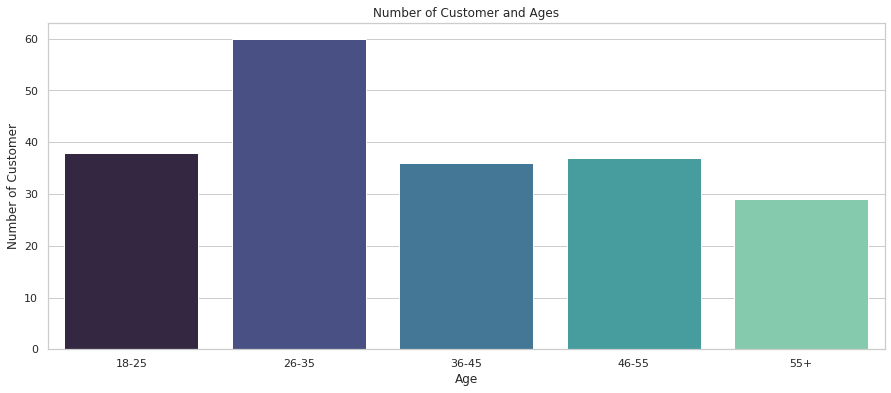
sns.barplot(x=age\_x, y=age\_y,palette = "mako")

plt.title("Number of Customer and Ages")

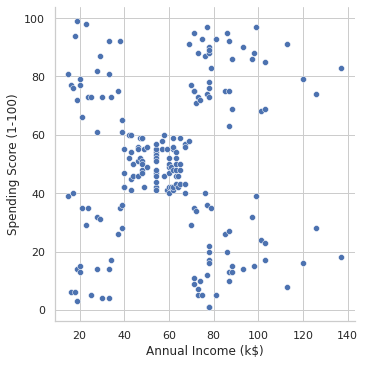
plt.xlabel("Age")

plt.ylabel("Number of Customer")

plt.show()



sns.relplot(x="Annual Income (k$)",y = "Spending Score (1-100)",data=df)



s\_21\_40 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 21) & (df["Spending Score (1-100)"] <= 40)]

ss\_41\_60 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 41) & (df["Spending Score (1-100)"] <= 60)]

ss\_61\_80 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 61) & (df["Spending Score (1-100)"] <= 80)]

ss\_81\_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(ss\_1\_20.values),len(ss\_21\_40.values),len(ss\_41\_60.values),len(ss\_61\_80.values),len(ss\_81\_100.values)]

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

sns.barplot(x=ssx,y=ssy, palette="rocket")

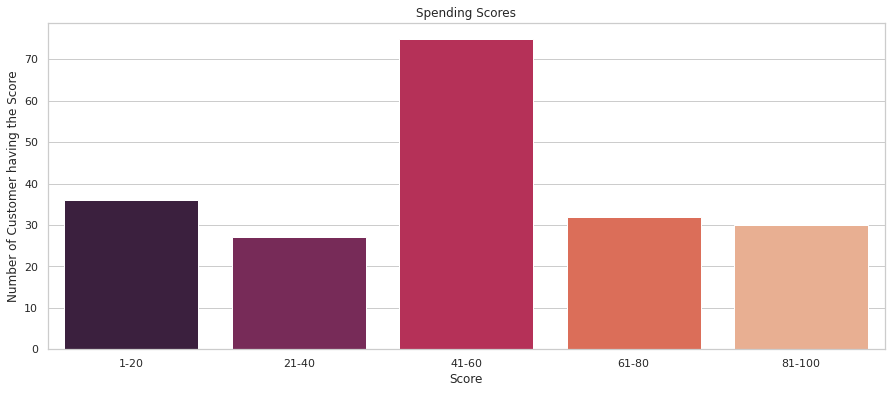
plt.title("Spending Scores")

plt.xlabel("Score")

plt.ylabel("Number of Customer having the Score")

plt.show()ss\_1\_20 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 1) & (df["Spending Score (1-100)"] <= 20)]

S



ai\_0\_30 = df["Annual Income (k$)"][(df["Annual Income (k$)"] >= 0) & (df["Annual Income (k$)"] <= 30)]

ai\_31\_60= df["Annual Income (k$)"][(df["Annual Income (k$)"] >=31)& (df["Annual Income (k$)"] <=60)]

ai\_61\_90= df["Annual Income (k$)"][(df["Annual Income (k$)"] >=61)& (df["Annual Income (k$)"] <=90)]

ai\_61\_90=df["Annual Income (k$)"][(df["Annual Income (k$)"] >=91)& (df["Annual Income (k$)"] <=120)]

ai\_121\_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(ai\_0\_30.values),len(ai\_31\_60.values),len(ai\_61\_90.values),len(ai\_61\_90.values),len(ai\_121\_150.values)]

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

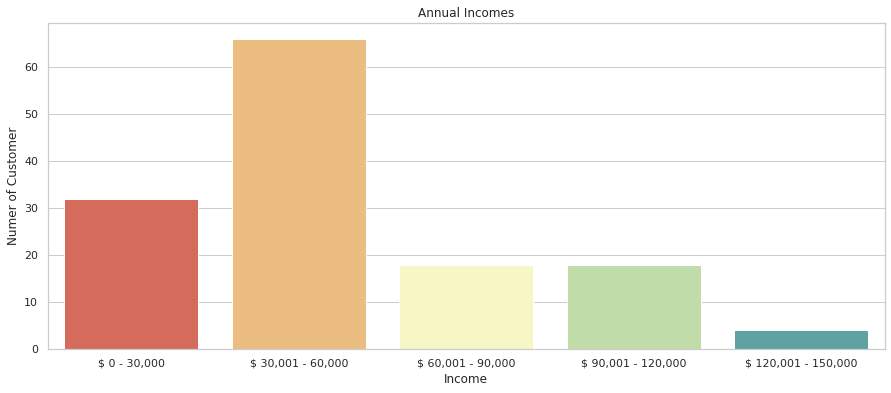
sns.barplot(x=aix,y=aiy,palette="Spectral")

plt.title("Annual Incomes")

plt.xlabel("Income")

plt.ylabel("Numer of Customer")

plt.show()



X1 = df.loc[:,["Age","Spending Score (1-100)"]].values

from sklearn.cluster import KMeans

wcss=[]

for k in range(1,11):

kmeans = KMeans(n\_clusters = k, init = "k-means++")

kmeans.fit(X1)

wcss.append(kmeans.inertia\_)

plt.figure(figsize =( 12,6))

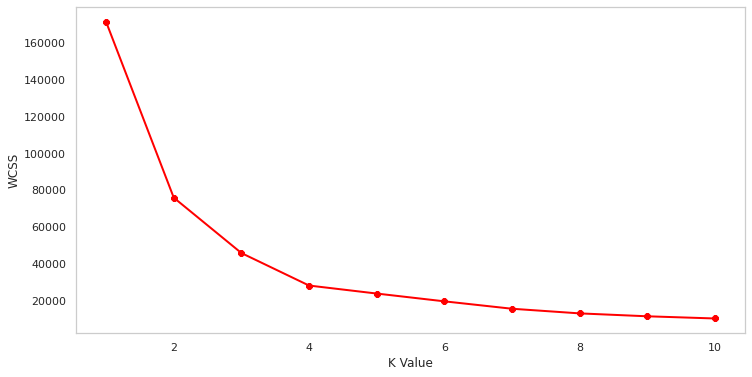
plt.grid()

plt.plot(range(1,11),wcss,linewidth=2,color="red",marker="8")

plt.xlabel("K Value")

plt.ylabel("WCSS")

plt.show()



kmeans = KMeans(n\_clusters=4)

label = kmeans.fit\_predict(X1)

print(label)

[1 2 0 2 1 2 0 2 0 2 0 2 0 2 0 2 1 1 0 2 1 2 0 2 0 2 0 1 0 2 0 2 0 2 0 2 0 2 0 2 3 2 3 1 0 1 3 1 1 1 3 1 1 3 3 3 3 3 1 3 3 1 3 3 3 1 3 3 1 1 3 3 3 3 3 1 3 1 1 3 3 1 3 3 1 3 3 1 1 3 3 1 3 1 1 1 3 1 3 1 1 3 3 1 3 1 3 3 3 3 3 1 1 1 1 1 3 3 3 3 1 1 1 2 1 2 3 2 0 2 0 2 1 2 0 2 0 2 0 2 0 2 1 2 0 2 3 2 0 2 0 2 0 2 0 2 0 2 0 2 3 2 0 2 0 2 0 2 0 1 0 2 0 2 0 2 0 2 0 2 0 2 0 2 1 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2]

print(kmeans.cluster\_centers\_)

[[43.29166667 15.02083333] [27.61702128 49.14893617] [30.1754386 82.35087719] [55.70833333 48.22916667]]

plt.scatter(X1[:,0],X1[:,1],c=kmeans.labels\_,cmap='rainbow')

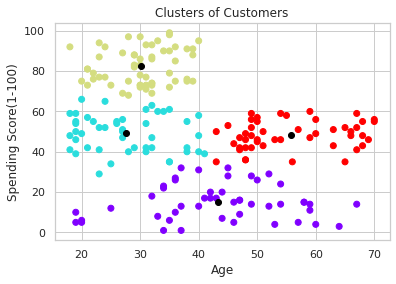
plt.scatter(kmeans.cluster\_centers\_[:,0],kmeans.cluster\_centers\_[:,1],color='black')

plt.title('Clusters of Customers')

plt.xlabel('Age')

plt.ylabel('Spending Score(1-100)')

Plt.show



X2 = df.loc[:,["Annual Income (k$)","Spending Score (1-100)"]].values

from sklearn.cluster import KMeans

wcss=[]

for k in range(1,11):

kmeans = KMeans(n\_clusters = k, init = "k-means++")

kmeans.fit(X2)

wcss.append(kmeans.inertia\_)

plt.figure(figsize =( 12,6))

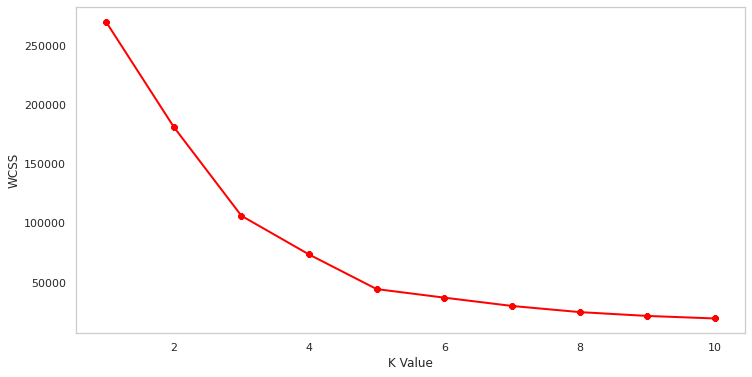
plt.grid()

plt.plot(range(1,11),wcss,linewidth=2,color="red",marker="8")

plt.xlabel("K Value")

plt.ylabel("WCSS")

plt.show()



The above datasets are based on Customer’s personal information

**Customer financial status**

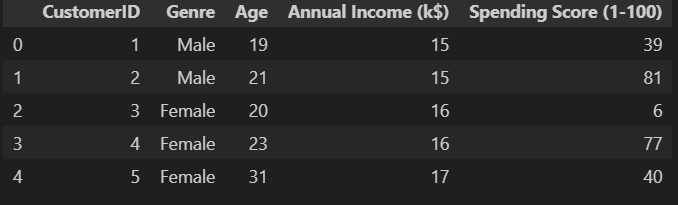
#Reading Data

train\_path = "/kaggle/input/mall-customers/Mall\_Customers.csv"

train\_data = pd.read\_csv(train\_path)

#Checking the data

train\_data.head()

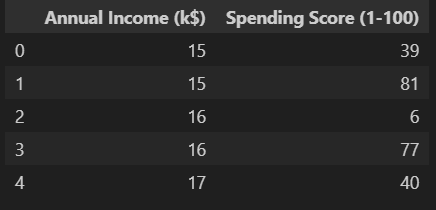


#Choosing potential features for clustering such as Annual Income and Spending Score

train\_data = train\_data.drop(columns=["CustomerID","Genre","Age"],axis=1)

#Checking the data after dropping columns

train\_data.head()



#Standardazing the dataset

sc = StandardScaler()

train\_data = sc.fit\_transform(train\_data)

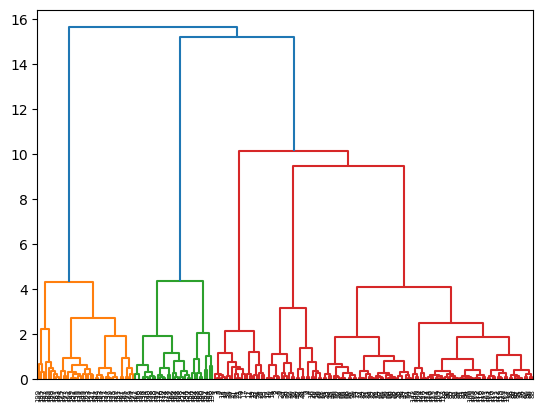
train\_data

#Dendogram

linkage\_data = linkage(train\_data, method='ward', metric='euclidean')

dendrogram(linkage\_data)

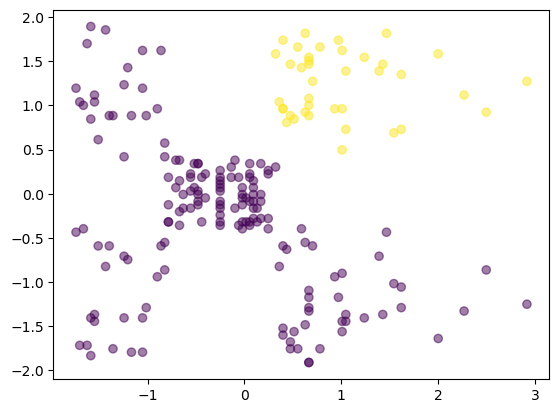
plt.show()



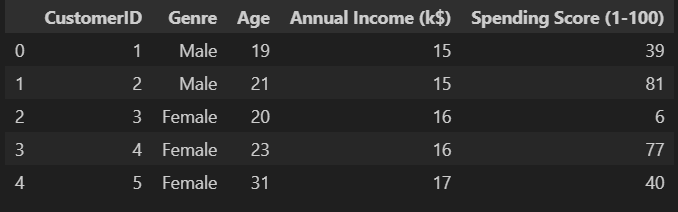
#Hierarchial Clustering Model

hierarchical\_cluster = AgglomerativeClustering(n\_clusters=2, affinity='euclidean', linkage='ward')

labels = hierarchical\_cluster.fit\_predict(train\_data)



**#Choosing potential features for clustering such as Annual Income and Spending Score**



#Checking the data after dropping columns

train\_data.head()

train\_data.columns

#Standardazing the dataset

sc = StandardScaler()

train\_data = sc.fit\_transform(train\_data)

train\_data

#Choosing the optimal value of K

inertias = []

k = []

for i in range(1,10):

model = KMeans(n\_clusters=i,verbose=1)

model.fit(train\_data)

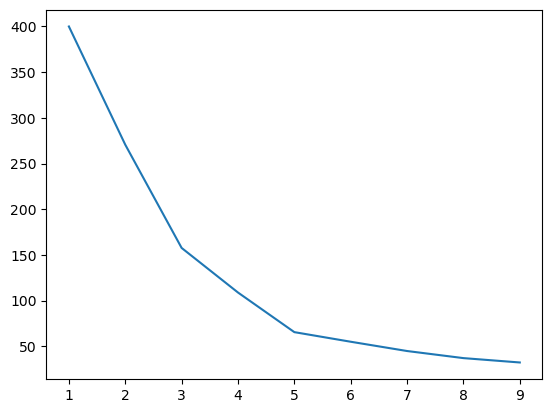
k.append(i)

inertias.append(model.inertia\_)

#Elbow Method using Inertia

plt.plot(k,inertias)

#As there is a bend at 5 the optimal value of k is 5



#Designing the final model after choosing optimal value of k

final\_model = KMeans(n\_clusters=5)

final\_model.fit(train\_data)

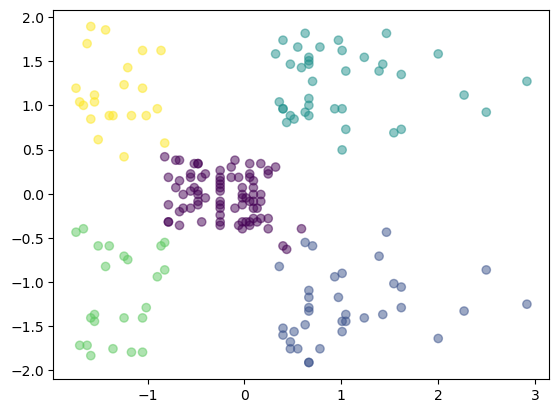
#Visualizing the clusters of KMeans

label = final\_model.fit\_predict(train\_data)

xs = train\_data[:,0]

ys = train\_data[:,1]

plt.scatter(xs,ys,c=label,alpha=0.5)



**Applying K-means to mall dataset**

#Plotting scatter plot of clusters along with their highlighted clusters.

sns.set\_style('whitegrid')

plt.figure(figsize=(10,6))

plt.scatter(X[y\_means==0,0],X[y\_means==0,1],s=50,c='red',label='Cluster 1',marker='\*') #X[y\_means==0,0] for x-coordinates for cluter1,X[y\_means==0,1] for y-coordinates ,s for size of datapoint

plt.scatter(X[y\_means==1,0],X[y\_means==1,1],s=50,c='blue',label='Cluster 2',marker='\*')

plt.scatter(X[y\_means==2,0],X[y\_means==2,1],s=50,c='green',label='Cluster 3',marker='\*')

plt.scatter(X[y\_means==3,0],X[y\_means==3,1],s=50,c='brown',label='Cluster 4',marker='\*')

plt.scatter(X[y\_means==4,0],X[y\_means==4,1],s=50,c='magenta',label='Cluster 5',marker='\*')

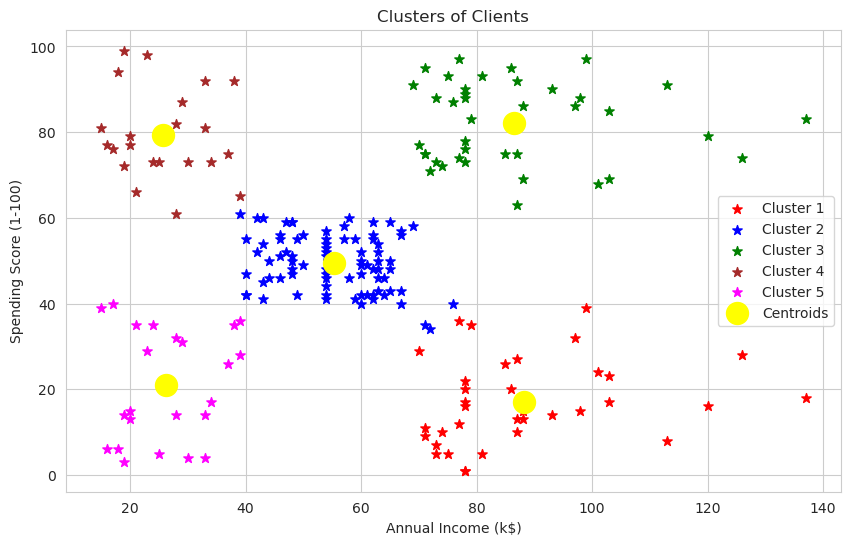
plt.scatter(km.cluster\_centers\_[:,0],km.cluster\_centers\_[:,1],s=250,c='yellow',label='Centroids') #Centroids are highlighted with bigger size

plt.title('Clusters of Clients')

plt.xlabel('Annual Income (k$)')

plt.ylabel('Spending Score (1-100)')

plt.legend();



sns.set\_style('darkgrid')

plt.figure(figsize=(10,6))

plt.scatter(X[y\_means==0,0],X[y\_means==0,1],s=50,c='red',label='Careful')

plt.scatter(X[y\_means==1,0],X[y\_means==1,1],s=50,c='blue',label='Standard')

plt.scatter(X[y\_means==2,0],X[y\_means==2,1],s=50,c='green',label='Target')

plt.scatter(X[y\_means==3,0],X[y\_means==3,1],s=50,c='brown',label='Careless')

plt.scatter(X[y\_means==4,0],X[y\_means==4,1],s=50,c='magenta',label='Sensible')

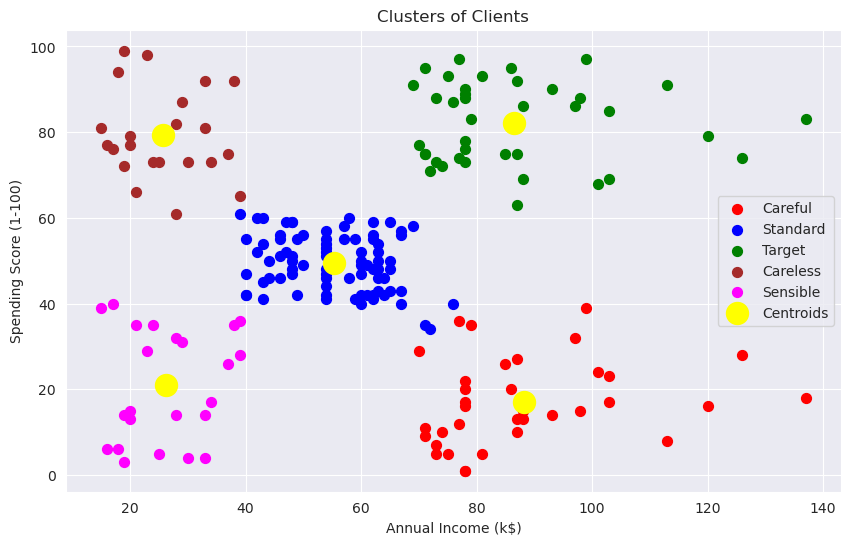
plt.scatter(km.cluster\_centers\_[:,0],km.cluster\_centers\_[:,1],s=250,c='yellow',label='Centroids')

plt.title('Clusters of Clients')

plt.xlabel('Annual Income (k$)')

plt.ylabel('Spending Score (1-100)')

plt.legend();



[MLXTEND]

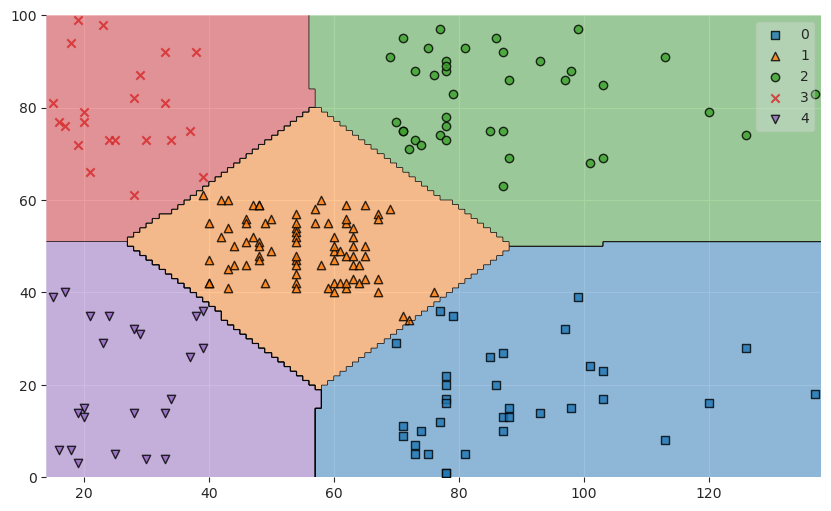
# library supported by Sebastian Raschka for visualizing clusters and many other fuctions

# linear boudries Visualization

from mlxtend.plotting import plot\_decision\_regions

plt.figure(figsize=(10,6))

plot\_decision\_regions(X,y\_means,clf=km);



**Conclusion**

In conclusion, customer segmentation using data science is a powerful strategy that enables businesses to gain a deeper understanding of their customer base and tailor their marketing and service efforts accordingly. By leveraging advanced analytics and machine learning techniques, companies can divide their diverse customer pool into distinct segments based on common characteristics, behaviors, and preferences. This not only facilitates more personalized and targeted marketing campaigns but also enhances overall customer satisfaction and loyalty.

Through the application of data-driven insights, businesses can optimize their resources by directing efforts towards segments that are most likely to respond positively. This approach not only maximizes the effectiveness of marketing initiatives but also helps in the efficient allocation of resources and budget.

Furthermore, customer segmentation fosters adaptability by allowing organizations to identify emerging trends and shifts in consumer behavior promptly. This agility is crucial in today's dynamic business environment, where customer preferences and market conditions can change rapidly.

In the long run, the strategic implementation of customer segmentation contributes to improved customer retention, acquisition, and overall profitability. As businesses continue to accumulate vast amounts of data, the role of data science in customer segmentation will remain pivotal, providing actionable insights that drive informed decision-making and sustainable business growth. It is an iterative process, and organizations that embrace the continuous refinement of their segmentation strategies will be better positioned to stay ahead in the competitive landscape. Ultimately, customer segmentation through data science is not just a tool; it is a dynamic and integral part of modern business strategies, offering a pathway to deeper customer understanding and enhanced business performance.