Beer Challenge Data Analysis Approach

# Technology Stack

Python 3.7, Tensorflow2.3, spacy, keras, numpy, pandas, sklearn, matplotlib, seaborn, datetime

# Problem statement

Do data analysis on beer challenge dataset and extract valuable insights, plot the graphs to make, analyze reviews, validate the ratings, etc.

# Stat Problems

## Problems

1. Rank top 3 Breweries which produce the strongest beers?
2. Which year did beers enjoy the highest ratings?
3. If you were to recommend 3 beers to your friends based on this data which ones will you recommend?
4. Which Beer style seems to be the favorite based on reviews written by users?

## Input

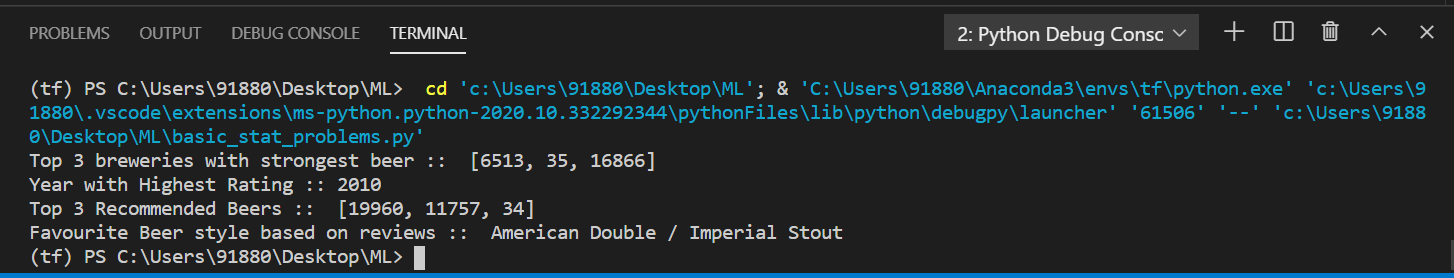
Beer challenge CSV file

## Algorithm

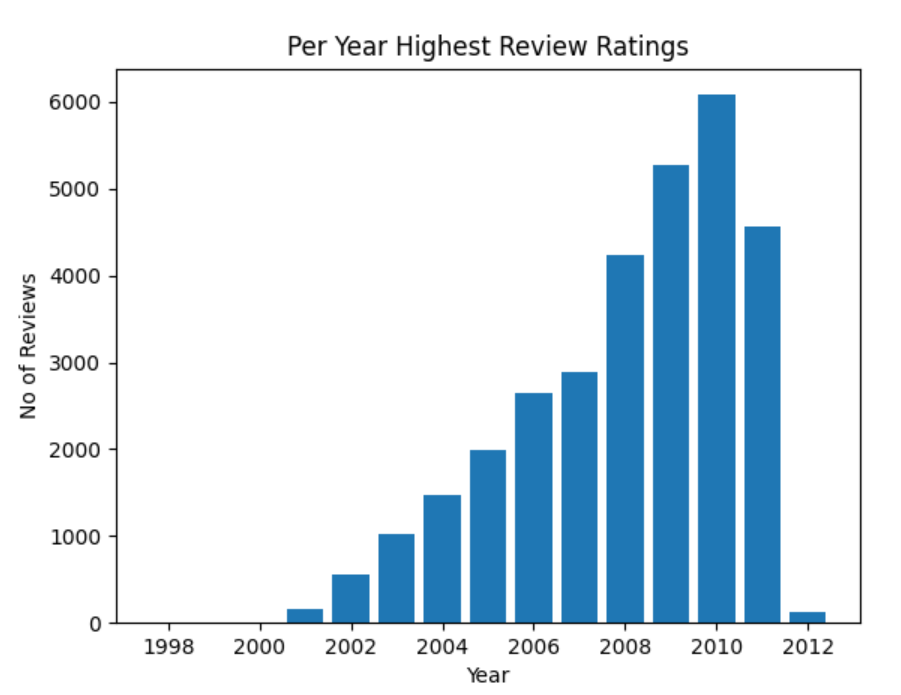
1. Read CSV file using Pandas library
2. Convert the date from numerical to datetime format using datetime library
3. Do grouping/filtering as per each problem
   1. Top breweries by ABV values
   2. Years and overall rating
   3. Number of reviews and overall rating
   4. Beer styles and number of reviews
4. Select top values based on condition/assumption as per each problem
   1. Top breweries - ABV value greater than mean to avoid whole dataset processing
   2. Recommend 3 beers – number of reviews with the higher ratings could be the beers people like the most.
   3. Highest rating per year and their count will give us the highest rating year
   4. Beer style – highest rating with most of the reviews could give us the favorite beer styles
5. Sort the Values
6. Finally select the top values based on user preference

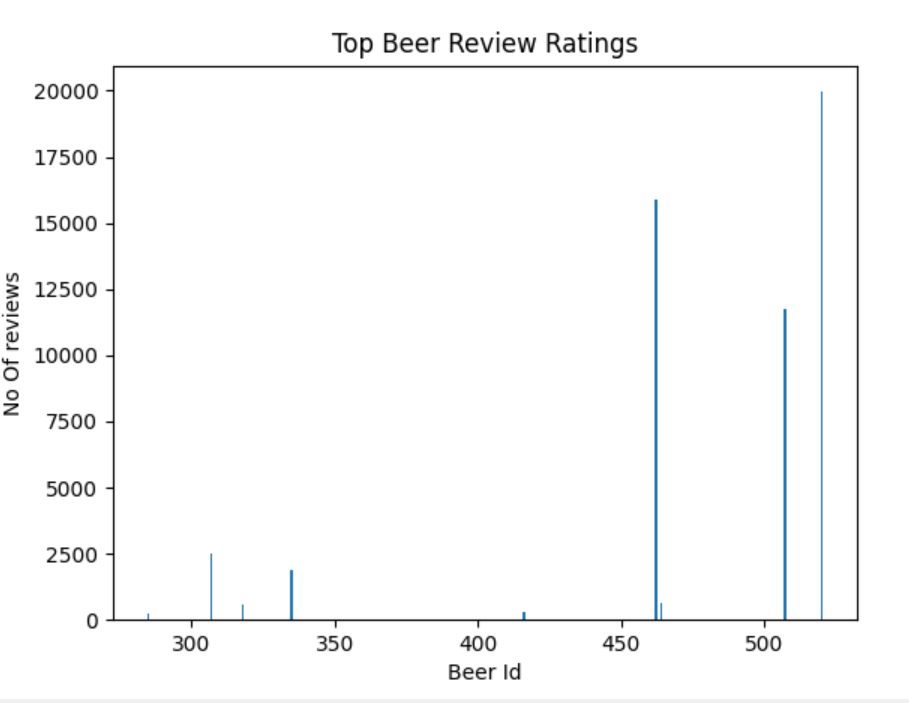
## Output

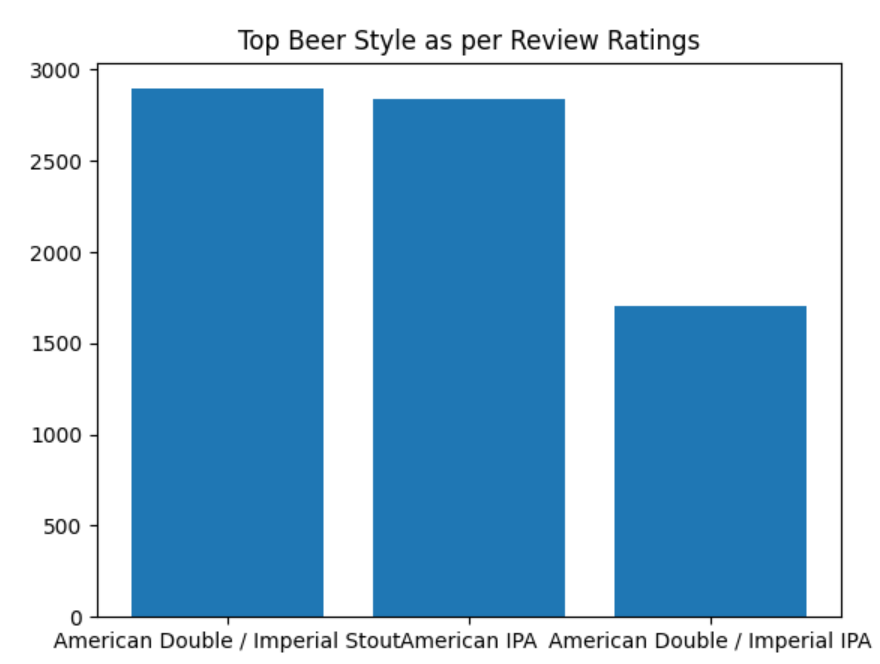
### Console output



### Plot graphs to support the answers







# Feature Engineering

## Problem statement

1. Based on the user’s ratings which factors are important among taste, aroma, appearance, and palette?

## Approach

Here, we are performing the feature engineering on the columns present in the dataset.

As, we are having all numerical columns for the ratings and overall rating score, we may not require standard scaling or other techniques.

To get the correlation between these features (taste, aroma, appearance, and palette) to overall score, we will perform the below correlation methods:

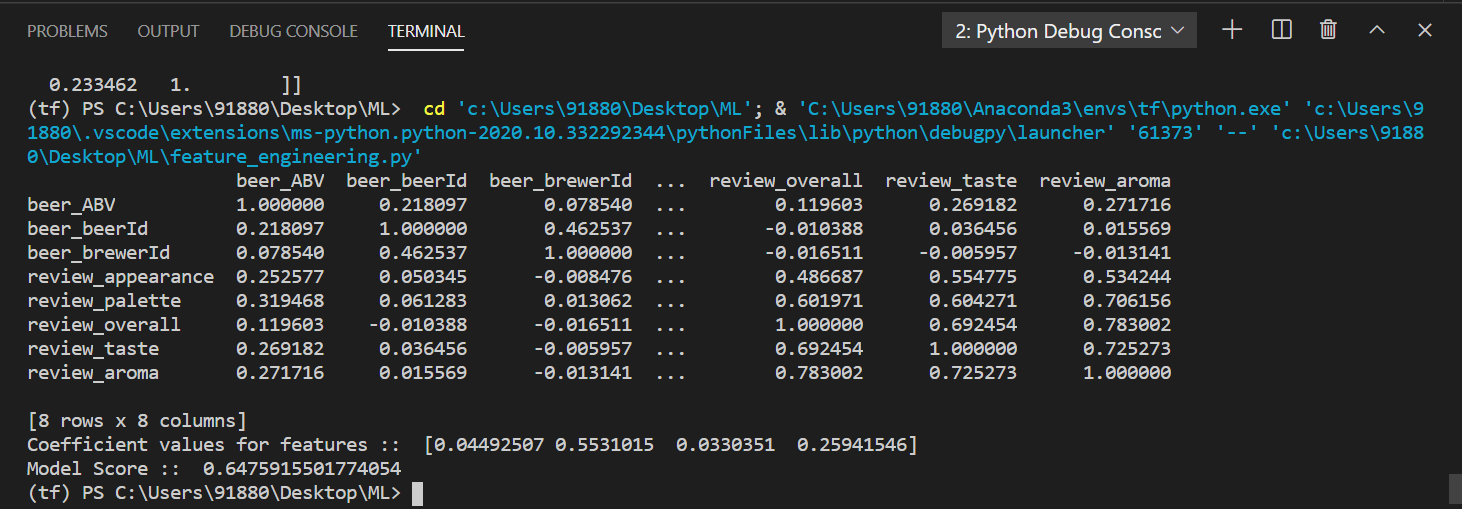
1. Pearson
2. Spearman

These correlations we will get using sklearn library.

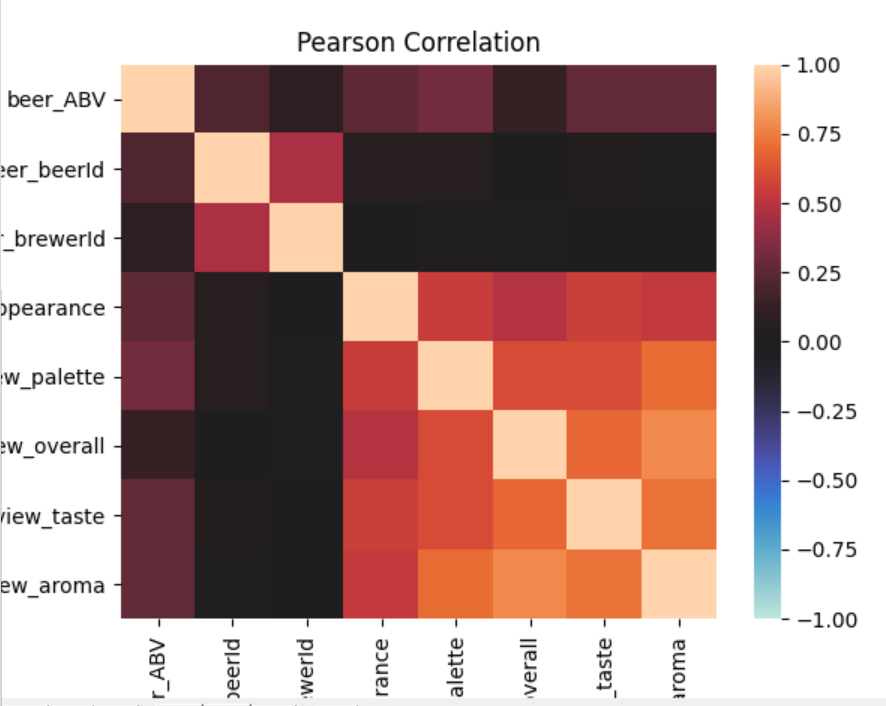
We have observed that taste and aroma are having more impact on the overall score. Please find below plots and output for the same.

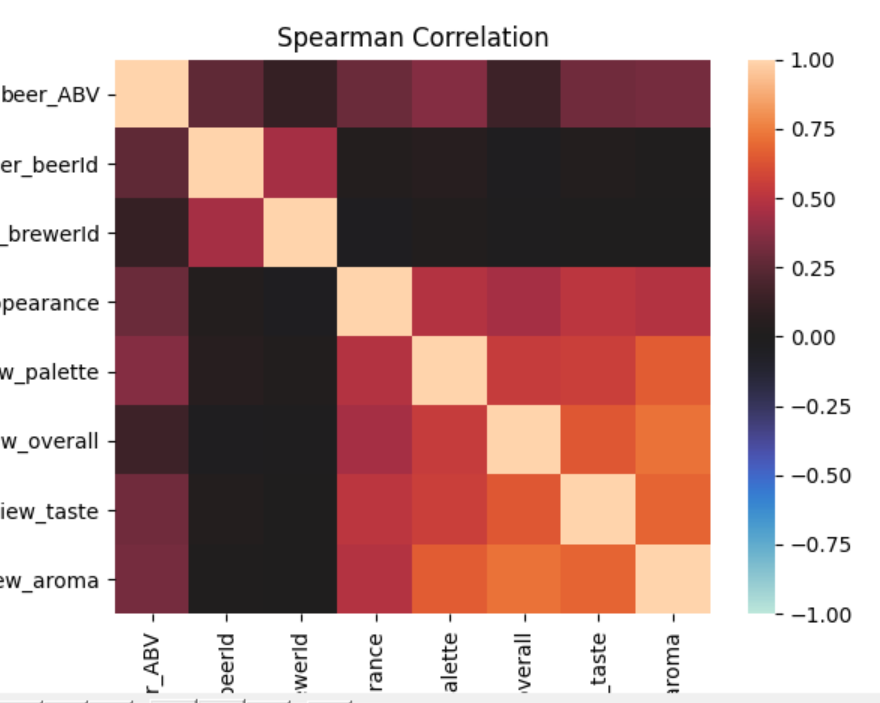
## Output

### Console output



### Plot graphs





# Similarity based on user reviews

## Problem statement

How do find similar beer drinkers by using written reviews only?

## Approach

Here, we need to find the similarity between the drinkers using their written reviews.

We have solved this problem using below techniques:

### Clustering using K – means algorithm

Algorithm:

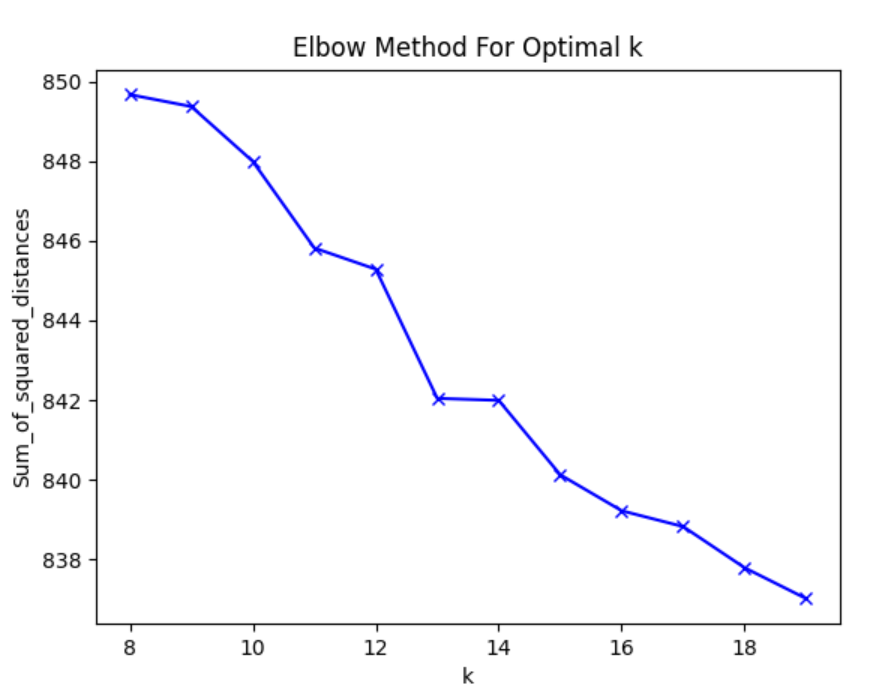
1. Read the dataset using pandas
2. Group the reviews based on the users
3. Find optimal n\_clusters value using elbow method
   1. I have performed several experiments to arrive at n=19 as per 100K reviews
   2. For smaller sets, I was getting optimal k as 6, 8, 12 for 100, 1k, 10k reviews respectively.
   3. We have used elbow method and plotting the graph to choose that value. Refer below plot for the same.
4. Train the model using k-means algorithm using that optimal clustering number
5. Evaluate and cluster the model on the dataset (10k) considered this for quick execution.
6. Map the clusters with the users
7. Refer below output for the smaller samples

### Cosine similarity

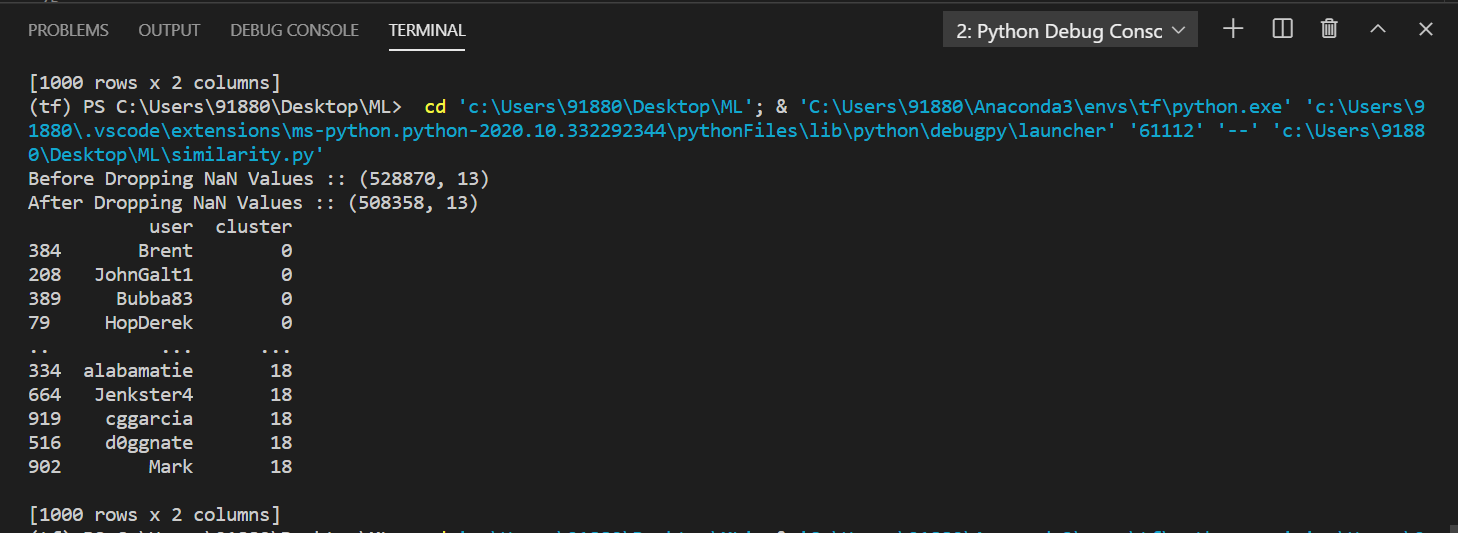
Here, we are calculating the cosine similarity between the two user reviews. We are using sklearn pairwise algorithm for this. It will generate the matrix of cosine similarities between all the user reviews. Finally, this matrix can be used to cluster the user reviews based on criteria like >0.70 between the pairs then they will be in same cluster. Due to time constraints, I am not able to finish the create clustering logic from the generated matrix. Refer the generated matrix output below.

## Output

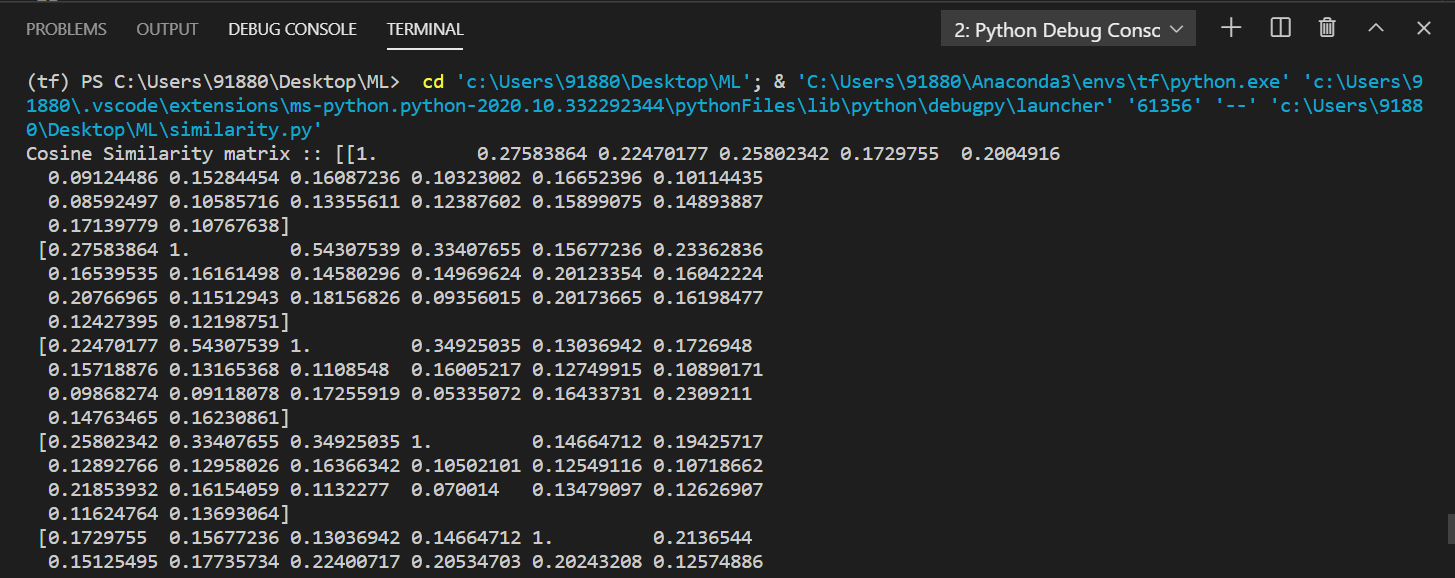
### Select Optimal K using Elbow method:



### Clustering Console Output:



### Cosine similarity output



# Compare rating score on written reviews

## Problem statement

How does written review compare to overall review score for the beer styles?

## Approach

Here, we are trying to identify the statistical approach to compare the overall rating score with the written reviews and how they are corelated. We are using below approach:

### Algorithm

1. We are trying to train and use a general-domain sentiment model to compare the score with ratings. We can also train on the beer challenge dataset with features (aroma, taste as they are more important) by processing written reviews. Though, it may overfit and will not suffice the valid comparison
2. Due to time constraint, I have trained and used a general sentiment model using below steps:
   1. Get Imdb dataset from keras
   2. Get word indexes (top K words only), create the test, train and validation set
   3. Vectorize all the input datasets
   4. Train using deep neural network with layer architecture (input, 2 dense layers) as it is giving more than 98% accuracy, I haven’t used any normalization or dropout.
   5. Create input vectors for beer reviews (group on the beer styles)
   6. Predict the scores on the input (predicted array output)
   7. Create the np array of review score (review\_overall)
   8. Find out the standard error between these two arrays (actual review scores and predicated)
   9. Plot the errors on graph. Refer the below plot graph.

## Alternative Approach

Here, we can also perform statistics on the reviews with the below steps:

1. Read dataset
2. Group all reviews based on beer style
3. Find TF-IDF on all reviews [we can also do bigram/trigram or noun phrase-verb phrase combination]
4. Pre-compute the term frequency map, term to term association map
5. Group all the reviews based on either a single token or phrase iteratively over the reviews till all reviews get associated to clusters
   1. Start with the top-most occurring keyword
   2. Find all the associated terms from pre-computed term-term dictionary
   3. See if step b finds more terms for a particular review and form a cluster based on these terms (we can assign this as a topic) and associate the review to that cluster
   4. Iteratively follow steps a, b, c to create a cluster / associate review to that and maintain the dictionary of the same
6. Due to time constraints, I am not able to finish this but this kind of statistics I have done previously for domain ontology creation, knowledge graph creation, hierarchical clustering, etc.

## Output

### Console output

