* **Best\_cluster\_number:**
  + **Data\_preprocessing** (Line 26):
    - Droping unnecessary columns from Data for preprocessing(Line 27).
    - Creating function for cleaning language column where removing the unnecessary character and then applying the lower method and again splitting the string on bases of remaining character. Join the splitted string back to string. Later replace the other characters such as ‘+,#, etc..’(Line 28-48).
    - Same thing is followed as written in above point(Line 50-66).
    - Using for loop on data for cleaning purpose (Line 69-81).
  + **find\_best\_cluster function** (Line 85):
    - Creating empty list for cluster score & best cluster numbers by passing cluster range (Line 86-88).
    - Using for loop by using cluster range number on model and training it, then using silhouette\_score to find the score for each cluster number, then looking on how they got that score later sorting it in ascending order. Using if condition to find good cluster number and adding those to empty list. At the end finding the best cluster number (Line 90-102).
  + **Cluster\_model** (Line 107):
    - Importing Mentor Json file and converting it to Pandas DataFrame & Creating a copy of original DataFrame (Line 108-110).
    - Adding a dummy row which includes all the domains and languages for vectorization(OneHotEncoder) (Line 111-112).
    - Calling data\_preprocessing function to clean and then drop few columns (Line 113).
    - Calling CountVectorizer() for domains(i.e, OneHotEncoder) and then creating a separate DataFrame for the values after fit\_transform method (Line 115-118).
    - Calling CountVectorizer() for languages(i.e, OneHotEncoder) and then creating a separate DataFrame for the values after fit\_transform method (Line 119-122).
    - Now Merging both the DataFrame(Domains, languages) after vectorization and then dropping the last row(Dummy Data) for next process (Line 123-124).
    - Using Dimensionality Reduction Method i.e, PCA to reduce number of columns as to increase the model performance (Line 125-126).
    - Calling find\_best\_cluster function to find the best number of clusters(Line 127).
    - Using the best number of clusters for training the model i.e, KMeans and the predicting the output(Line 128-129).
* **Matching\_script:**
  + - Load Model using pickle (Line 18).
  + **Data\_preprocessing** (Line 20):
    - Droping unnecessary columns from Data for preprocessing(Line 21).
    - Creating function for cleaning language column where removing the unnecessary character and then applying the lower method and again splitting the string on bases of remaining character. Join the splitted string back to string. Later replace the other characters such as ‘+,#, etc..’(Line 23-39).
    - Same thing is followed as written in above point(Line 41-50).
    - Using for loop on data for cleaning purpose (Line 53-69).
  + **Perfect\_match\_process** (Line 74):
    - Assign scores for domain, languages & Branch score (Line 75-77).
    - Create a empty list for match score, Mentor matched id & mentee matched id (Line 78-81).
    - Create a copy of mentor Data (Line 84).
    - For loop for each mentee to find matching in mentor and stored in target\_matched\_df (Line 86-87).
    - Consider each mentees data by droping target column (Line 89). Now consider a list of domains of each mentee in temp\_mentee\_domain. Similarly for language as temp\_mentee\_language & for branch ad temp\_mentee\_branch (Line 90-92).
    - For loop for each mentor matched to mentee based on target score. Where initial score for every mentor be zero. Consider temp\_df as each numpy array of each matched mentor data by droping target & experience level. Use another for loop so as to convert numpy array to string as st (Line 94-101).
    - Looking into each items from temp\_mentee\_domain list and if those each item is present in mentor string i.e, st then add domain the score to each\_score variable. Similarly is done with temp\_mentee\_language & temp\_mentee\_branch using for loop and if condition (Line 105-116).
    - Now add each mentor and mentee score in mentor\_Score list and then clear st which was used to store each mentor values (Line 117-120).
    - Using try and except to avoid error or control error if there is no mentor matched or found for mentee. Inside try adding match\_score of each mentor who was matched on bases of cluster number(target) and then filtering it with the maximum match score (Line 121-123).
    - Checking if the dataFrame length is more than one after filtering with maximum score, if yes then sort the dataframe bases on the experience level as descending order and then select the index number of first row from sorted dataframe. Later use the index value to find the mentor id from original dataframe and add it to mentor\_matched\_id. After adding the id drop the mentor from dataframe which is used for matching using index value so no further same mentor should be assigned (Line 125-131).
    - If the dataframe length is exactly one then consider the index number of that mentor and then add the mentor id using index number to mentor\_matched\_id. Later drop that mentor row from the dataframe used for matching process (Line 134-140).
    - Now add mentee id to mentee\_matched\_id (Line 142). And then clear the match score for next mentor mentee matching scores (Line 143).
    - Except condition is used when it there are no match found and through error as no index found and hence prints output as Not Found (Line 144-145).
    - Create a new dataframe as final\_perfect\_matched to have best matched mentor mentee id (Line 146-148).
  + **Mentor\_matching** (Line 155):
    - Importing mentor json file and then converting it to pandas dataframe (Line 156-157).
    - Copying dataframe for the preprocess of duplicate(copy) dataframe (Line 160).
    - Adding dummy values for data preprocessing like vectorization in copied data(Line 164-165).
    - Calling data\_preprocessing function on data (Line 166).
    - Undergoing vectorization on domain and language column and their values are saved in separate dataframe later both the dataframe after vectorization is merged using concat as final\_df where the dummy values are later dropped from final\_df(Line 171-181).
    - Using Dimensionality Reduction method as PCA to increase the performance of model by compressing n columns to 2 columns (Line 182-183).
    - Using pretrained model to predict the cluster groups (Line 184).
    - Droping dummy values from mentor\_df\_pre which was cleaned in data preprocessing and then adding the clusters values as target column (Line 185-186).
    - Converting Mentees Json file to pandas dataframe and then creating a duplicate of that for preprocessing(Line 188-190).
    - Using duplicate mentees dataframe in data\_preproceesing function(Line 197).
    - Transforming the domain and language values as vectorization and storing it in two different dataframe. Later both the dataframes are merged using concat and stored it as final\_df(Line 198-202).
    - Using Dimensionality Reduction menthod as transform on final\_df of mentees (Line 204).
    - Predicting the cluster using pretrained model and those cluster numbers are then stored in predict\_data under target column (Line 205-206).
    - Calling perfect\_match\_process function to find the perfect matching pairs with their id numbers as output in dataframe format (Line 207).
    - Returning the matched id dataframe (Line 211-214).