Customer Segmentation Using K Means

Neil Coleman RUID: 151007807
Aravinda Reddy Dandu RUID: 198006834
Chaitanya Sharma Domudala RUID: 198009015
Andrey Efremchev RUID: 172001181

***Abstract*— The tone of modern period is innovation, where everybody is involved into rivalry to be superior to others. The present business run based on such innovation having potential to enchant the customers with the products, yet with such an enormous pontoon of items leave the consumers bewildered, what to purchase and what to not and furthermore the business are puzzled about what areas to focus to sell their products. This is the place Machine Learning becomes possibly the most important factor, different algorithms are applied for unwinding the concealed designs in the information for better dynamic for what's to come.This thought of which segment to target is made unequivocal by applying segmentation. The path towards sectioning the customers with similar behaviours into a same segment and with different patterns into different segments is called customer segmentation. In this paper, k-Means clustering algorithm has been implemented to segment the customers and finally compare the results of clusters obtained.**

I. Project Description

Similarity is a metric that mirrors the quality of relationship between two data objects. Clustering is primarily utilized for exploratory data mining. K-Means falls under the class of centroid-based clustering.This centroid might not necessarily be an individual from the dataset. Centroid-based clustering is an iterative algorithm in which the thought of similitude is determined by how close a data point is to the centroid of the cluster. k-means will help in partitioning the customers into mutually exclusive groups, for instance, into 3 clusters. The customers in each cluster are similar to each other demographically, utilizing which we can make a profile for each group, considering the common characteristics of each cluster. For example, the 3 clusters can be:

- AFFLUENT, EDUCATED AND OLD AGED
- MIDDLE AGED AND MIDDLE INCOME
- YOUNG AND LOW INCOME

The project has four phases: Requirement Gathering, Analytic Approach and Data Understanding Stage, Design Stage, Infrastructure Implementation Stage, and User Interface Stage. And our implementation language is Python.

A. Stage1 - ***Requirement Gathering, Analytic Approach and Data Understanding Stage.***

These days, individuals' decisions of banking have been incredibly enhanced by numerous factors. In any case, now and then it is hard for users to settle on choice on what to invest on despite huge measure of various policies accessible. In a similar manner, banks also face the difficulty of choosing right customer to promote their products. In numerous ways to deal with fathom this "Data Overload" issue, it is a successful strategy to change users from perspective of active requesting to passively accepting recommendations. The viable solution for this is recommending products based on Customer Segmentation. The segmentation helps in suggesting the product or information that the user might be keen on, based on the user's demand characteristics, historical behaviour attributes, and intrigue qualities, and then pushes it to the user.

In this project, we center around the use of Customer Segmentation in the field of Banking sector. It embraces machine learning technology and applies a K-Means algorithm on the customer demographic subtleties. The data set is from a UK Bank has been utilized. In this algorithm, the system adopts customer-based collaborative filtering.

So to summarize our deliverables for Stage 1 here:

- The general framework depiction: A Customer Segmentation system to cluster users utilizing KMeans algorithm.
- We have three groups of users: Each user is distinguished by a Customer Id.
- The real world scenarios:
  - Banks and numerous different kinds of monetary organizations characterize their customers and attempt to perceive their behavioural structure which incorporates on the off chance that they will pay their debts by all means.
  - Mobile operators have millions of users whom the the organization attempts to make them feel exceptional. The competition is high and regaining a customer once left is hard. Subsequently, customer segmentation in the mobile world is at extreme significance.
  - E-Commerce group has the most elevated potential in capitalizing on customer segmentation. As everything is devitalised, the owner of the site will have access to very detailed information about his customer base.
  - Customer segmentation will help NGOs in making progressively customized strategies and lift the reserve for more note worthy's benefit.
- Project Time line :
  - Time Line
    - Stage 1 - Requirements Gathering, Analytic Approach and Data Understanding Stage: April 16 to April 19
    - Stage 2 - The Design: April 21 to 25
    - Stage 3 -The Implementation Stage: April 27 to May 3
    - Stage 4 -User Interface: May 4 to May 5

	Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Defaulted	Address	DebtIncomeRatio
0	1	41	2	6	19	0.124	1.073	0.0	NBA001	6.3
1	2	47	1	26	100	4.582	8.218	0.0	NBA021	12.8
2	3	33	2	10	57	6.111	5.802	1.0	NBA013	20.9
3	4	29	2	4	19	0.681	0.516	0.0	NBA009	6.3
4	5	47	1	31	253	9.308	8.908	0.0	NBA008	7.2

Fig. 1. Data Snippet

B. *Stage2 - **The Design Stage.***

This project involves clustering credit card customers based on different characteristics, such as age, years of education, and number of years of work experience. The steps of the project are as follows.

1. Download data on customer segmentation for credit card history. The data is in the CSV format.
2. Since k-means clustering is based on Euclidean distance, remove any variables for which the Euclidean distance would not be a meaningful measure of similarity. Since Euclidean distance is not meaningful for analyzing the address variable, do not use the address variable in the analysis
3. Decide the number of clusters to divide the data into. We chose to divide the data into 3 clusters because this seems like the most natural fit for analyzing the data.
4. Run the k-means clustering algorithm in order to generate the clusters.
5. Create a 3-dimensional plot showing the clusters generated by the k-means clustering. Make each of the clusters its own color in order to visualize the differences between the clusters.
6. Determine what variables in the dataset are most different among the three clusters. This enables the user to determine which predictor variables in the dataset are most important in differentiating among the customers.
7. Make a Flask web app that performs the k-means clustering analysis mentioned in steps 2-5 on any dataset that is similarly formatted to the one downloaded in step 1. The Flask web app will have an input text box where the number of clusters can be entered as well as a place where a file name can be entered.
8. The Flask app in step 7 will generate a plot that shows the clusters that the customers are divided into. Each cluster will be in its own color.

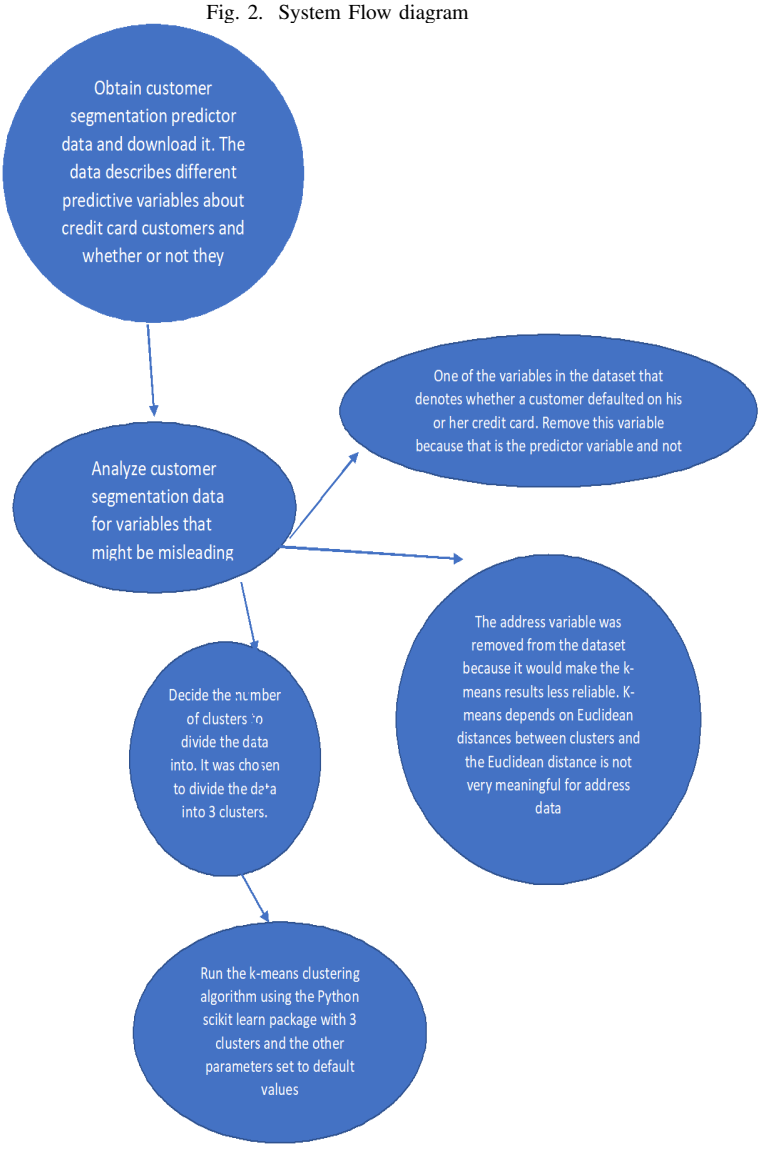
Stage 2 deliverables:

- Short Textual Project Description.
- Flow Diagram.
  See Fig 2.

C. *Stage3 - **The Implementation Stage.***

We mostly utilized Python programming language for this venture, and extraordinarily Numpy, Sklearn, pandas and matplotlib bundle in Python for its capacity in logical processing. Here's a basic breakdown of what the algorithm does :

Fig. 2. System Flow diagram



- **Sample Data Input:**
  - Number of Clusters: 4
  - Choose the Customer Segmentation CSV

Here's a basic breakdown of what the algorithm does :

- We input a customer data set and the number of clusters, or interest groups among our customers, that we desire to have after the algorithm is done.
- We make use of the web Page which utilizes the k-mean algorithm with our particular and returns us the clustering.
- These clusters would then be able to be utilized to more profound examine the data. It is essential to take note of that, we don't specify how to cluster the data for the algorithm. Thusly it will simply find customers who are most similar to each other and put them together in groups. Working as such it is

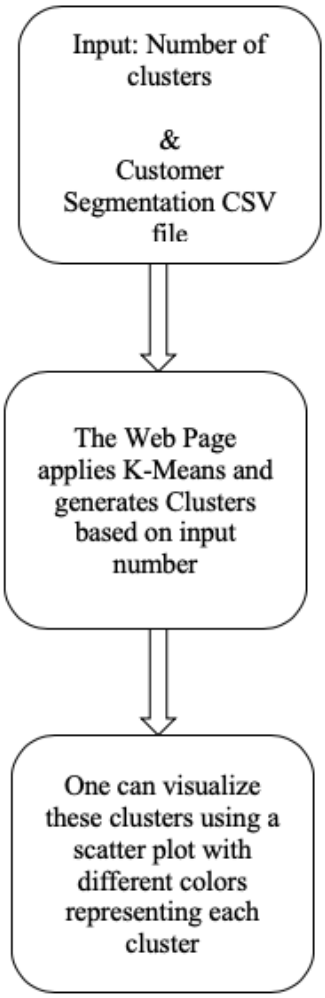
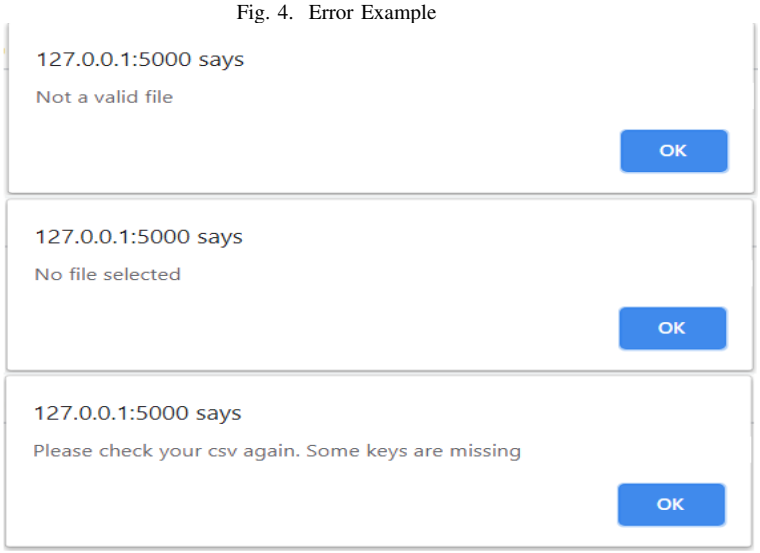


Fig. 3. Implementation Flow

not necessarily true that the clusters we described in the project description will manifest, especially if the data doesn't have the right parameters. What this project doesn't do is coming up with specific clusters based on specified parameters. The user has to still infer why these clusters were formed and then use that knowledge to accompany with the marketing effort. Considering this the user must be very specific and careful with how many clusters they must use and their customer data is sufficiently formatted.

Working code: The code is attached in the Appendix section.

Fig. 4. Error Example



*D. Stage4 - **User Interface Stage.***

Deliverables for Stage 4 are as follows:

- The initial stage to start web App is entering the url which takes to the home page and there you would be asked to enter a few attributes.
  One is the number of clusters needed and the other is the csv file with required headers.
- Model of interaction:
  For the number of clusters needed, a drop-down is given ranging from 1 to 10 and for csv file input option is given which accepts only file with proper headers. On click of submit, two visualization, 2-D and 3-D are shown in the web page.
  User has the option to download a sample csv file using the download sample hyperlink.
- The error messages popping-up on click of submit after given input (along with explanations and examples):
  - The error messages:
    - 1)"No file selected"
    - 2)"Not a valid file"
    - 3)"Please check your csv again. Some keys are missing"
  - The error message explanation (upon which violation it takes place):
    - 1)The file is not present
    - 2)The file is not a csv file
    - 3)File does not contain all the headers needed or some are missing.
  - The error message example according to user(s) scenario(s):
    See Fig 4.
- The visualizations displayed on successful parsing and applying K-means on given data.
  - 2-D Visualization:
    "This Visualization plots Age and Income in X

Fig. 5. 2-D scatter plot

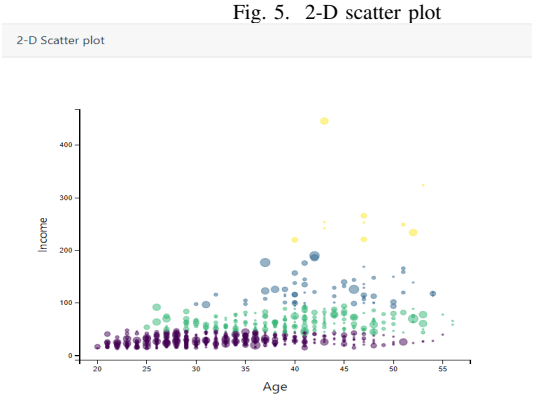
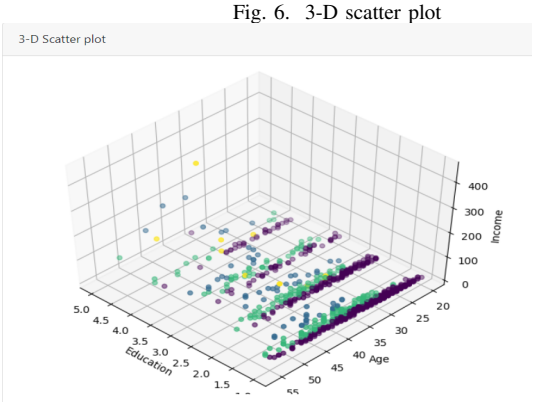


Fig. 6. 3-D scatter plot



and Y axes respectively. Based on the number of clusters chosen, clusters are coloured differently. As the number of clusters increase, colors increase.

  - 3-D Visualization:
    This Visualization plots Education, Age and Income on X, Y and Z axes respectively. Number of clusters will be the same for 2D and 3D.

- The interface showing these two plots:
  - Example
    See Fig 5 and 6.

II. Appendix

A. Code

movielens.py: Load Data

```
"""
Scripts to load data and cluster it(Python)
"""
#!/bin/python
import os
from flask import Flask, render_template,
    jsonify, send_file
from flask import request
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt, mpld3
from sklearn.cluster import KMeans
from mpl_toolkits.mplot3d import Axes3D

def makeplots(clusterNum, filepath):
    # cust = pd.read_csv(
    #
        'https://s3-api.us-geo.objectstorage.softlayer.ne
    #    '/Cust_Segmentation.csv')

    cust = pd.read_csv(filepath)
    cust.head()

    cust.shape
    cust.drop(['Address'], axis=1,
        inplace=True)

    print(cust.shape)
    cust.head()

    from sklearn.preprocessing import
        StandardScaler

    X = cust.values[:, 1:]
    X = np.nan_to_num(X)
    Clus_dataSet =
        StandardScaler().fit_transform(X)
    Clus_dataSet

    # clusterNum = 3
    k_means = KMeans(init="k-means++",
        n_clusters=clusterNum, n_init=12)
    k_means.fit(X)
    labels = k_means.labels_
    print(labels)

    cust["Clus_km"] = labels
    cust.head(5)

    cust.groupby('Clus_km').mean()

    sfig, ax = plt.subplots()
    area = np.pi * (X[:, 1]) ** 2
    b = area.tolist()
    c = labels.astype(np.float).tolist()
    ax.scatter(X[:, 0], X[:, 3], s=b, c=c,
        alpha=0.5)
    plt.xlabel('Age', fontsize=18)
    plt.ylabel('Income', fontsize=16)

    # plt.show()

    temp = mpld3.fig_to_html(sfig)
    fig = plt.figure(1, figsize=(8, 6))
    plt.clf()
    ax = Axes3D(fig, rect=[0, 0, .95, 1],
        elev=48, azim=134)

    plt.cla()
    # plt.ylabel('Age', fontsize=18)
    # plt.xlabel('Income', fontsize=16)
    # plt.zlabel('Education', fontsize=16)
    ax.set_xlabel('Education')
    ax.set_ylabel('Age')
    ax.set_zlabel('Income')

    ax.scatter(X[:, 1], X[:, 0], X[:, 3],
        c=labels.astype(np.float))
    try:
       os.remove('D:/Projects/K_means/foo.png')
```

```
    except:
        print('unable to delete')
    plt.savefig('D:/Projects/K_means/foo.png')
    plt.close(sfig)
    plt.close(fig)
    return temp
    # print(temp)

# App End points for server
@app.route('/')
def index():
    return render_template('kmeans.html')


@app.route('/plot/<clusternum>',
     methods=['GET', 'POST'])
def plot(clusternum):
    isthisFile = request.files.get('file')
    htmlCon = twodeeplot(int(clusternum),
        isthisFile)
    return htmlCon


@app.route('/plotthreed', methods=['GET',
    'POST'])
def plotthreed():
    filename = 'foo.png'
    return send_file(filename,
        mimetype='image/png')


@app.route('/getFile', methods=['GET',
    'POST'])
def getFile():
    filename = 'static/Sample_data.csv'
    return send_file(filename,
        mimetype='text/csv')


if __name__ == '__main__':
    app.run(debug=True)
```

III. References

- https://scikit-learn.org/
- https://www.papaparse.com/
- https://flask.palletsprojects.com/
- https://getbootstrap.com/
- https://matplotlib.org/
- https://www.learndatasci.com/tutorials/k-means-clustering-algorithms-python-intro/
- https://medium.com/ml-research-lab/part-1-data-science-methodology-from-problem-to-approach-e2d05e7afc6b