TWIN: Personality-based Intelligent Recommender System

Poojith S Shetty, Prashant Reddy, Vinyas Kaushik Tumakunte Raghavendrarao
{Shetty.poo, Reddy.pra, tumakunteraghaven.v}@husky.neu.edu
INFO7390 ADS Fall 2017, Northeastern University

1. Abstract

To create the recommendation system the trip advisor data set, personality score data set, and reviews given by those users to different hotels data set, was taken into consideration. The data has been clustered based on the personality. After the clustering, the kmean cluster value was added as a column to the reviews dataset. Based on the username and Location input, the model looks at the users with similar personalities and outputs all the hotels which have rating 5.

To validate the given recommendation, the input of the username, and personality test inputs were given to to get the cluster. Then the prediction model is passed the kmean value and the hotel name, and using this the rating of the hotel is predicted. If the hotel is present in the training set then we have the predicted rating similar to the rating in the test dataset. The code was later deployed to google cloud using the flask application. The link to the application is: https://tripadvisor-189301.appspot.com/.

2. Introduction and Background

Traditional Recommender Systems collect information from the user explicitly by asking the user to fill in the fields in the user profile (usually demographic data or products ratings) or implicitly by studying user behaviour (logs of purchases, content analysis, etc.) (Tuzhilin, 2005). With the growing interest in the connection between the consumer personality and specific characteristics of the products (e.g. brands) the person is more likely to purchase, (Mulyanegara et al., 2007) the challenging task of introducing the personality dimension into Recommender Systems has arisen.

Background:
Only a few approaches of personality-based user model construction exist in the field of Recommender Systems. One of them retrieves personality information through asking the user to manually fill in questionnaires (Nunes, 2008). However it seems to be at least problematic to require each user to go through this procedure during the profile construction step. Furthermore, people do not always provide sincere answers and incorrect data can produce a negative impact on the quality of the recommendation.
One of the alternatives to questionnaires could be the estimation of the personality from the user generated content that is freely available in many online communities. Much work has been done in the field of psychology to extract specific features from the text to establish the connection between the way the person writes and her personality (Tausczik and Pennebaker, 2009). Thus, the unique approach we follow here is the challenging task of adding the personality dimension to the Recommender System through the automatic personality recognition from linguistic cues from the texts of the users (Mairesse, 2007).

The Dataset for the personality dimension was present and we used the same dataset in the recommendation system to get the recommendations.

3. Datasets

We are using the TripAdvisor dataset
To evaluate the performance of the TWIN system, we apply it in the travelling domain, to suggest hotels from the TripAdvisor site by filtering out reviews produced by people with like-minded views to those of the user.

Parameter
User Name
Date of stay
Text of the review (value, rooms, location, cleanliness, service, sleep quality)
Detailed hotel ratings
Helpfulness

3.1 Personality Recognizer

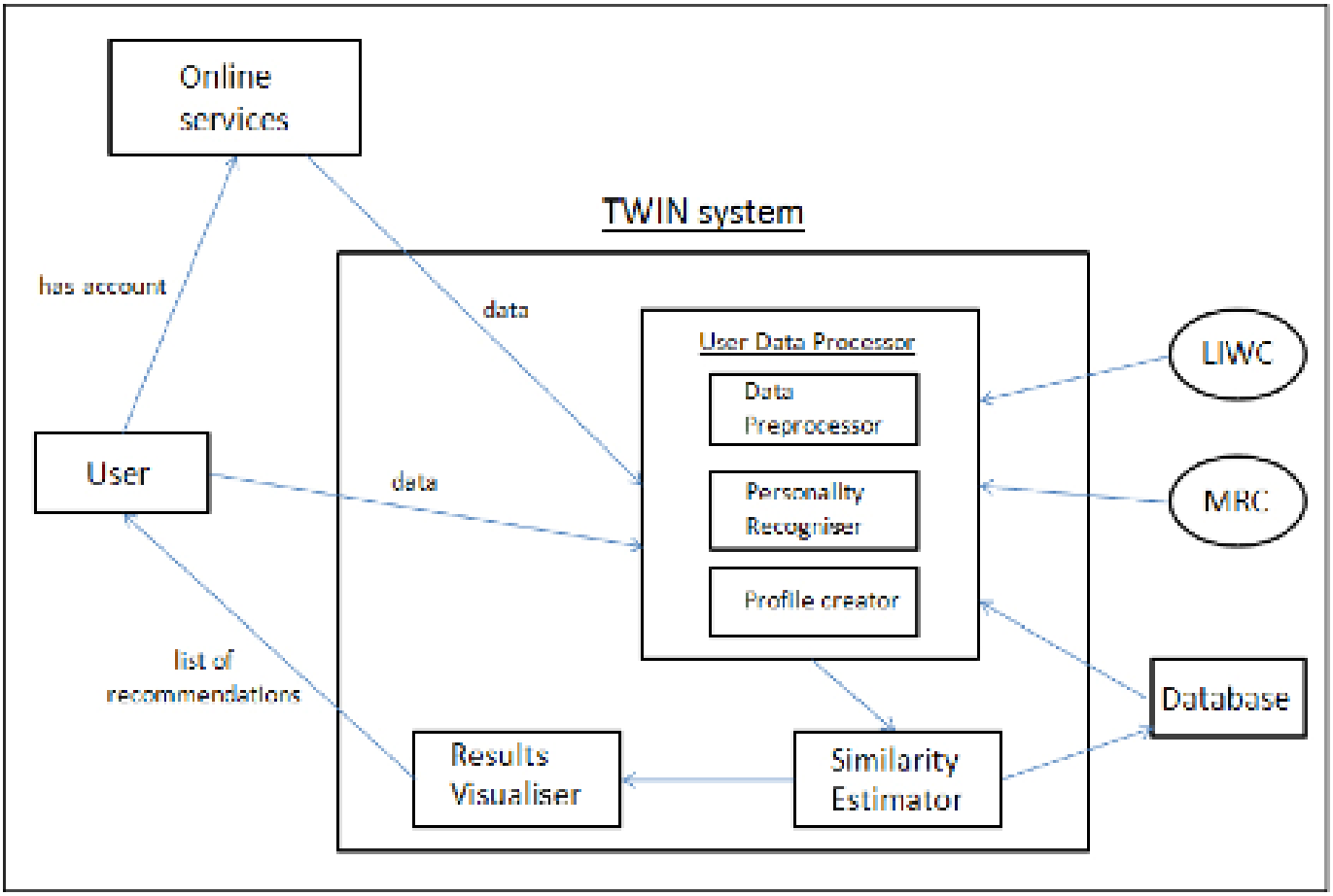
The Personality Recognizer processes the text word by word getting the category of each word and calculating the overall percentage of each of the discovered categories. In order to establish the personality of the author the Personality Recognizer applies WEKA models trained on the Psychology Essays corpora (Pennebaker and King, 1999), which is comprised of texts, associated LIWC categories scores and the real personality scores of the authors collected through the Big Five questionnaire. Finally, 5 scores (corresponding to each of the Big Five dimensions) are produced for the processed text - each of them ranging from 0 to 1 (where 0 means that the trait is weekly expressed and 1 means strong expressiveness).
The ready dataset by the personality recognizer we have used in our project to cluster the users based on the personality Score.

Big Five trait
Openness to experience
Consciousness
Extraversion
Agreeableness
Neuroticism

4. Key topics Used

4.1 TWIN system components

In this Section, we provide a more detailed introduction to the structure of TWIN. The main components of the TWIN Recommender System are presented in Figure



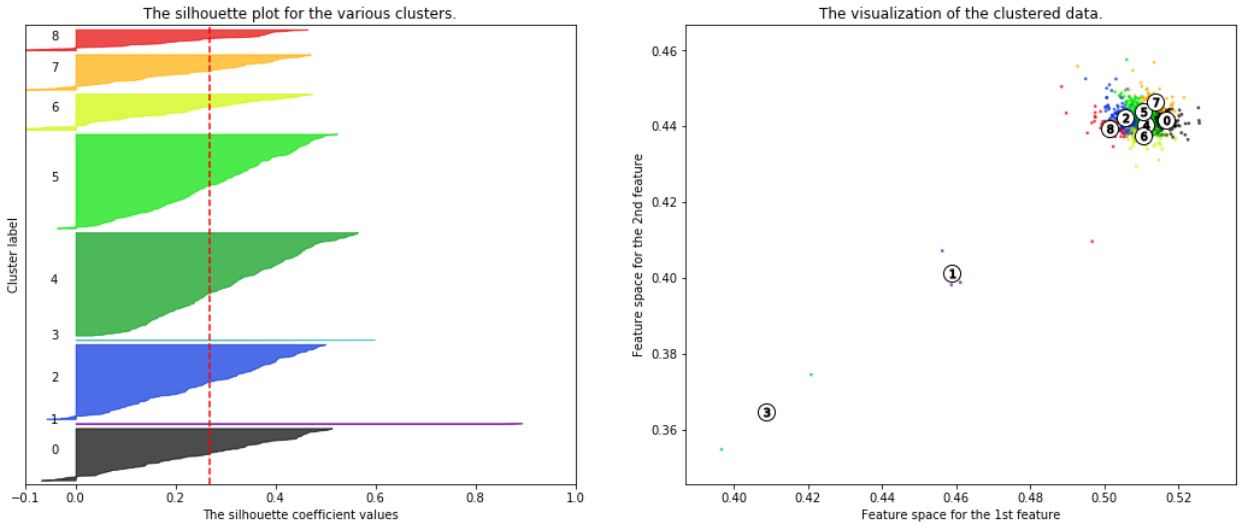
4.2 Data Preprocessing

Since the original dataset is serialized for storage. We have found a way to deserialize the dataset and use them as our training and test data. We have checked and removed null values from data sets

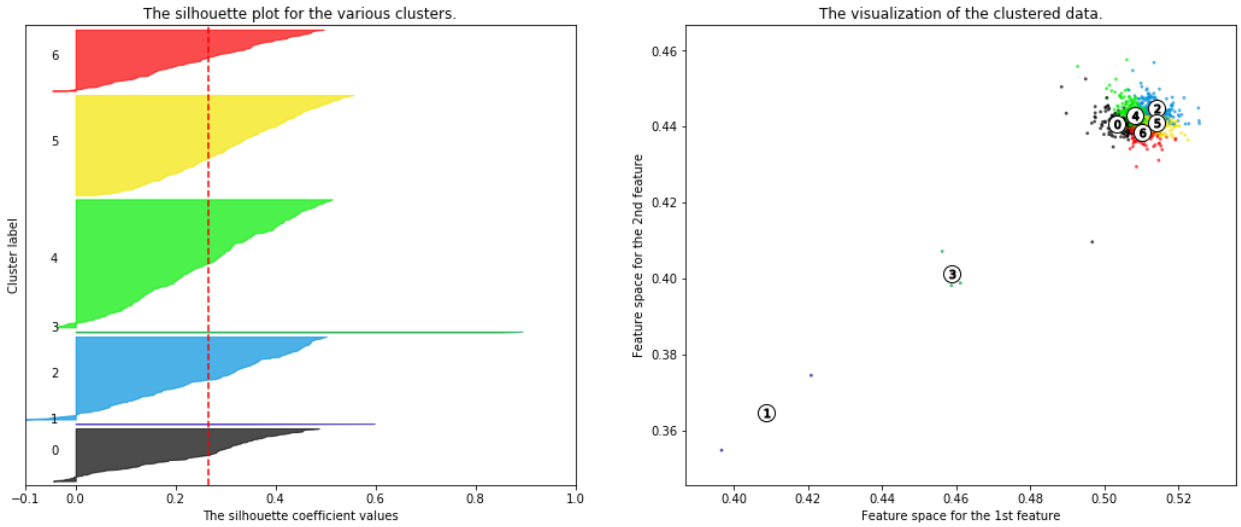
4.3 Silhouette-Analysis

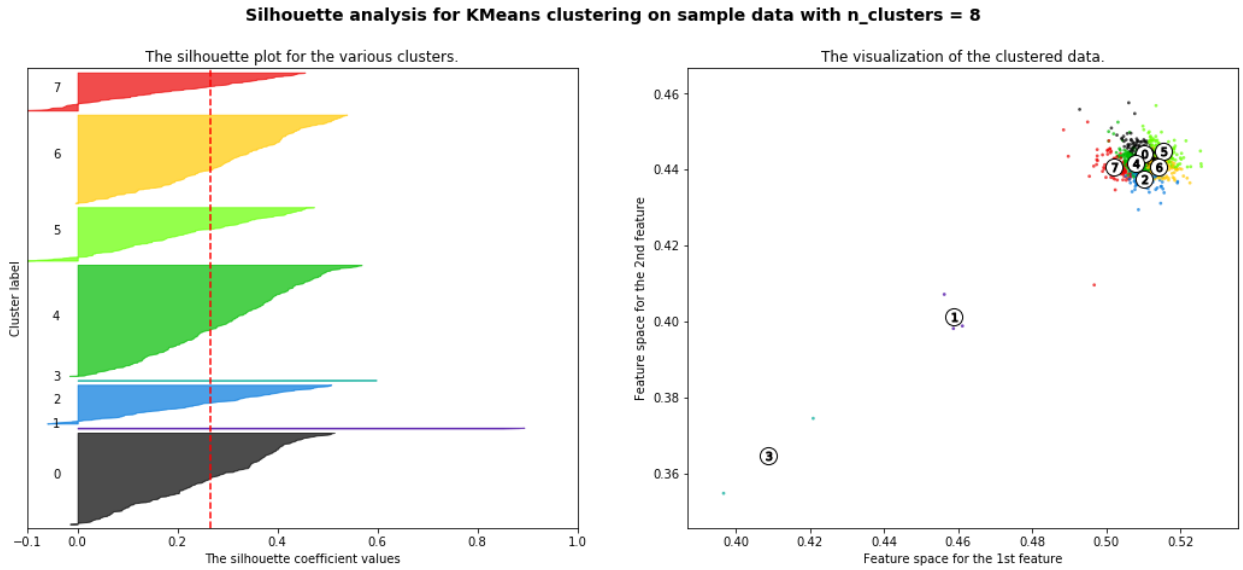
Silhouette analysis can be used to study the separation distance between the resulting clusters. The silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters and thus provides a way to assess parameters like number of clusters visually. This measure has a range of [-1, 1].
Silhouette coefficients (as these values are referred to as) near +1 indicate that the sample is far away from the neighboring clusters. A value of 0 indicates that the sample is on or very close to the decision boundary between two neighboring clusters and negative values indicate that those samples might have been assigned to the wrong cluster.
From silhouette analysis, we can choose how many points we should choose in our K-means clustering.

Silhouette analysis for KMeans clustering on sample data with n_clusters = 9



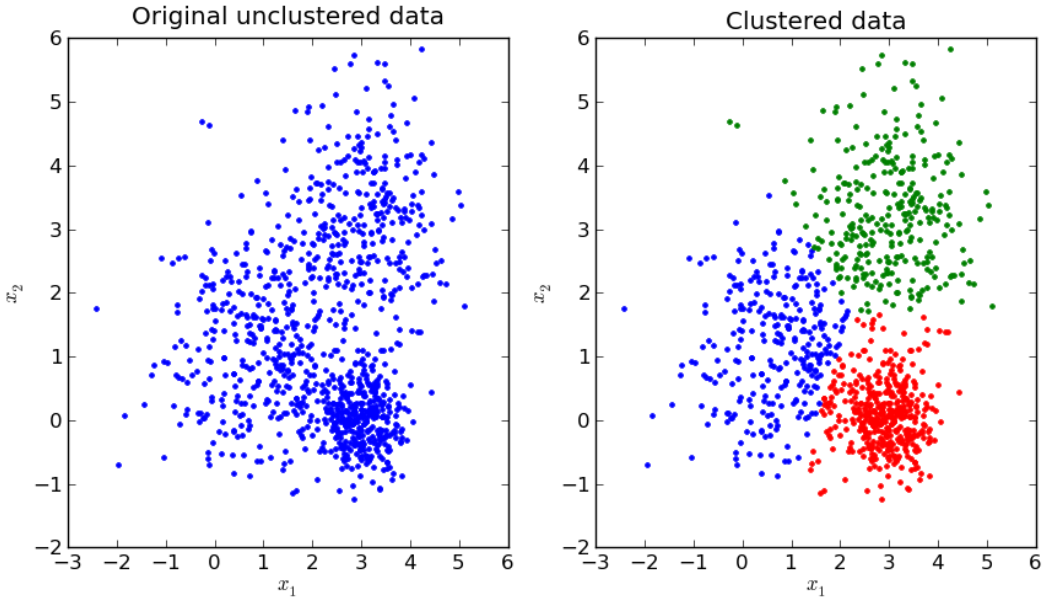
Silhouette analysis for KMeans clustering on sample data with n_clusters = 7





4.4. K-means clustering

K-Means Clustering is one of the popular clustering algorithm. The goal of this algorithm is to find groups(clusters) in the given data.
The k-means algorithm searches for a pre-determined number of clusters within an unlabeled multidimensional dataset. It accomplishes this using a simple conception of what the optimal clustering looks like:
The "cluster center" is the arithmetic mean of all the points belonging to the cluster.
Each point is closer to its own cluster center than to other cluster centers.
Those two assumptions are the basis of the k-means model



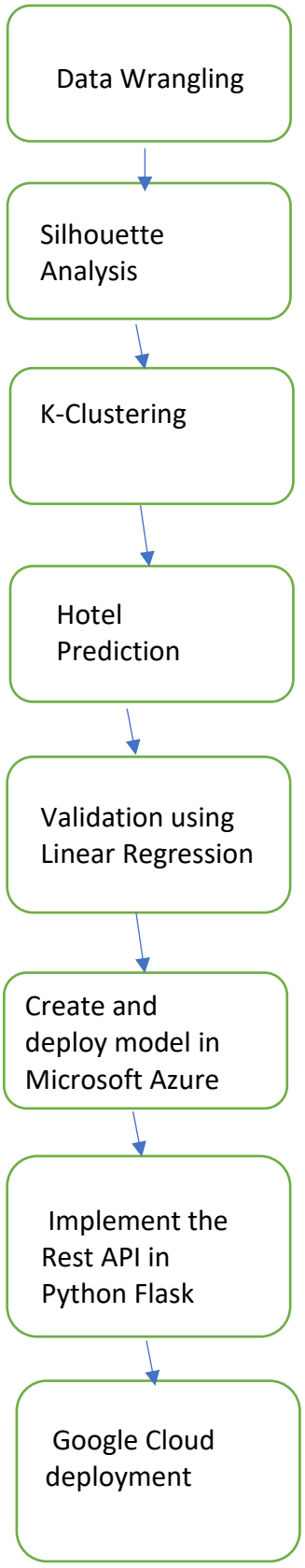
4.5 Multiple Linear regression.

Multiple linear regression (MLR) is a statistical technique that uses several explanatory variables to predict the outcome of a response variable. The goal of multiple linear regression (MLR) is to model the relationship between the explanatory and response variables.

A simple linear regression is a function that allows an analyst or statistician to make predictions about one variable based on the information that is known about another variable. Linear regression can only be used when one has two continuous variables – an independent variable and a dependent variable. The independent variable is the parameter that is used to calculate the dependent variable or outcome.

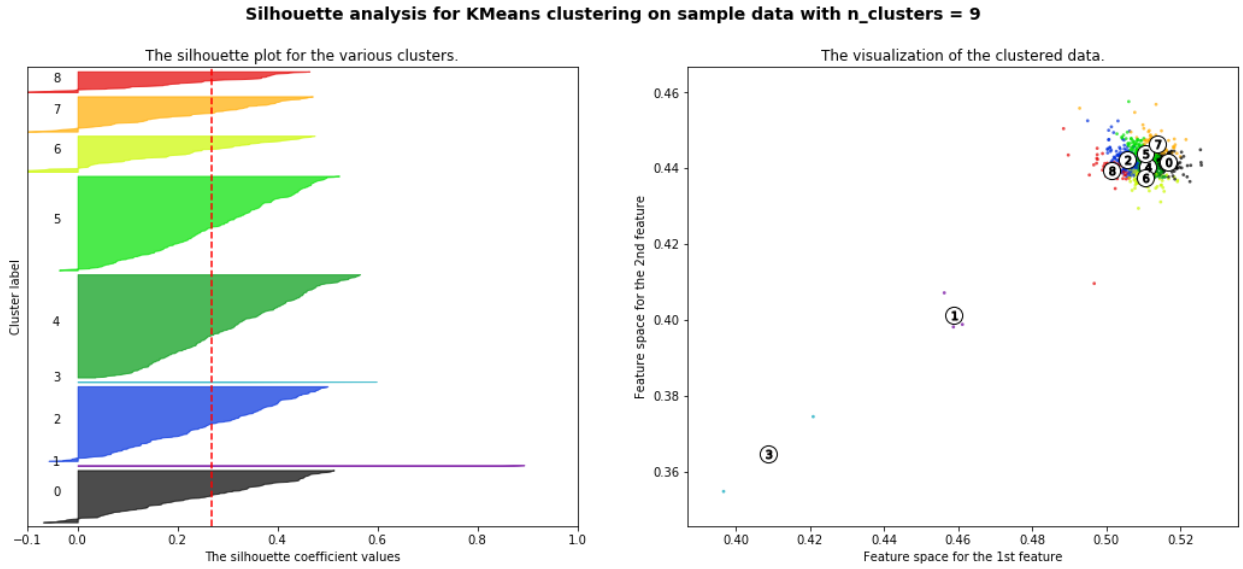
Multiple linear regression (MLR) is used to determine a mathematical relationship among many random variables. In other terms, MLR examines how multiple independent variables are related to one dependent variable. Once each of the independent factors have been determined to predict the dependent variable, the information on the multiple variables can be used to create an accurate prediction on the level of effect they have on the outcome variable. The model creates a relationship in the form of a straight line (linear) that best approximates all the individual data points.
The model for multiple linear regression is: $yi = B0 + B1xi1 + B2xi2 + ... + Bpxip + E$
In our project, to validate the recommendation we have considered the rating to a particular hotel as to be predicted by identifying the cluster of where the particular user belongs and based on the other people in the same cluster, we predict the rating of the hotel from the entered user.

5. Methodology:



a) Silhouette Analysis:

Silhouette Analysis is used to get the ideal number of clusters for K- means clustering.
By using this we found that 9 clusters will give maximum silhouette score.



b) K – means Clustering:

K- means Clustering is used to cluster individual user based on their personality scores provided by the dataset. Number of clusters is taken as 9

```
In [13]: # k-Means clustering
         k_cluster=cluster.KMeans(n_clusters=9)
         k_cluster.fit(norm_data)
         k_cluster.inertia_
         cluster_label = k_cluster.labels_

In [26]: #Saving it as pkl file
         with open('kmeanCluster.pkl', 'wb') as f:
             joblib.dump(k_cluster, f)
```

c) Hotel Prediction:

We are taking username and location as input, based on the cluster and hotel rating given by the users in that cluster we are recommending the hotel to the user.

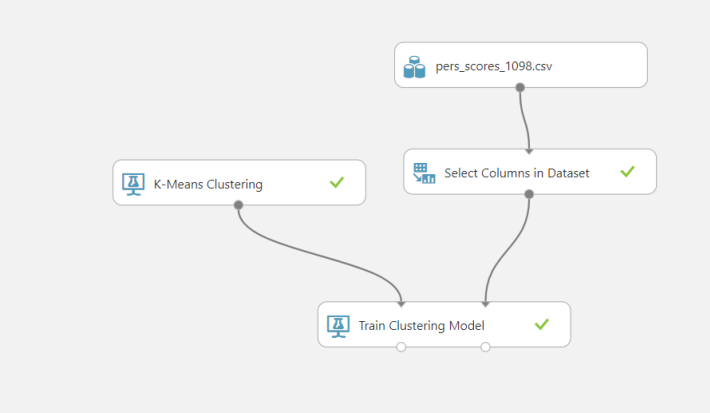
```
In [23]: #Variable the input name and input city
         inputname='007solotraveler'
         inputcity='Stockholm'

In [30]: # Creating list of hotels
         hotels= []

         for index, row in reviews.iterrows():
             if(dictionary[inputname] == row['kmean'] and inputcity==row['taObjectCity'] and row['rating'] == 5 and row['username'] != dic
         #        print(row['username'])
                 hotels.append(row['taObject']);

         print(hotels)

         ['Radisson Blu Waterfront Hotel', 'Vasa Museum', 'Courtyard Stockholm Kungsholmen', 'Stockholm Old Town', 'Stockholm Old Town',
         'Strandvagen 1', 'PA & Co']
```

d) Validating the model using linear Regression.

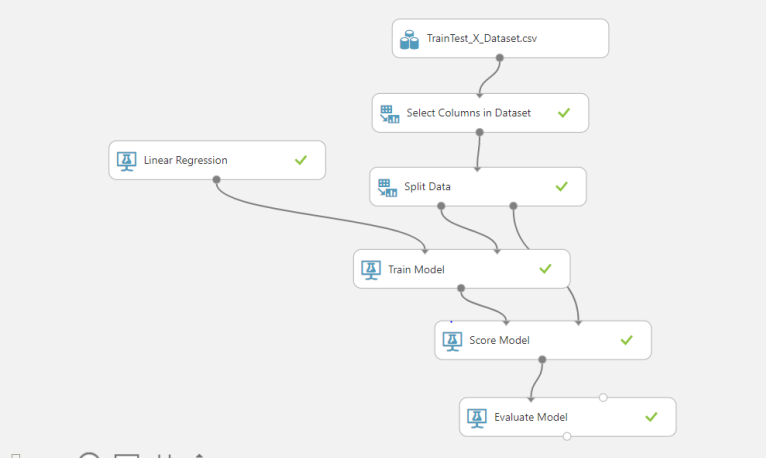
Based on the cluster and hotel we are predicting hotel rating for the user.
By using linear Regression model we got root mean square error value of 0.2835 for mean of 4.438 rating

e) Microsoft Azure:

K-Mean clustering:



Linear regression model:



f) Python Flask application

We have first taken in the input from user for username, open, extra, agree, and neuro to determine which cluster the user would belong to by passing those as input to the cluster API. Then we use that as one of the input in addition to the hotel name the user gave to pass to the linear regression prediction api to display the ratings score to the user.

We have first ran our application locally and ran the below commands to implement the python flask .py file:

- set FLASK_APP=main.py
- set FLASK_DEBUG=1
- python -m flask run

Then once viewing the right input there, we deployed it to google cloud platform. We first created a new project there, then we deployed four files: python .py flask file, html file, yaml file, and requirements file. The yaml file (screen shot below) determines the environment requirements for running the specific application.

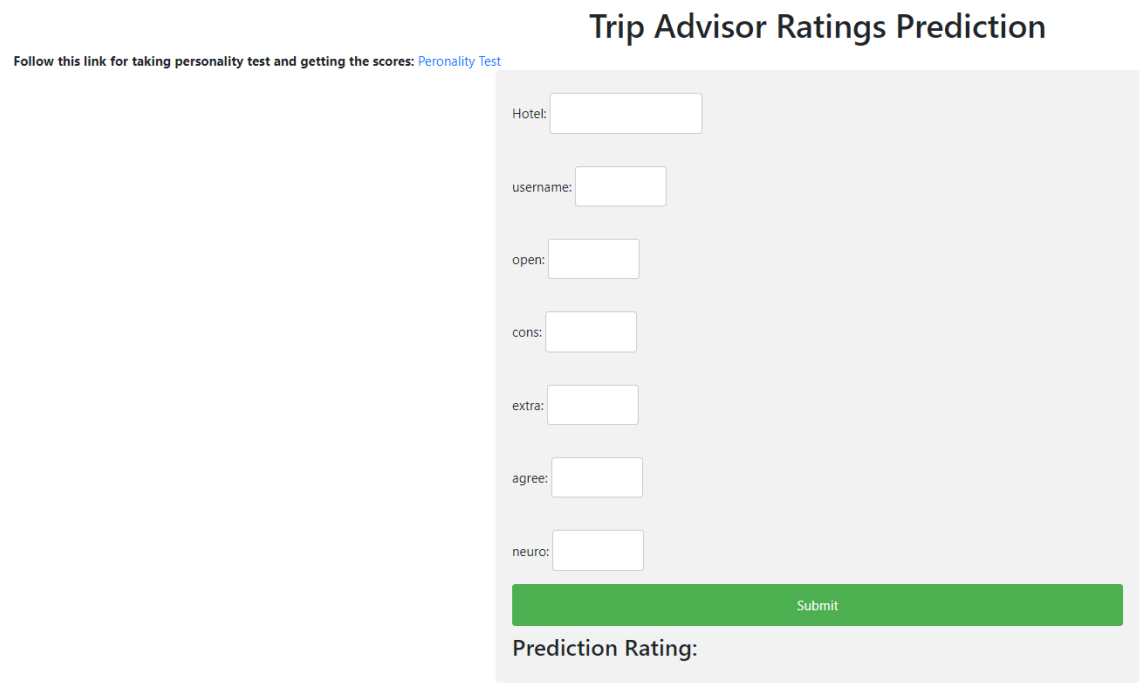
```
runtime: python
env: flex
entrypoint: gunicorn -b :$PORT main:app

runtime_config:
    python_version: 3

```

The python flask file uses the html file as the ui. The requirements file (screen shot below) determines all the python and flask imports the application would need to run.

```
Flask==0.11.1
gunicorn==19.6.0
beautifulsoup4==4.6.0
DateTime==4.2
numpy==1.13.3
pandas==0.20.3
scikit-learn==0.19.1
scipy==0.19.1
urllib3==1.22
virtualenv==15.1.0
requests>=2.9,<3
lxml>=2.2.0
```

g) Google Cloud Deployment:

We have then deployed the code using “gcloud app deploy”.
Below is the screenshot of the application (https://tripadvisor-189301.appspot.com/):



Below is the sample inputs for this site:
'hotel': :'ANA InterContinental Tokyo'
'username': "007solotraveler",
'open': "0.63272",
'cons': "0.55666",
'extra': "0.5636",
'agree': "0.58108",
'neuro': "0.45881",

6. References

1. A. Roshchina, J. Cardiff and P. Rosso. (2015). TWIN: Personality-based Intelligent Recommender System, Journal of Intelligent & Fuzzy Systems, IOS Press, vol. 28, no. 5, pp. 2059–2071, DOI: 10.3233/IFS-141484.
2. F. Celli et al. (2014). The Workshop on Computational Personality Recognition. Proceedings of ACM Multimedia 2014, p.1245-1246, DOI: 10.1145/2647868.2647870.
3. http://twin-persona.org
4. https://en.wikipedia.org/wiki/Recommender_system
5. www.cp.jku.at/research/papers/Tkalcic_Chen_2015.pdf
6. http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html