

B.Tech. Project Report

entitled

JOB RECOMMENDATION SYSTEM

Submitted in partial fulfillment

for

the award of the degree of

Bachelor of Technology

by

Mr. Ashish Lalwani(18UCS194)

Mr. Akshat Jain(18UCS071)

Supervisor

Dr. Suvidha Tripathi



December, 2020

Department of Computer Science & Engineering

THE LNM
INSTITUTE OF INFORMATION TECHNOLOGY,
JAIPUR INDIA

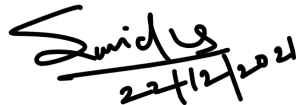
Declaration

We solemnly declare that the work being presented in this project report entitled “JOB RECOMMENDATION SYSTEM” is an authentic record of my own work carried out during the period 2020 – 2021 under the supervision of Dr. Suvidha Tripathi.

Neither the source code there in, nor the content of the project report have been copied or downloaded from any other source. I understand that my result grades would be revoked, if later it is found to be so.

C E R T I F I C A T E

This is to certify that the submitted B.Tech. project report entitled “JOB RECOMMENDATION SYSTEM” is an official record of actual work carried out by Mr. ASHISH LALWANI (18UCS194), Mr. AKSHAT JAIN (18UCS071) under my supervision and guidance in the institute. To the best of my knowledge, the report embodies the record of an authentic work of the candidate, has duly been completed, is up to the desired standard for the purpose of which it is submitted, and fulfills the requirement of the ordinance relating to the B.Tech. degree of the institute.



(Dr. Suvidha Tripathi)

**Department of CSE,
The LNMIIT,
Jaipur – 302031,
India**

**Head of Department,
Department of CSE,
The LNMIIT,
Jaipur – 302031,
India**

Acknowledgements

We would like to express our heartfelt gratitude to Our Supervisor, Dr. Suvidha Tripathi, for providing us with the wonderful opportunity to work on this wonderful project on the topic of Application of Recommendation Systems. She also assisted us in conducting extensive research and introducing us to many new things, for which we are extremely grateful. Second, we'd want to express our gratitude to all of our friends who assisted us in completing this project within the time constraints.

Abstract

For any Project We should optimize the technique and resources which we are using So we can get the best output from any process or task it's is very important to compare the resources which we are going to use in our project Through this project we are trying to find best suitable technique to implement our idea to give them a platform which can help to give better recommandation service This comparison will help us to find the better and optimize solution and to choose our algorithm which will give as more accurate and efficient result so can a user can get the best outcome So our problem statement is to compare the two different algorithm/technique for better outcome Sustainable development requires critical thinking, which is encouraged by the All Protocol to pay attention to the definition of effective and accurate systems. For a better idea we must find another option as well. Our goal of the project is to find the best, most efficient and accurate method of our project It is necessary to compare you to better understand all aspects of the project Be aware of any potential problems during launch project Determining that, after considering all the priorities, the project is operationalIt is worth noting that the main reason why many people seem to care about money laundering programs. For companies like Amazon, Netflix, and Spotify, complimentary systems drive greater engagement with revenue. But this is an attitude of distrust. The reason these (and other) companies see increased revenue is because they bring real value to their customers - recommendation programs offer an awesome way to customize user content in multidimensional environments. Another reason why data scientists especially need to take care of recommendation systems is that they are a real data science problem. That is, at least by going to the crossroads between software engineering, machinelearning, and math. As we will see, building effective commendation systems requires all these skills

Table of Contents

1 Introduction	1
1.1 The Area of Work	1
1.2 Problem Statement:	1
1.3 Motivation :	2
1.4 Objective :	2
2 Literature Survey	3
2.1 Recruiting Process:	3
2.2 E-Recruiting Platforms :	3
2.3 Categories of E-Recruitment Platform:	4
2.4 Collaborative Filtering Approach	4
2.4.1 Memory-Based CF Methods :	5
2.4.2 Model Based CF-Methods :	5
2.4.3 Characteristics and Challenges of CF	5
3 Data Set,Methodologies and Algorithms Analysis	7
3.1 Data Set Information and Main Source	7
3.2 Detailed View of Data-set :	7
3.2.1 Alternate-Title:	7
3.2.2 Education Experience	8
3.2.3 Knowledge	8
3.2.4 Occupation	9
3.3 Algorithms Implemented	9
3.3.1 K-Means	9
3.3.1.1 About :	9
3.3.1.2 Advantage :	11
3.3.1.3 Dis-Advantage :	11
3.3.2 Random-Forest	11
3.3.2.1 About:	11
3.3.2.2 Advantage :	12

3.3.2.3	Dis-Advantage :	13
4	Implementation and Execution	15
4.1	Overall Flow	15
4.2	Tech Stack	16
4.3	Creating Cluster	16
4.3.1	Data-Prepossessing	16
4.3.2	Fitting the data in model	17
4.3.3	Predicting Clusters	17
4.3.4	Saving The Clustered Data:	18
4.3.5	Predicting The Class Label (or Cluster)	18
4.4	Loading K-means	19
4.4.1	Data-Prepossessing	19
4.4.2	Sample-Prediction	20
4.5	Career Classify Helper	20
4.5.1	Objective	20
4.5.2	The Data-set	21
4.5.3	Importing Necessary Libraries and Loading Experience	21
4.5.4	Input preprocessing (Sample)	22
4.5.5	Predicting the class/Cluster:	25
4.5.6	Data Preprocessing	27
4.5.7	Get user's input on Education Level	28
4.5.8	Get Experience Level	29
4.5.9	Get the current title:	30
4.5.10	Alternate titles	31
4.5.11	Make Predictions Based on probability of users education and ex- perience:	32
4.5.12	Job Descriptions:	33
4.6	Skill Processing	34
4.6.1	Reading the skills	34
4.6.2	Creating new File for sorted skills for each title	35

4.6.3	The Random Forest Classifier	36
4.7	Front-End Development	37
4.7.1	Tech Stack	37
4.7.2	Home Page	37
4.7.3	Find Job Page	37
4.7.4	Trends Page	39
4.7.5	Data Page	39
4.7.6	About Page	40
5	Result And Discussion	41
5.1	Predicted Jobs of predicted cluster:	41
5.2	Result by the Random Forest Classifier:	42
5.3	Evaluation metrics	42
5.3.1	Elbow Method	42
5.3.2	Silhouette method:	44
	Bibliography	45

Chapter 1

Introduction

1.1 The Area of Work

For any Project We should optimize the technique and resources which we are using So we can get the best output from any process or task it's is very important to compare the resources which we are going to use in our project Through this project we are trying to find best suitable technique to implement our idea to give them a platform which can help to give better recommendation service This comparison will help us to find the better and optimize solution and to choose our algorithm which will give as more accurate and efficient result so can a user can get the best outcome So our problem statement is to compare the two different algorithm/technique for better outcome

1.2 Problem Statement:

As Technology trends are Changing with Automation,Cloud,Analytics It is Currently tough for a Fresher to find a right Job Title according to their Potential and skills education and experience for Finding a better Job they end up. Eventually we are trying to help the entire Recruiting Ecosystem to improve Candidate Conversion This is an Unsupervised learning Algorithm which will help us to find a suitable Job for a Potential Candidate. This will help The Job seeker to understand the market Trends

1.3 Motivation :

Sustainable development requires critical thinking, which is encouraged by the All Protocol to pay attention to the definition of effective and accurate systems. For a better idea we must find another option as well. Our goal of the project is to find better and right Job Positions for an aspirant This application gives the perfect work titles and compensation for the title over the US for a client based on the aptitudes, instruction and experience. The client input is bolstered into a machine learning show that has been prepared to utilize the O*NET Data set for the whole US and a proprietary calculation to discover the leading coordinate of titles the client is best suited to. This comes about for the client incorporating the working title, centre errands, advances required and the compensation for the prescribed titles. Too, the most recent patterns for the year 2018, in terms of work development, most elevated developing and declining employments moreover has been shown.

1.4 Objective :

Nowadays Recommendation System widely used in many Technological domains for the recommendation of many services weather it is music shopping, Jobs, movies,Leading Technological companies use recommendation system to optimize their services like recommendation of their product,Movies,friends in their Social Networks To fulfill all the necessities, the foremost methods utilized in recommendation frameworks are collaborative filtering and content-based frameworks. The collaborative filtering does not take into consideration the type of things, nor their qualities. It takes only under consideration the communicated conclusion around the other things in order to create suggestions. In the interim, content-based sifting employs the information it has of the things and their qualities to create suggestions.

Chapter 2

Literature Survey and Theoretical Background

2.1 Recruiting Process:

The primary goal of selection preparation is to hire people that are critical to the company's success . There are two points of view: recruiters' and job seekers'. The scouts decide on a set of prerequisites and limits for aptitudes, talent levels, and degrees to develop the job description. The job seeker, on the other hand, constructs his or her CV by describing his or her educational background, work experience, and skills. The IT backbone for candidate selection extends beyond attracting and locating applicants to choosing and retaining them . The degree of process integration reflects the difficulty of deploying e-recruitment arrangements, as seen in their proposed model, which depicts the interaction between enlisting assignments and divides the selection process into two phases: Both the interest and choice stages

2.2 E-Recruiting Platforms :

E-recruitment could be a way to reach out to a big number of potential job searchers quickly. Since the late 1990s, when rapid economic changes created a large demand for qualified individuals that the labour market could not fully meet, e-recruiting has seen rapid expansion. This development has been fueled by e-recruiting stages such

as company homepages and job portals . According to the websites of the Universal Affiliation of Employment, there are more than 40,000 work destinations around the world, making a distinction between job seekers and recruiters . Companies post open positions on these portals, and job seekers use them to distribute their profiles, resulting in an unlimited amount of job descriptions and candidate profiles being available online. In any event, implementing these e-recruiting steps will result in cost savings, adequacy, and suitability for both recruiters and job seekers

2.3 Categories of E-Recruitment Platform:

We show the six categories of e-recruiting sources displayed by to deliver a better recommendation:

(1) General-purpose work sheets that provide comprehensive internet recruiting services. Rather of searching for jobs by category, such as involvement, location, or instruction, scouts search applicant databases by aptitudes, involvement, inclination, education, salary, or any combination of key words.

(2) Specialty job boards cater to niche markets such as a specific occupation, industry, or field of education, or any combination of specialties;

(3) E-recruiting application benefit providers display a range of services including enrollment, enrollment process administration, instruction, and preparation.

(4) Providers of hybrid recruitment benefits (for case, magazines and Journals)

5)An e-recruiting consortium is a search engine that directs traffic to a member's career website.

(6) A corporate job site is a type of employment source that is most widely used by Fortune 500 businesses, and its use is a regular extension of ebusiness applications

2.4 Collaborative Filtering Approach

One of the most successful ways for developing recommender frameworks is collaborative filtering (CF). It can either be unequivocal, which refers to a user declaring his or her preference for something on a numerical scale such as 1–5, or certain, which refers to

inferring client behaviour or decision to relegate the user preference . CF techniques can operate in environments where getting content is difficult or where it cannot be digested automatically. CF approaches, on the other hand, can make unanticipated suggestions that aren't comparable to the things in the dynamic user's profile yet excite him/her .present illustrations of recommendation systems based on CF techniques

2.4.1 Memory-Based CF Methods :

To generate expectation, a test of a user-item database is used. Each client is a member of a group of users who share a similar interface. When distinguishing the dynamic client's neighbours, the user's expectation for new thing inclinations can be met . We use correlation or other measures to precisely compare clients to one another . The user-based and item-based correlation/similarity measurements are also included in the Memory-based CF methods. By aggregating the watched inclinations of like customers, user-based measures forecast a target user's future inclinations. To begin, the algorithm computes a user similarity score that is based on the vector similarity work. A high proximity score indicates that the two clients have similar preferences . The itembased measures, on the other hand, differ from the userbased measures in that thing likenesses are computed instead of client likeliness.

2.4.2 Model Based CF-Methods :

Could be a method of generating a demonstration from previous data and using it to infer predictions . The improvement of models allows the system to learn and detect complicated patterns using preparation data, and then generate test information expectations. Memory-based CF strategies have flaws, while model-based CF strategies use techniques like Bayesian models, clustering models, and dependency organise to solve them.

2.4.3 Characteristics and Challenges of CF

The most essential feature of CF techniques is that they are completely independent of any machine-readable representation of the objects being proposed, and they perform well for complex objects like sounds and movies, where differences in taste have altered

the range of preferences. On the other hand, CF has a number of significant hurdles, including cold-start issues including information scarcity and ramp-up issues. The issue of data scarcity necessitates the use of accurate data. Users' previous data, such as what they've seen, acquired, or rated, is scarce in many real-world applications because the site is still in its early stages of operation. As a result, it's very feasible that the similarity between any two customers is almost zero, or that the metrics are unreliable.

Chapter 3

Data Set, Methodologies and Algorithms Analysis

3.1 Data Set Information and Main Source

- O*Net Resource Center Available
- https://www.onetcenter.org/db_releases.html *Bureau of Labor Statistics Available :*
<https://www.bls.gov>
- Glassdoor (2017) Local Pay Reports: Historical Data Available: <https://www.glassdoor.com/research/datasets/>

3.2 Detailed View of Data-set :

3.2.1 Alternate-Title:

The following properties of each career title are measured and included in this data:

Title: Job Title

Alternate title: Multiple different which can be related to that particular job title

3.2. DETAILED VIEW OF DATA-SET :

	Title	Alternate Title
0	Chief Executives	Aeronautics Commission Director
1	Chief Executives	Agricultural Services Director
2	Chief Executives	Alcohol and Drug Abuse Assistance Program Admi...
3	Chief Executives	Arts and Humanities Council Director
4	Chief Executives	Bakery Manager

●

3.2.2 Education Experience

The following properties of each career title are measured and included in this data:

- Title: Job Title
- Education: Shares whether it is High School Diploma or Bachelors or Masters Degree (in string form)
- Education Level: Number mapping to Education Level
- Data Value Education: Weight for Education Level for that specific title
- Work Experience Level: Number mapping for Experience Level
- Work Experience: Experience (in words) whether it is in months or years
- Data Value Experience: Weight for Education Level for that specific title

3.2.3 Knowledge

The following properties of each career title are measured and included in this data:

- Title: Job Title
- Element Name: A particular skill relevant to the job title

- Importance: Score of Importance of that skill in the job title

	Title	Education	Education Level	Data Value_Education	Work Experience Level	Work Experience	Data Value_Experience
0	Chief Executives	Less than High School Diploma	1	0.0	1	0	0.0
1	Chief Executives	Less than High School Diploma	1	0.0	2	>0 and upto 1 month	0.0
2	Chief Executives	Less than High School Diploma	1	0.0	3	>1 month and upto and including 3 month	0.0
3	Chief Executives	Less than High School Diploma	1	0.0	4	>3 month and upto and including 6 month	0.0
4	Chief Executives	Less than High School Diploma	1	0.0	5	>6 month and upto and including 1 year	0.0

3.2.4 Occupation

The following properties of each career title are measured and included in this data:

- Title: Job Title
- Element Name: A particular skill relevant to the job title
- Importance: Score of Importance of that skill in the job title

	Title	Element Name	Importance
0	Chief Executives	Administration and Management	4.75
1	Chief Executives	Clerical	2.66
2	Chief Executives	Economics and Accounting	3.70
3	Chief Executives	Sales and Marketing	3.23
4	Chief Executives	Customer and Personal Service	4.09

3.3 Algorithms Implemented

3.3.1 K-Means

3.3.1.1 About :

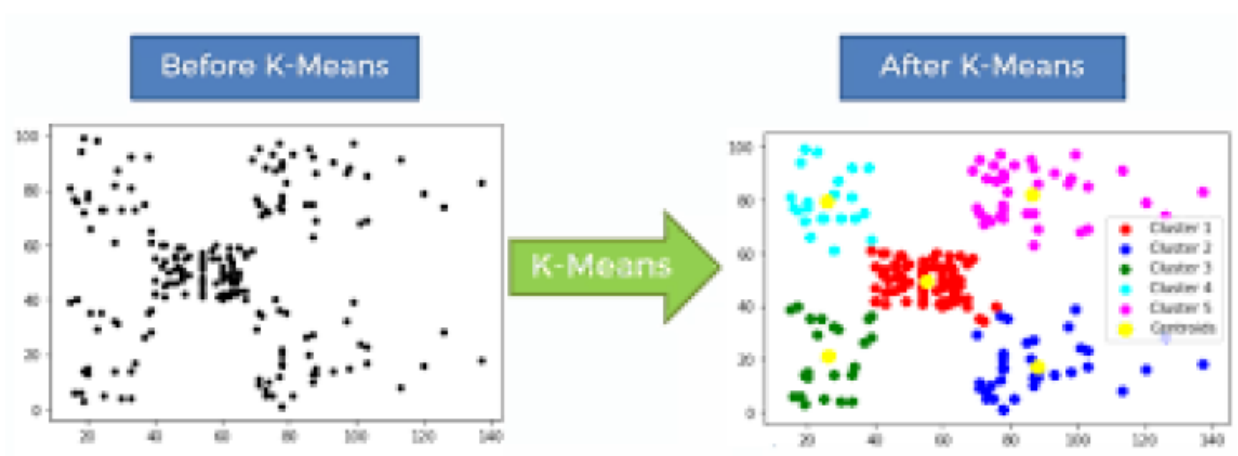
We are provided a data set of items with specific characteristics and values for these characteristics (similar to a vector). The goal at hand is to sort the things into groups.

3.3. ALGORITHMS IMPLEMENTED

We'll utilize the kMeans clustering algorithm, which is an unsupervised learning algorithm, to accomplish this. (Thinking of items as points in an n-dimensional space will assist.) The algorithm divides the objects into k groups based on their similarity. We'll utilise a similarity criterion (such as euclidean distance) to calculate that similarity.

The Algorithm :

1. First of all, k random points are initialized. Let's call them means
2. Each point is now assigned to its nearest mean, and accordingly, the mean is updated.
3. To update the means, we take averages of the items assigned to the previous means.
4. This process is repeated for a certain number of iterations, till the optimal number of clusters is achieved.



The following are some of the most commonly used similarity measures

1. Cosine distance: The cosine of the angle between the point vectors of two points in n-dimensional space is determined by the cosine distance.
2. Manhattan distance: The total of the absolute differences between the coordinates of the two data points is calculated.
3. Minkowski distance: The generalised distance metric is also known as the Minkowski distance. Both ordinal and quantitative variables can be used with it

3.3.1.2 Advantage :

1. Easy Implementation, as compared to other clustering algorithms
2. A point of convergence is always found
3. Is adaptive to different problems. Fits in every shapes of colors, like elliptical, parabolic ,etc.
4. Faster Computation than hierarchical clustering, for small values of k.

3.3.1.3 Dis-Advantage :

1. Manually selecting 'k'
2. Being reliant on starting values.
3. Clustering data of varying sizes and density.
4. Outliers are clustered
5. Scaling in relation to the number of dimensions

3.3.2 Random-Forest

3.3.2.1 About:

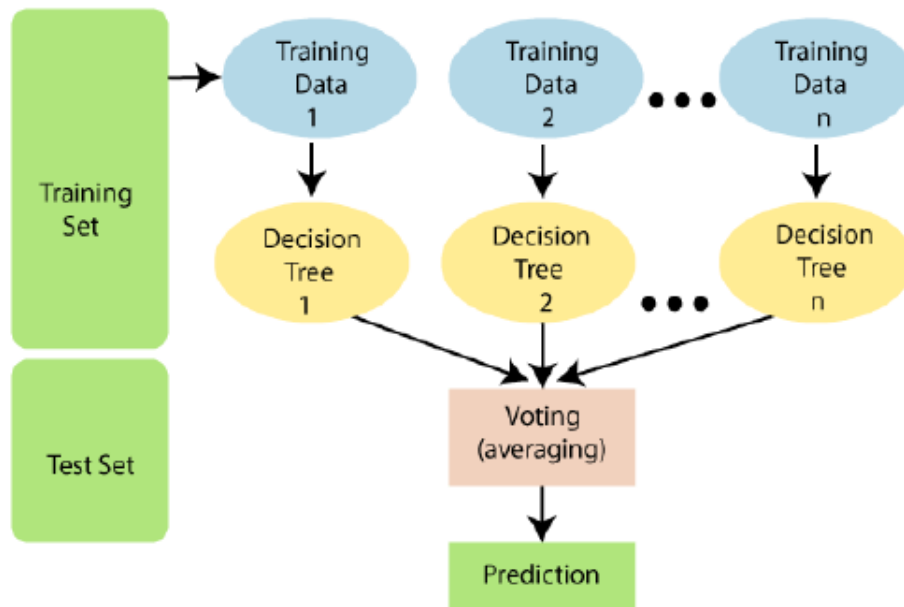
The Random Forest Classifier is a widely used supervised machine learning algorithm. It is used for both classification as well as regression(but not recommended for regression). It follows the concept of ensemble learning, in which, we use multiple different classifying algorithms to achieve a learning task, hence increasing the performance of the model. Just from its name, one can guess what the classifier does. The classifier divides the dataset into various subsets. For each subset, an individual decision tree is created. The average (in most cases, maybe some other parameter in some cases) of the predictions made by all the trees is taken to increase the predictive accuracy. Accuracy is directly proportional to the number of trees, and also overfitting is eliminated.

Requirements :

3.3. ALGORITHMS IMPLEMENTED

1. The features must be given some signal order for their respective models so that there is no random guessing.
2. The predictions and errors from separate trees must have less correlation

So this is how the algorithm actually works:



3.3.2.2 Advantage :

1. Random Forest can handle both classification and regression problems.
2. It can handle huge datasets with a lot of dimensionality.
3. It improves the model's accuracy and eliminates the problem of overfitting.'

3.3.2.3 Dis-Advantage :

1. It necessitates a significant amount of computational power as well as resources because it constructs several trees and combines their outcomes.
2. It also takes a huge lot of time to train because it forms a number of decision trees to select the class.
3. Despite the fact that random forest may be used for both classification and regression tasks, it is not better suited to regression tasks

Chapter 4

Implementation and Execution

4.1 Overall Flow

- First of all, complete the questionnaire with your Skills, Education, Work, Experience Contact Information The user input is stored into MongoDB.
- Now the K-Means Clustering Machine Learning model processes the user input skills to predict a list of most likely job titles.
- The output from the model is passed into a scoring algorithm using weightages to score user's education and experience against the preprocessed dataset from O*NET.
- The list of predicted titles is sorted in descending order based on the scoring algorithm.
- Additional information such as job description, technology, skills, core tasks, expected and alternate titles is retrieved from the dataset for the sorted job titles and the output file is generated.
- At last, the output file is utilized to display the results

4.2 Tech Stack

- Language: Python
- IDE: Google Collab
- Libraries: pandas, Numpy, Pickle, sklearn
- Machine Learning Algorithms using SkLearn and Pickle, K-Means Clustering, Random Forest Algorithm
- DataBase : MomgoDB

4.3 Creating Cluster

4.3.1 Data-Preprocessing

Loading Data set:

```
import pandas as pd  
skills=pd.read_csv("/content/knowledgecleansed.csv")
```

Dropping the title column:

```
target=skills["Title"]  
data=skills.drop("Title",axis=1)  
feature_name=data.columns  
data.head()
```

	Administration and Management	Clerical	Economics and Accounting	Sales and Marketing	Customer and Personal Service	Personnel and Human Resources	Production and Processing	Food Production	Computers and Electronics	Engineering and Technology	...	English Language	Foreign Language	Fine Arts	History and Archeology	Philosophy and Theology
0	6.23	3.50	4.36	3.90	5.55	5.02	2.92	0.29	2.54	1.59	...	4.56	0.78	0.87	1.09	1.82
1	4.72	3.64	3.60	4.84	4.92	4.08	2.35	1.36	3.68	4.20	...	4.48	0.88	0.96	1.88	2.42
2	5.21	3.70	3.84	4.05	5.06	4.43	4.34	0.49	4.00	2.31	...	3.68	1.13	0.47	0.43	0.87
3	4.39	3.86	1.98	4.73	4.47	3.12	3.51	0.34	4.27	0.62	...	5.06	1.29	3.51	0.64	1.06
4	4.61	4.10	3.55	6.05	4.85	3.55	2.92	0.28	4.05	2.47	...	5.04	1.36	1.32	1.09	1.12

5 rows × 33 columns

4.3.2 Fitting the data in model



```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=100)
```

```
[ ] kmeans.fit(data)
```

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
        n_clusters=100, n_init=10, n_jobs=None, precompute_distances='auto',
        random_state=None, tol=0.0001, verbose=0)
```

4.3.3 Predicting Clusters



```
predicted_clusters = kmeans.predict(data)
predicted_clusters
cluster_labels=[]
j=0
for i in predicted_clusters:
    cluster_labels.append([i,target[j]])
    j+=1
cluster_labels
testdf=pd.DataFrame(cluster_labels)
testdf=testdf.rename(columns={0: 'Class', 1: 'Title'})
testdf=testdf.sort_values('Class', ascending=True)
testdf=testdf.set_index("Class")
#testdf=testdf.groupby("Class")
```

Printing Labels:

```
[ ] centers = kmeans.cluster_centers_
labels = kmeans.labels_
print(labels)
```

```
[85 14 21 41 56 56 56 28 26 31 76 55 84 92 99 55 55 21 21 21 21 94 94 54
99 99 99 59 9 11 54 36 69 23 61 1 93 85 85 94 86 87 23 21 85 47 85 27
4 31 21 85 85 55 59 14 38 21 28 21 4 4 19 86 4 94 88 49 87 59 94 1
94 21 55 21 54 4 56 94 54 56 88 59 28 85 14 28 76 31 76 28 76 76 31 56
87 76 10 10 0 76 76 31 31 94 8 17 27 26 17 17 26 17 26 26 26 26 17 26
17 26 36 30 72 17 56 85 38 41 28 31 81 81 17 73 87 17 68 68 72 30 30 33
23 60 60 59 59 13 33 13 26 23 23 92 59 92 33 27 2 2 60 13 13 2 2 60
2 60 33 59 55 13 13 13 55 60 2 88 80 80 17 80 33 33 59 33 33 15 45 61
33 33 33 61 55 33 55 55 55 33 33 33 13 15 80 72 71 71 48 84 34 84 84 73
34 84 14 14 37 86 37 84 81 81 81 61 60 86 48 14 14 14 23 30 66 66 56 83
47 47 54 35 79 75 43 43 43 64 79 30 84 1 52 3 30 61 45 15 9 75 14 86
86 96 14 72 47 47 47 47 9 83 47 47 9 9 9 9 11 9 94 50 4 94 42 50
50 50 54 8 37 95 23 48 48 48 48 34 48 81 43 43 66 48 79 11 79 5 5 11
54 11 62 11 62 46 62 62 79 62 37 11 37 69 44 44 37 37 27 37 27 37 37 37
83 37 44 37 64 64 70 4 10 40 48 11 36 44 41 78 41 41 33 91 51 41 68 51
68 39 46 40 36 38 40 21 69 44 39 39 46 41 39 39 46 38 89 89 56 89 26 38
39 89 40 77 4 40 41 40 40 20 20 67 20 18 35 20 20 20 35 67 67 67 35 67
67 20 35 35 20 20 20 67 67 20 20 20 35 12 20 35 9 35 12 47 47 47 12 83
20 20 35 35 35 35 5 20 35 35 20 35 35 12 52 52 52 52 52 12 12 12 12 12
12 49 58 87 83 12 12 87 87 35 50 69 18 69 20 12 12 12 92 92 20 35 67 44
44 44 25 83 44 32 32 87 87 32 52 50 24 87 32 9 32 42 42 90 90 90 90 90
90 90 74 42 42 32 42 42 89 49 32 42 42 42 32 4 4 74 25 32 74 4 1 1]
```

4.3.4 Saving The Clustered Data:

```
testdf.to_csv("KnowledgeCleansed_Clusters.csv")
```

4.3.5 Predicting The Class Label (or Cluster)

```
test-data[:1]
test
```

	Administration and Management	Clerical	Economics and Accounting	Sales and Marketing	Customer and Personal Service	Personnel and Human Resources	Production and Processing	Food Production	Computers and Electronics	Engineering and Technology	Design
0	6.23	3.5	4.36	3.9	5.55	5.02	2.02	0.29	2.54	1.50	2.04

```
[ ] kmeans.predict(test)

array([85], dtype=int32)
```

(The Sample Data belongs to cluster no. 85)

4.4 Loading K-means

4.4.1 Data-Prepossessing

Importing Libraries

```
[ ] import pickle
    from sklearn.cluster import KMeans
    import pandas as pd
```

Loading Data-set and Processing

```
loaded_model = pickle.load(open('kmeans_knowledge_cluster.sav', 'rb'))
testdf=pd.read_csv("KnowledgeCleansed_Clusters.csv")
testdf=testdf.set_index("Class")
skills=pd.read_csv("knowledgecleansed.csv")
skills.head()
```

Education and Training	English Language	Foreign Language	Fine Arts	History and Archeology	Philosophy and Theology	Public Safety and Security	Law and Government	Telecommunications	Communications and Media	Transport
4.45	4.56	0.78	0.87	1.09	1.02	3.61	4.00	1.59	3.35	
5.19	4.48	0.88	0.96	1.88	2.42	3.38	3.08	0.81	2.92	
3.87	3.68	1.13	0.47	0.43	0.87	3.12	3.12	2.33	2.68	

```
[ ] target=skills["Title"]
data=skills.drop("Title",axis=1)
feature_name=data.columns
test=data[:1]
test
```

	Administration and Management	Clerical	Economics and Accounting	Sales and Marketing	Customer and Personal Service	Personnel and Human Resources	Production and Processing	Food Production	Computers and Electronics	Engineering and Technology	Design	Building Construct
0	6.23	3.5	4.36	3.9	5.55	5.02	2.92	0.29	2.54	1.59	2.04	

4.4.2 Sample-Prediction

```
[ ] result = loaded_model.predict(test)
print("The test data belongs to Class: ", result[0])
df=testdf.loc[testdf.index==result[0]]
print("The jobs are:",df.values)
```

```
The test data belongs to Class: 58
The jobs are: [['Compensation, Benefits, and Job Analysis Specialists']
['Labor Relations Specialists']
['Compensation and Benefits Managers']
['Human Resources Specialists']
['Medical and Health Services Managers']
['Municipal Clerks']
['Chief Executives']
['Lawyers']
['Spa Managers']]
```

4.5 Career Classify Helper

4.5.1 Objective

Identify candidate's chances for a specific career (job title) based on their education level and experience level, and then recommend them the most suitable title accordingly.

Input:

- The array or groups of clustered titles for the user's skills
- input : Education Level and Experience Level of user

- Output : Scores and shares (in descending order) the career titles that are high likely and map to users' skill, education level and experience level.


4.5.2 The Data-set

The following properties of each career title are measured and included within the CSV:

- Title: Job Title
- Education: Shares whether it is High School Diploma or Bachelors or Masters Degree (in string form)
- Education Level: Number mapping to Education Level
- Data Value Education: Weight for Education Level for that specific title
- Work Experience Level: Number mapping for Experience Level
- Work Experience: Experience (in words) whether it is in months or years
- Data Value Experience: Weight for Education Level for that specific title

Score: Calculated variable based on sum of (Education Level Wt for Ed Level) + (Work Experience Level Wt for Experience) = Scores are for each row for that title

4.5.3 Importing Necessary Libraries and Loading Experience



```
import numpy as np
import pandas as pd
import numbers
import pickle
import sklearn
from sklearn.cluster import KMeans
#print(sklearn.__version__)
```

4.5. CAREER CLASSIFY HELPER

```
[ ]
```

```
Ed_Exp = pd.read_csv('Education_Experience.csv')  
Ed_Exp.head()
```

	Title	Education	Education Level	Data Value_Education	Work Experience Level	Work Experience	Value
0	Chief Executives	Less than High School Diploma	1	0.0	1		0
1	Chief Executives	Less than High School Diploma	1	0.0	2	>0 and upto 1 month	
2	Chief Executives	Less than High School Diploma	1	0.0	3	>1 month and upto and including 3 month	
3	Chief Executives	Less than High School Diploma	1	0.0	4	>3 month and upto and including 6 month	
4	Chief Executives	Less than High School Diploma	1	0.0	5	>6 month and upto and including 1 year	

4.5.4 Input preprocessing (Sample)

Test Data for Now

```
input_cluster_list1 = [  
    [9, "Statisticians"],  
    [9, "Database Architects"],  
    [9, "Software Quality Assurance Engineers and Testers"],  
    [9, "Computer User Support Specialists"],  
    [9, "Mathematical Technicians"],  
    [9, "Computer Systems Analysts"],  
    [9, "Web Developers"],  
    [9, "Software Developers, Applications"],  
    [9, "Computer Programmers"],  
    [9, "Electronic Drafters"]  
]  
  
input_cluster_list2 = [  
    [77, "Information Security Analysts"],  
    [77, "Telecommunications Engineering Specialists"],  
    [77, "Computer Network Architects"],  
    [77, "Computer Systems Engineers/Architects"],  
    [77, "Database Administrators"],  
    [77, "Software Developers, Systems Software"],  
    [77, "Computer Network Support Specialists"],  
    [77, "Network and Computer Systems Administrators"]  
]
```

Load the K Means model and predict to get the highly likely cluster group based on skills entered by user:

```
loaded_model = pickle.load(open('kmeans_knowledge_cluster.sav', 'rb'))
```

```
[ ] test_data = pd.read_csv("test_data.csv")
test_data.head()
```

	Administration and Management	Clerical	Economics and Accounting	Sales and Marketing	Customer and Personal Service	Personnel and Human Resources	Production and Processing	Food Production	Computers and Electronics	Engineering and Technology	...	Foreign Language	Fi	Ar
0	3	5	3	4	5	5	4	2	3	2	..	1		

1 rows × 34 columns

```
expected_target=test_data["Title"]
input_data=test_data.drop("Title",axis=1)
feature_name=test_data.columns

print(expected_target)
print(input_data)
print(feature_name)
```

```
0    Management Analyst
Name: Title, dtype: object
Administration and Management    Clerical    Economics and Accounting \
0                               3         5                               3

Sales and Marketing    Customer and Personal Service \
0                    4                               5

Personnel and Human Resources    Production and Processing \
0                             5                               4

Food Production    Computers and Electronics    Engineering and Technology \
0                2                3                2

...    English Language    Foreign Language    Fine Arts \
0    ...                5                1                0

History and Archeology    Philosophy and Theology \
0                    2                2

Public Safety and Security    Law and Government    Telecommunications \
0                3                3                4

. . . . .
```

```
[1 rows x 33 columns]
Index(['Administration and Management', ' Clerical',
      ' Economics and Accounting', ' Sales and Marketing',
      ' Customer and Personal Service', ' Personnel and Human Resources',
      ' Production and Processing', ' Food Production',
      ' Computers and Electronics', ' Engineering and Technology', ' Design',
      ' Building and Construction', ' Mechanical', ' Mathematics', 'Physics',
      ' Chemistry', ' Biology', ' Psychology', ' Sociology and Anthropology',
      ' Geography', ' Medicine and Dentistry', ' Therapy and Counseling',
      ' Education and Training', ' English Language', ' Foreign Language',
      ' Fine Arts', ' History and Archeology', ' Philosophy and Theology',
      ' Public Safety and Security', ' Law and Government',
      ' Telecommunications', ' Communications and Media', ' Transportation',
      'Title'],
      dtype='object')
```



```
test_data=test_data.drop("Title",axis=1)
test_data.head()
```

	Administration and Management	Clerical	Economics and Accounting	Sales and Marketing	Customer and Personal Service	Personnel and Human Resources	Production and Processing	Food Production	Computers and Electronics	Engineering and Technology	...	Lar
0	3	5	3	4	5	5	4	2	3	2

1 rows x 33 columns

Read cluster grouping done by KMeans Cluster model:

```
[ ] read_cluster_grouping = pd.read_csv("KnowledgeCleansed_Clusters.csv")
    cluster_group_df = read_cluster_grouping.set_index("Class")

    cluster_group_df.head()
```


Class	Title
0	Tank Car, Truck, and Ship Loaders
0	Model Makers, Wood
0	Cabinetmakers and Bench Carpenters
0	Layout Workers, Metal and Plastic
0	Welders, Cutters, and Welder Fitters

```
▶ print(cluster_group_df.index)
```

```
Int64Index([ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
             ...
            99, 99, 99, 99, 99, 99, 99, 99, 99, 99],
            dtype='int64', name='Class', length=966)
```

4.5.5 Predicting the class/Cluster:

```
[ ] result = loaded_model.predict(test_data)
    print("The test data belongs to Class: ", result[0])
```

```
The test data belongs to Class: 95
```

Jobs Related to that class/cluster:

```
[ ] selected_title_group = cluster_group_df.loc[cluster_group_df.index==result[0]]
    print("The jobs are:", selected_title_group.values)

    selected_title_group.head()
```

```
The jobs are: [['Online Merchants']  
['Business Intelligence Analysts']  
['Real Estate Brokers']  
['Marketing Managers']  
['Sales Managers']  
['Management Analysts']  
['Distance Learning Coordinators']  
['First-Line Supervisors of Non-Retail Sales Workers']]
```

Title	
Class	
95	Online Merchants
95	Business Intelligence Analysts
95	Real Estate Brokers
95	Marketing Managers
95	Sales Managers

Getting all the clustered titles:

```
▶ title_list = selected_title_group["Title"]  
  
print(title_list)  
  
print("*****List of Titles parsed out*****")  
for title in title_list:  
    print(title)  
print("*****End of List of Titles parsed out*****")
```

```

Class
95                                     Online Merchants
95                               Business Intelligence Analysts
95                               Real Estate Brokers
95                               Marketing Managers
95                               Sales Managers
95                               Management Analysts
95                               Distance Learning Coordinators
95 First-Line Supervisors of Non-Retail Sales Wor...
Name: Title, dtype: object
*****List of Titles parsed out*****
Online Merchants
Business Intelligence Analysts
Real Estate Brokers
Marketing Managers
Sales Managers
Management Analysts
Distance Learning Coordinators
First-Line Supervisors of Non-Retail Sales Workers
*****End of List of Titles parsed out*****

```

4.5.6 Data Preprocessing

Reading the scoring data set that is cleansed:

```

[ ]
    Ed_Exp = pd.read_csv('Education_Experience.csv')
    Ed_Exp.head()

```

```

[ ] Ed_Exp_filtered = pd.DataFrame(Ed_Exp[Ed_Exp["Title"].isin(title_list)])

    Ed_Exp_filtered.head(50)

    Ed_Exp_filtered['Score'] = ((Ed_Exp_filtered['Education Level'] * Ed_Exp_filtered['Data Value_Education']) +
                                (Ed_Exp_filtered['Work Experience Level'] * Ed_Exp_filtered['Data Value_Experience']))

    Ed_Exp_filtered.head()

```

	Title	Education	Education Level	Data Value_Education	Work Experience Level	Work Experience	Data Value_Experience	Score
528	Marketing Managers	Less than High School Diploma	1	0.0	1	0	0.00	0.00
529	Marketing Managers	Less than High School Diploma	1	0.0	2	>0 and upto 1 month	1.43	2.86
530	Marketing Managers	Less than High School Diploma	1	0.0	3	>1 month and upto and including 3 month	0.00	0.00
531	Marketing Managers	Less than High School Diploma	1	0.0	4	>3 month and upto and including 6 month	0.00	0.00
532	Marketing Managers	Less than High School Diploma	1	0.0	5	>6 month and upto and including 1 year	9.04	45.20

4.5.7 Get user's input on Education Level

```
while True:
    try:
        print("*****Education Level*****")
        print ("1 - Less than High School Diploma")
        print ("2 - High School Diploma")
        print ("3 - Post Secondary Certificate")
        print ("4 - Some College Course")
        print ("5 - Associate's Degree")
        print ("6 - Bachelors's Degree")
        print ("7 - Post-Baccalaureate Certificate")
        print ("8 - Master's Degree")
        print ("9 - Post-Master's Certificate")
        print ("10 - First Professional Degree")
        print ("11 - Doctoral Degree")
        print ("12 - Post-Doctoral Training")
        print("*****End of Education Levels*****")
        ed_level = int(input("Please enter suitable Education Level (1-12):"))
    except ValueError:
        print("Sorry, I didn't understand that.")
        continue
    else:
        break

print(f"Education Level: {ed_level}")
```

```
*****Education Level*****
1 - Less than High School Diploma
2 - High School Diploma
3 - Post Secondary Certificate
4 - Some College Course
5 - Associate's Degree
6 - Bachelors's Degree
7 - Post-Baccalaureate Certificate
8 - Master's Degree
9 - Post-Master's Certificate
10 - First Professional Degree
11 - Doctoral Degree
12 - Post-Doctoral Training
*****End of Education Levels*****
Please enter suitable Education Level (1-12):8
Education Level: 8
```

4.5.8 Get Experience Level



```
while True:
    try:
        print("*****Experience Level*****")
        print ("1 - No experience")
        print ("2 - Upto 1 month experience")
        print ("3 - (1-3 months) experience")
        print ("4 - (3-6 months) experience")
        print ("5 - (6 months - 1 year) experience")
        print ("6 - (1-2 years) experience")
        print ("7 - (2-4 years) experience")
        print ("8 - (4-6 years) experience")
        print ("9 - (6-8 years) experience")
        print ("10 - (8-10 years) experience")
        print ("11 - (> 10 years) experience")
        print("*****End of Experience Level*****")
        exp_level = int(input("Please enter suitable Experience Level (1-11):"))
    except ValueError:
        print("Sorry, I didn't understand that.")

        continue
    else:
        break

print(f"Experience Level: {exp_level}")
```

```
*****Experience Level*****
1 - No experience
2 - Upto 1 month experience
3 - (1-3 months) experience
4 - (3-6 months) experience
5 - (6 months - 1 year) experience
6 - (1-2 years) experience
7 - (2-4 years) experience
8 - (4-6 years) experience
9 - (6-8 years) experience
10 - (8-10 years) experience
11 - (> 10 years) experience
*****End of Experience Level*****
Please enter suitable Experience Level (1-11):11
Experience Level: 11
```

4.5.9 Get the current title:

```
[ ]
current_title = str(input("Please enter current title:"))

print(f"Current Title: {current_title}")
```

```
Please enter current title:Business Analyst
Current Title: Business Analyst
```

Filter the user list based on user experience and education level under the clustered group of titles and Shows results for the user scores:

```
print(f"Education Level: {ed_level}")
print(f"Experience Level: {exp_level}")
print(f"Current Title: {current_title}")

user_ed_list = [ed_level]
user_exp_list = [exp_level]

Ed_Exp_User_filtered = pd.DataFrame(Ed_Exp_filtered[(Ed_Exp_filtered["Education Level"].isin(user_ed_list)) &
(Ed_Exp_filtered["Work Experience Level"].isin(user_exp_list))])
|
Ed_Exp_User_filtered.head(50)
```

```
Education Level: 8
Experience Level: 11
Current Title: Business Analyst
```

	Title	Education	Education Level	Data Value_Education	Work Experience Level	Work Experience	Data Value_Experience	Score
615	Marketing Managers	Master's Degree	8	24.36	11	>10 year	2.47	222.05
747	Sales Managers	Master's Degree	8	8.70	11	>10 year	0.00	69.60
4179	Distance Learning Coordinators	Master's Degree	8	63.64	11	>10 year	0.00	509.12
10119	Management Analysts	Master's Degree	8	46.15	11	>10 year	11.54	496.14
11571	Online Merchants	Master's Degree	8	0.00	11	>10 year	5.49	60.39
16587	Business Intelligence Analysts	Master's Degree	8	33.33	11	>10 year	8.33	358.27
76911	First-Line Supervisors of Non-Retail Sales Wor..	Master's Degree	8	7.04	11	>10 year	6.10	123.42
79287	Real Estate Brokers	Master's Degree	8	0.00	11	>10 year	1.59	17.49

```
[ ] Curr_Title_user_filtered = pd.DataFrame()
Title_Empty = False

if (not current_title):
    Title_Empty = True
    curr_user_title_list = [current_title]
    Curr_Title_user_filtered = Ed_Exp_filtered[(Ed_Exp_filtered["Title"].isin(curr_user_title_list))]

Curr_Title_user_filtered.head()
```

4.5.10 Alternate titles

Reading the Data-set:

```
▶ Alternate_Titles = pd.read_csv('AlternateTitles.csv')
  Alternate_Titles.head()

  AltTitles_pd = Alternate_Titles[['Title', 'Alternate Title', 'Short Title']]
  AltTitles_pd.head() |
```

	Title	Alternate Title	Short Title
0	Chief Executives	Aeronautics Commission Director	NaN
1	Chief Executives	Agricultural Services Director	NaN
2	Chief Executives	Alcohol and Drug Abuse Assistance Program Admi...	NaN
3	Chief Executives	Arts and Humanities Council Director	NaN
4	Chief Executives	Bakery Manager	NaN

To map the current title with alternating titles:

```
[ ] curr_user_title_list = [current_title]

  mapping_title = pd.DataFrame()

  if ((Curr_Title_user_filtered.empty == True) and (Title_Empty == False)):
      mapping_title = AltTitles_pd[(AltTitles_pd["Alternate Title"].isin(curr_user_title_list))]

  mapping_title = mapping_title.reset_index(drop=True)
  mapping_title.head()
```

	Title	Alternate Title	Short Title
0	Management Analysts	Business Analyst	NaN
1	Accountants	Business Analyst	NaN
2	Computer Systems Analysts	Business Analyst	NaN
3	Geographic Information Systems Technicians	Business Analyst	NaN
4	Business Intelligence Analysts	Business Analyst	NaN

Returning the alternate titles:

```
▶ Alternate_titles = mapping_title["Title"]
  print(Alternate_titles)
```

4.5. CAREER CLASSIFY HELPER

```
0           Management Analysts
1           Accountants
2           Computer Systems Analysts
3   Geographic Information Systems Technicians
4           Business Intelligence Analysts
5           Operations Research Analysts
Name: Title, dtype: object
```

4.5.11 Make Predictions Based on probability of users education and experience:

Printing results based on classification possibilities:

```
Ed_Exp_User_Ordered = Ed_Exp_User_filtered.sort_values(by='Score', ascending=False)
Ed_Exp_User_Ordered.head(20)
```

	Title	Education	Education Level	Data Value_Education	Work Experience Level	Work Experience	Data Value_Experience	Score
4179	Distance Learning Coordinators	Master's Degree	8	63.64	11	>10 year	0.00	509.12
10119	Management Analysts	Master's Degree	8	46.15	11	>10 year	11.54	496.14
16587	Business Intelligence Analysts	Master's Degree	8	33.33	11	>10 year	8.33	358.27
615	Marketing Managers	Master's Degree	8	24.36	11	>10 year	2.47	222.05
76911	First-Line Supervisors of Non-Retail Sales Wor...	Master's Degree	8	7.04	11	>10 year	6.10	123.42
747	Sales Managers	Master's Degree	8	8.70	11	>10 year	0.00	69.60
11571	Online Merchants	Master's Degree	8	0.00	11	>10 year	5.49	60.39
79287	Real Estate Brokers	Master's Degree	8	0.00	11	>10 year	1.59	17.49

```
for index, row in Ed_Exp_User_Ordered.iterrows():
    print (row['Title'], row['Education'], row['Work Experience'])
```

```
Marketing Managers Master's Degree >8 year and upto and including 10 year
Distance Learning Coordinators Master's Degree >8 year and upto and including 10 year
Management Analysts Master's Degree >8 year and upto and including 10 year
Business Intelligence Analysts Master's Degree >8 year and upto and including 10 year
Sales Managers Master's Degree >8 year and upto and including 10 year
First-Line Supervisors of Non-Retail Sales Workers Master's Degree >8 year and upto and including 10 year
Online Merchants Master's Degree >8 year and upto and including 10 year
Real Estate Brokers Master's Degree >8 year and upto and including 10 year
```


4.5.12 Job Descriptions:

Reading the occupation dataset:

```

Additional_JobTitle_Details = pd.read_csv('occupation.csv')
Additional_JobTitle_Details.head()

JobDetails_pd = Additional_JobTitle_Details[['Title', 'Description', 'Technology', 'CoreTasks']]

JobDetails_pd.head()

```

	Title	Description	Technology	CoreTasks
0	Chief Executives	Determine and formulate policies and provide o...	[Adobe Systems Adobe Acrobat', 'Blackbaud The...	['Direct or coordinate an organization's finan...
1	Chief Sustainability Officers	Communicate and coordinate with management, sh...	['Microsoft Access', 'Microsoft Dynamics GP', ...	['Develop or execute strategies to address iss...
2	General and Operations Managers	Plan, direct, or coordinate the operations of ...	['Adobe Systems Adobe Acrobat', 'Adobe Systems...	['Review financial statements, sales or activi...
3	Legislators	Develop, introduce or enact laws and statutes ...	['Adobe Systems Adobe Acrobat', 'IBM Domino', ...	['
4	Advertising and Promotions Managers	Plan, direct, or coordinate advertising polici...	['Adobe Systems Adobe Acrobat', 'Adobe Systems...	['Inspect layouts and advertising copy and edi...

Merge the data with additional column:

```

final_scored_title_list = pd.merge(Ed_Exp_User_Ordered, JobDetails_pd, on = 'Title')

final_scored_title_list.head(20)

```

	Title	Education	Education Level	Data Value_Education	Work Experience Level	Work Experience	Data Value_Experience	Score	Description	Technology	CoreTasks
0	Distance Learning Coordinators	Master's Degree	8	63.64	11	>10 year	0.00	509.12	Coordinate day-to-day operations of distance l...	['Adobe Systems Adobe Acrobat', 'Adobe Systems...	['Communicate to faculty, students, or other u...
1	Management Analysts	Master's Degree	8	46.15	11	>10 year	11.54	496.14	Conduct organizational studies and evaluations...	['Adobe Systems Adobe Acrobat', 'Adobe Systems...	['Document findings of study and prepare recom...
2	Business Intelligence Analysts	Master's Degree	8	33.33	11	>10 year	8.33	358.27	Produce financial and market intelligence by q...	['Adobe Systems Adobe Acrobat', 'Adobe Systems...	['Analyze competitive market strategies throug...
3	Marketing Managers	Master's Degree	8	24.36	11	>10 year	2.47	222.05	Plan, direct, or coordinate marketing policies...	['Adobe Systems Adobe Acrobat', 'Adobe Systems...	['Identify, develop, or evaluate marketing str...
4	First-Line Supervisors of Non-Retail Sales Wor...	Master's Degree	8	7.04	11	>10 year	6.10	123.42	Directly supervise and coordinate activities o...	['Delphi Technology', 'Microsoft Access', 'Mic...	['Confer with company officials to develop met...

4.6 Skill Processing

4.6.1 Reading the skills

Convert skills into machine learning training data-set

```
import pandas as pd
skills=pd.read_csv("Knowledge.csv")
skills.head()
```

	Title	Element Name	Level
0	Chief Executives	Administration and Management	6.23
1	Chief Executives	Clerical	3.50
2	Chief Executives	Economics and Accounting	4.36
3	Chief Executives	Sales and Marketing	3.90
4	Chief Executives	Customer and Personal Service	5.55

Get the unique Skills to convert into column names and append Title:

```
columns=skills["Element Name"].unique().tolist()
columns.append("Title")
columns
```

```
['Administration and Management',
 'Clerical',
 'Economics and Accounting',
 'Sales and Marketing',
 'Customer and Personal Service',
 'Personnel and Human Resources',
 'Production and Processing',
 'Food Production',
 'Computers and Electronics',
 'Engineering and Technology',
 'Design',
 'Building and Construction',
 'Mechanical',
 'Mathematics',
 'Physics',
```

Get the unique titles to convert into rows:

```
[3] row=skills["Title"].unique()
```

4.6.2 Creating new File for sorted skills for each title

Store the result as a list of lists → For each title find the Skills Required → Store all the skills for the specific title as a list → Append the title to the new row and then append to result:

```
results=[]
for title in row:
    new_row=[]
    for sk in columns:
        if sk!="Title":
            try:
                val=skills.loc[(skills["Title"]==title) & (skills["Element Name"]==sk),"Importance"].tolist()
                new_row.append(val[0])
            except:
                new_row.append(0)
    new_row.append(title)
    results.append(new_row)
```

Converting into CSV file:

```
DF=pd.DataFrame(results,columns=columns)
DF.to_csv("ImportanceCleansed.csv",index=False)

DF=pd.read_csv("ImportanceCleansed.csv")
DF.head()
```

	Administration and Management	Clerical	Economics and Accounting	Sales and Marketing	Customer and Personal Service	Personnel and Human Resources	Production and Processing	Food Production	Computers and Electronics	Engineering and Technology	...	Foreign Language	Fine Arts	History and Archeology	Philosophy and Theology	Public Safety and Security
0	4.75	2.66	3.70	3.23	4.09	4.10	2.63	1.14	2.23	1.75	...	1.56	1.43	1.48	1.70	3.30
1	3.85	2.58	2.96	3.50	3.62	2.72	2.23	1.64	2.65	3.35	...	1.40	1.38	1.80	1.85	2.40
2	4.35	3.51	3.47	3.47	3.95	3.76	3.39	1.34	3.33	2.42	...	1.62	1.16	1.21	1.51	3.10
3	4.11	3.10	2.21	3.88	3.79	2.40	3.12	1.20	3.43	1.33	...	1.68	2.90	1.34	1.47	1.78
4	4.04	3.01	3.10	4.85	3.85	2.71	2.46	1.12	3.51	2.77	...	1.64	1.70	1.68	1.64	2.50

4.6.3 The Random Forest Classifier

Sample data:

```
▶ target=DF["Title"]
data=DF.drop("Title",axis=1)
feature_name=data.columns

[8] target.head()

0          Chief Executives
1  Chief Sustainability Officers
2  General and Operations Managers
3  Advertising and Promotions Managers
4          Marketing Managers
Name: Title, dtype: object
```

Splitting the data-set:

```
[11] from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(data, target, random_state=42)
```

Importing the Classifier and fitting the data-set:

```
▶ from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators=200)
rf = rf.fit(data, target)
feature_names=data.columns
target_names=target
print(X_test)
```

	Reading Comprehension	Active Listening	Writing	Speaking	Mathematics	\
244	4.12	4.00	4.00	3.88	2.50	
467	3.62	3.75	3.25	4.00	2.88	
836	3.00	3.00	2.62	3.00	2.00	
557	3.00	3.50	2.50	3.50	2.75	
70	3.88	4.00	3.50	4.00	2.25	
314	4.38	4.12	4.00	4.50	2.00	
921	3.00	3.75	2.62	4.00	2.12	
787	3.12	3.12	2.62	3.00	2.38	
88	3.75	3.88	3.75	3.62	3.88	
665	3.62	3.62	3.25	3.62	2.38	
76	4.12	4.12	4.00	4.00	3.00	
425	4.12	3.88	3.75	4.00	2.38	
355	3.75	4.00	3.50	3.88	2.25	
656	3.25	3.88	3.12	4.00	2.00	
529	3.12	3.00	2.75	3.12	2.75	
842	2.75	3.12	2.50	2.88	2.75	
30	4.12	4.12	4.00	4.12	2.88	
67	3.75	3.88	3.75	4.00	2.50	
578	3.25	4.00	3.12	4.12	1.62	
359	3.00	3.50	2.88	3.25	2.00	

(and many more lines)

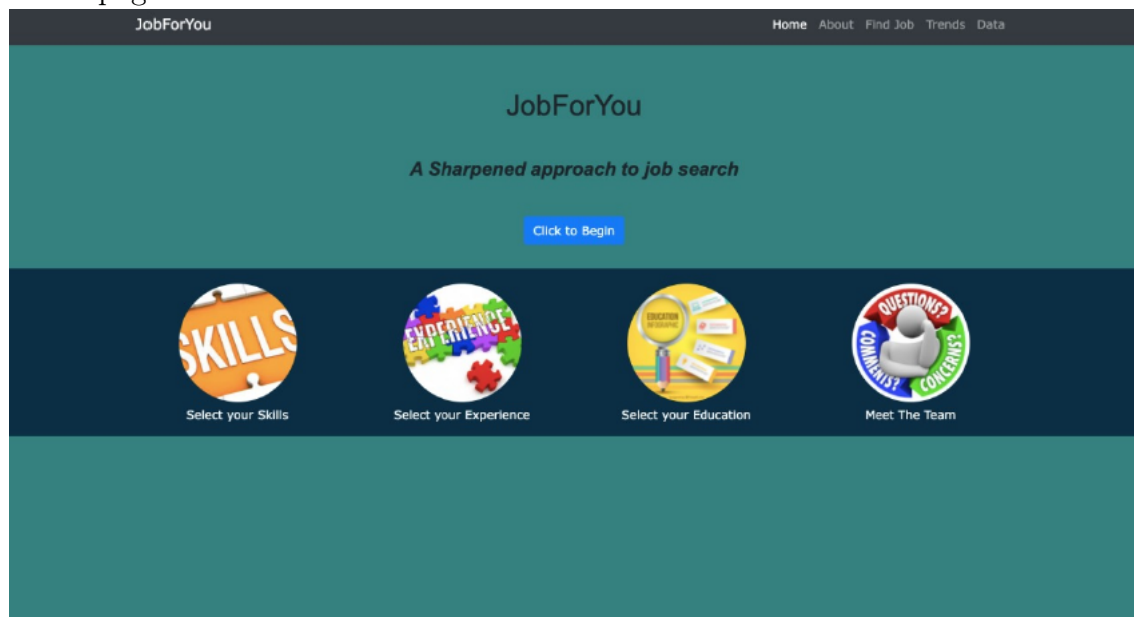
4.7 Front-End Development

4.7.1 Tech Stack

- Languages: HTML, CSS, JavaScript
- IDE: Visual Studio Code
- JQuery AJAX(Pass User Input From Client side To Server side and get response from server)

4.7.2 Home Page

This is the first page that appears when the site opens. This page contains links to all the other pages.



4.7.3 Find Job Page

This is the page that achieves the main objective of this project. It takes all the inputs of the user's skills, educational experience, Work Experience, etc.. Based on these details, it finds the suitable jobs for the user. For taking the inputs of educational and work experience, the user is asked to rate himself from 0 to 10, each number representing a particular experience category (example, fresher, 2 year experience, graduate, etc.) For

4.7. FRONT-END DEVELOPMENT

each skill, the user is asked to check a circle from a number of circles, representing his level of that skill (like newbie, advanced ,etc.)

JobForYou

HomeAboutFind JobTrendsData

COMPLETE THE QUESTIONNAIRE TO FIND YOUR PERFECT JOB TITLES

Name (*)

Last Name (*)

Email Address (*)

a@jobfitt.com

Phone Number (*)

7771111111

Complete Step1: Skills Assessment

(NOTE: Default of Beginner Level is selected for all skill areas)

Complete Step 2: Career and Education Level

SUBMIT FORM

JobForYou

HomeAboutFind JobTrendsData

Complete Step1: Skills Assessment

(NOTE: Default of Beginner Level is selected for all skill areas)

Skills	Beginner	Basic	Skilled	Advanced	Expert
Administration and Management	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Clerical	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Economics and Accounting	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sales and Marketing	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Customer and Personal Service	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Personnel and Human Resources	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Production and Processing	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Food Production	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Computers and Electronics	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Engineering and Technology	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

4.7.4 Trends Page

Contains Information about the latest and past years job growths



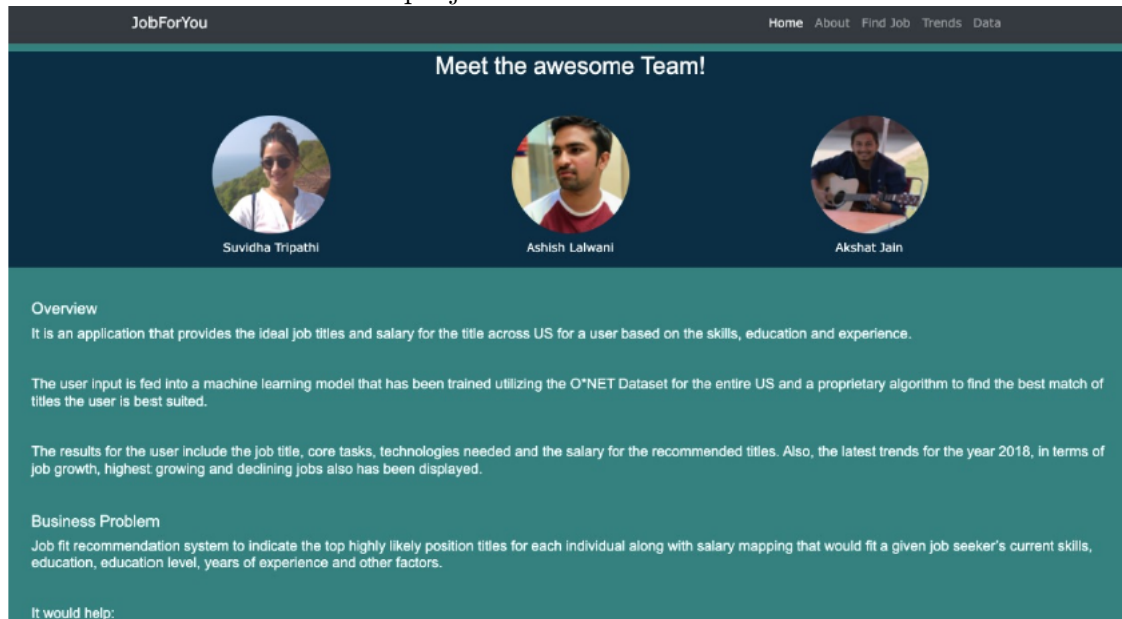
4.7.5 Data Page

Contains data about what jobs are available, how many openings are there, etc

JobForYou					Home About Find Job Trends Data		
Title	Employment 2016 (thousands)	Employment 2026 (thousands)	Occupational openings, 2016-2026 annual average (thousands)	2017 median annual wage	Typical entry-level education	Work experience in a related occupation	Typical on-the-job training
					All	All	All
First Next Previous Last							

4.7.6 About Page

Contains information about the project and the team.



Chapter 5

Result And Discussion

5.1 Predicted Jobs of predicted cluster:

```
[ ] result = loaded_model.predict(test)
print("The test data belongs to Class: ", result[0])
df=testdf.loc[testdf.index==result[0]]
print("The jobs are:",df.values)
```

```
The test data belongs to Class: 58
The jobs are: [['Compensation, Benefits, and Job Analysis Specialists']
['Labor Relations Specialists']
['Compensation and Benefits Managers']
['Human Resources Specialists']
['Medical and Health Services Managers']
['Municipal Clerks']
['Chief Executives']
['Lawyers']
['Spa Managers']]
```

```
[ ] selected_title_group =cluster_group_df.loc[cluster_group_df.index==result[0]]
print("The jobs are:",selected_title_group.values)

selected_title_group.head()
```

5.2. RESULT BY THE RANDOM FOREST CLASSIFIER:

```
The jobs are: [['Online Merchants']  
['Business Intelligence Analysts']  
['Real Estate Brokers']  
['Marketing Managers']  
['Sales Managers']  
['Management Analysts']  
['Distance Learning Coordinators']  
['First-Line Supervisors of Non-Retail Sales Workers']]
```

Title	
Class	
95	Online Merchants
95	Business Intelligence Analysts
95	Real Estate Brokers
95	Marketing Managers
95	Sales Managers

5.2 Result by the Random Forest Classifier:

```
test=DF[:1]  
test_target=test["Title"]  
test=test.drop("Title",axis=1)  
test["Active Listening"]=0  
test["Mathematics"]=0  
test["Writing"]=3.25  
test["Reading Comprehension"]=3.62  
test["Critical Thinking"]=0  
test["Science"]=0  
rf.predict(test)
```

```
array(['Chief Executives'], dtype=object)
```

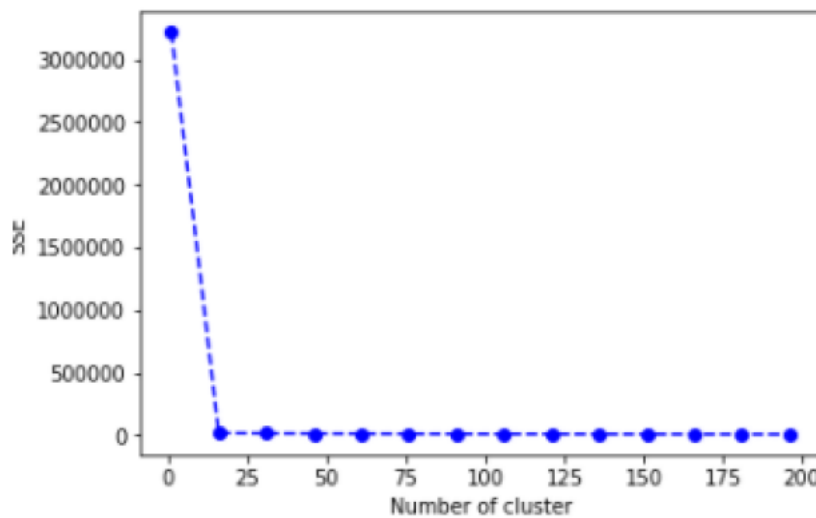
5.3 Evaluation metrics

5.3.1 Elbow Method

The Elbow method is an algorithm to find the optimal number of clusters in the data. In this method, we plot the variation in data corresponding to the number of clusters,

and the number of clusters where the elbow is formed, is used as ‘k’. This method can also be used to select the number of parameters in other models, for example, how many principal components should be used to describe a dataset. The elbow point of the curve (also called knee point) is a common formulation that is used to achieve optimization, so as to control the increase in parameters, to prevent loss. In our case, this implies that we have to select a certain number of clusters, after which increasing the number of clusters would not improve the model’s performance. There is a standard understanding that when the more clusters are added, the performance will be improved, since there are more parameters (more clusters) to use, but after a certain number of clusters, the model starts to overfit, and that is explained by the elbow method. The idea is that the original clusters will add more information (specify more variations), as the data actually contains those multiple groups (so these clusters are required), but if the number of clusters exceeds the actual number of groups in the data, the additional information will go down a lot, because it just separates real groups. Assuming this is the case, there will be a sharp elbow in the graph of variations against clusters: rapid increase to k, and then increase slightly after k.

Following is the elbow curve for our implemented model:



So as you can see in the above graph, an elbow is formed near number of clusters = 25, so that’s our elbow point

5.3.2 Silhouette method:

Silhouette method: In this method, we interpret and validate the consistency within clusters. The algorithm creates a representation of how accurately the points are clustered. Here we find the cohesion and separation, that is, the similarity of the point to its own cluster, in comparison to other clusters. The range of the silhouette coefficient is from -1 to +1. A high Silhouette value indicates that the point is highly matched to its own cluster and very badly to the other clusters, and the clustering is appropriate. Low value indicates a bad clustering (too low, or too few clusters, and poorly matched). Various Distance measures can be used to calculate the Silhouette coefficient, such as the Euclidean distance or the Manhattan distance.

$$\text{Silhouette Score} = \frac{(\text{average inter-cluster distance} - \text{average intra-cluster distance})}{\max(\text{average inter-cluster distance}, \text{average intra-cluster distance})}$$

The Average Silhouette Coefficients for our algorithm:

```
For n_clusters=25, The Silhouette Coefficient is 0.25819784539767043
For n_clusters=50, The Silhouette Coefficient is 0.2849557725377086
For n_clusters=75, The Silhouette Coefficient is 0.30417968181741606
For n_clusters=100, The Silhouette Coefficient is 0.3307248059253102
For n_clusters=125, The Silhouette Coefficient is 0.3500809376536469
For n_clusters=150, The Silhouette Coefficient is 0.3532717073774063
For n_clusters=175, The Silhouette Coefficient is 0.3460653058030282
```

Bibliography