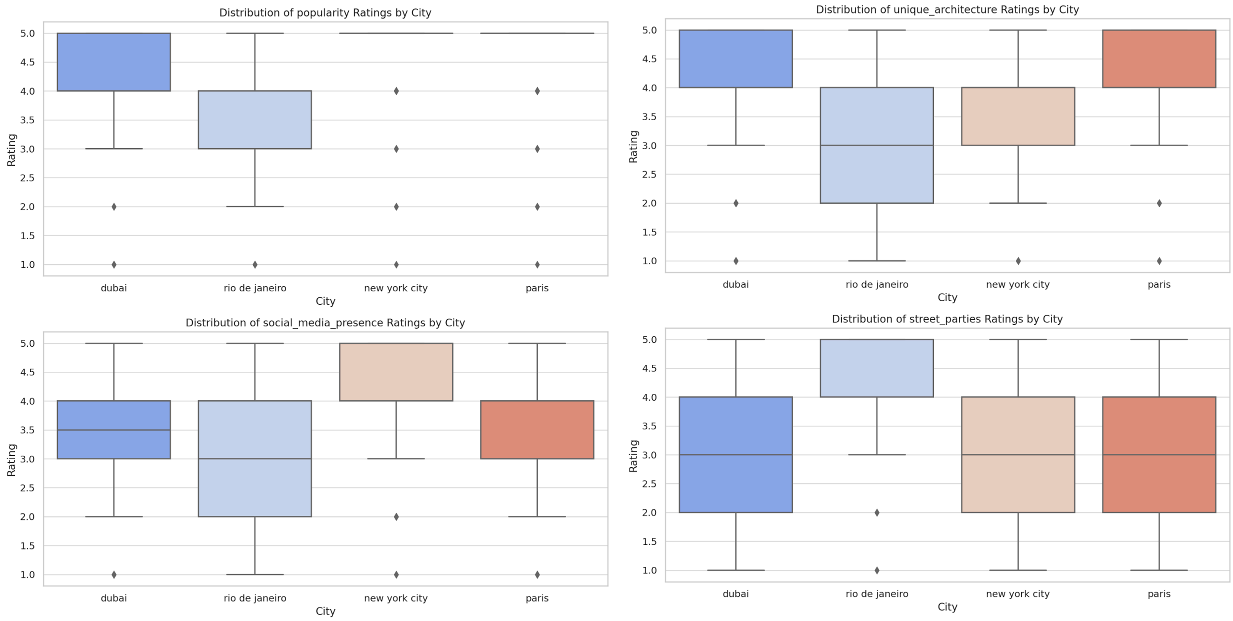
CSC311 ML Challenge Report

Gayatri Sijimon Chakkithara, Haofei Chen, Arindam Thakur, Sizhe Fan

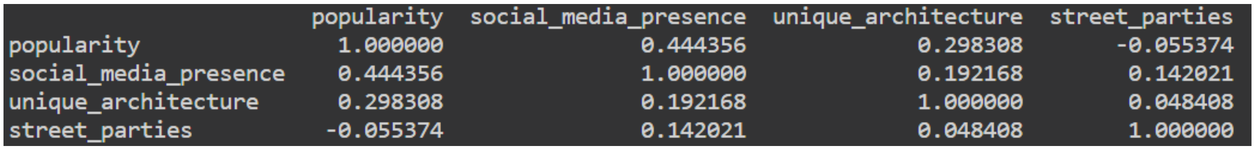
I. Data

1. Data Exploration

1.1 Responses to Questions 1-4



The graphs above show the distribution of responses to Q1 (popularity, top-left), Q2 (social media presence, bottom-left), Q3 (architecture uniqueness, top-right) and Q4 (street parties, bottom-right) of the survey for each city. The dots outside the whiskers are outliers below the 5th percentile or above the 95th percentile. We can see that some features are good discriminators for the label (e.g. popularity below 3 strongly indicates the city being Rio de Janeiro), where others (e.g. sreet_parties) are not clearly separable. For Q1, both NYC and Paris had approximately the mean of 5 which was the highest score possible. For unique architecture Dubai and Paris hold the top spot. While for the remaining 2, which is Social media presence, NYC comes on top and for street parties Rio de Janeiro.

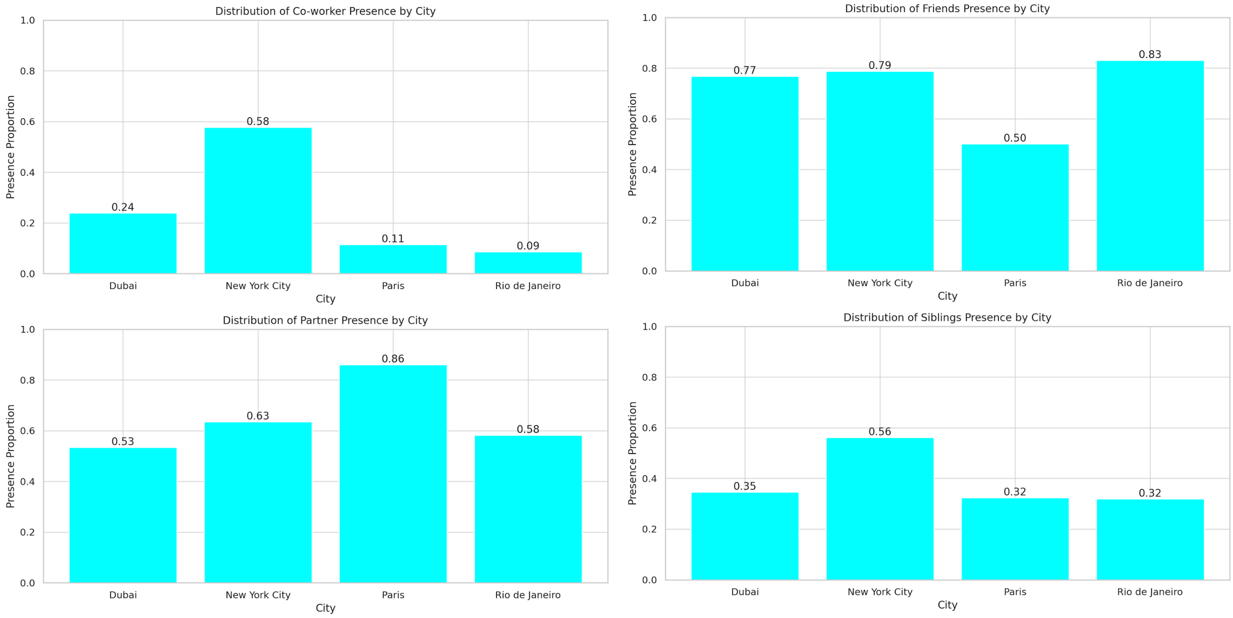


```
                     popularity  social_media_presence  unique_architecture  street_parties
popularity             1.000000               0.444356             0.298308       -0.055374
social_media_presence  0.444356               1.000000             0.192168        0.142021
unique_architecture    0.298308               0.192168             1.000000        0.048408
street_parties        -0.055374               0.142021             0.048408        1.000000
```

This is the matrix which shows the correlation between these features. Logically also in these modern times social media presence in a way directly corresponds to popularity which is shown here with the highest correlation of them all. Unique Architecture and popularity also show significant correlation, not as high as social media presence though. The others are almost negligible.

There are 6 responses with no answer to question 1, 7 with no answer to question 2, 7 with no answer to question 3, and 6 with no answer to question 4.

1.2 Responses to Question 5 (Multiple Select Question)



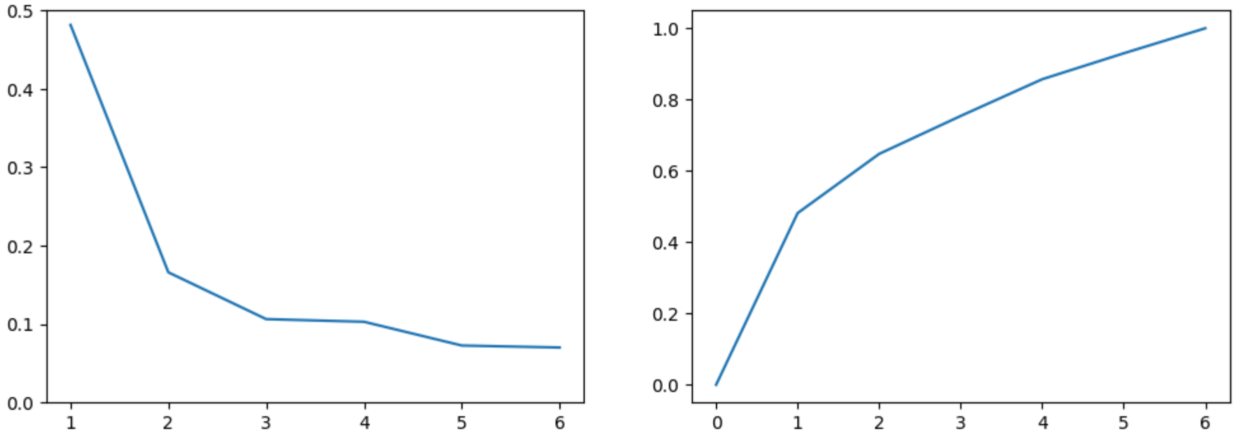
The graphs above show the proportion of responses checking each keyword for each city. Through the bar graphs we can see, except for the distribution of option “Friends”, Q5 helps to predict one single city based on the response. While not being as inclusive and diverse of a question, Q5 still gives us insight into if the response suggests New York or Paris and helps us to negate the possibility of Rio De Janeiro.

1.3 Responses to Question 6 (Keyword Ranking)

For each response, students are asked to rank the 6 keywords on a scale of 1 to 6 in order of relevance to the city. Ideally, we would assume that the six numbers in each response are a permutation of 1 to 6, so that there are only 6! = 720 possible answers. But upon further inspection, we find out that it is not always the case.

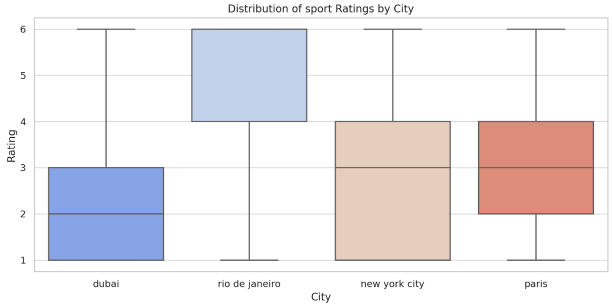
Out of the 1468 responses, 6 are entirely empty (the student did not input a ranking for any of the keywords), 1 is partially empty, and 610 (41.5%) have one or more pairs of keywords ranked equally (e.g. both “Skyscrapers” and “Sport” are ranked the 3rd). This causes the size of the input space to drastically expand to |{1, 2, 3, 4, 5, 6, (empty)}|^6 = 117649.

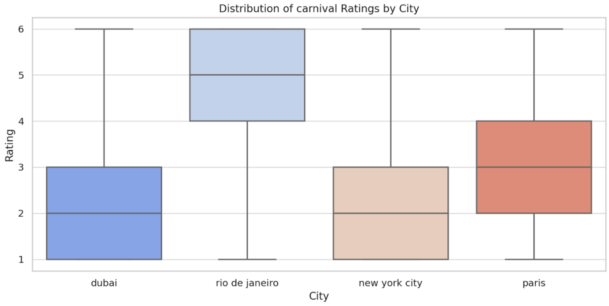
Given that the majority (57.9%) of the responses that are “well-behaved” still lies in a small (0.6%) portion of the input space, we tried PCA to find out if (linear) dimensionality reduction is possible. The graph below shows the ratio of variance explained by each principal component on the left, and the cumulative ratio of variance explained by the first x principal components on the right.

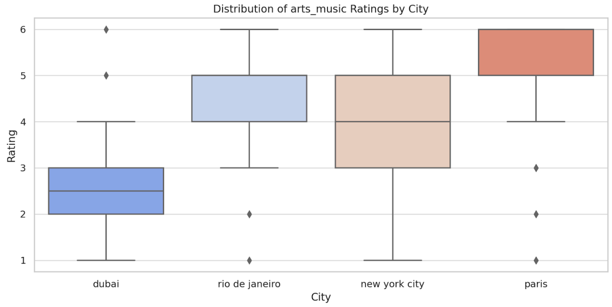


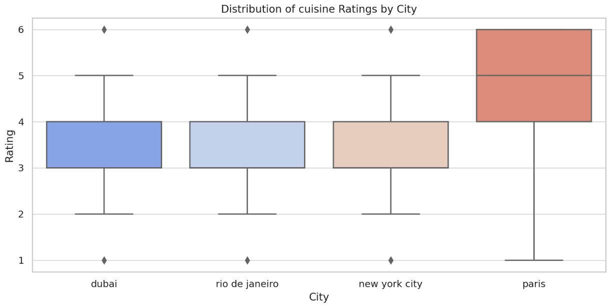
Left: ratio of variance explained by each principal component. Right: cumulative ratio of variance explained by the first x principal components.

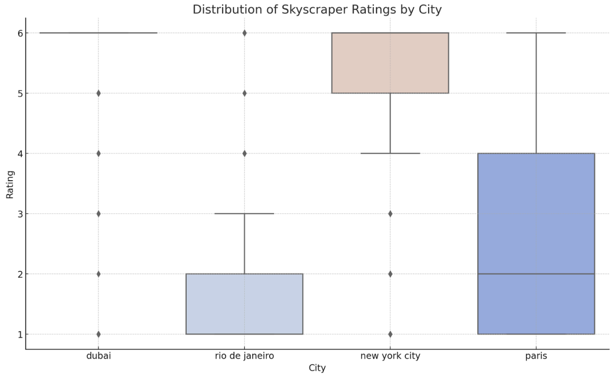
It can be seen that PCA is not effective in dimensionality reduction over the response data, as we still need all 6 dimensions to capture 93% of the variance. Hence, we consider each keyword independently. The graphs below show the distribution of rankings of each keyword grouped by city. Rio de Janeiro is the most distinguishable city among the four, as it almost exclusively has high rankings on Carnival and low rankings on Skyscrapers.

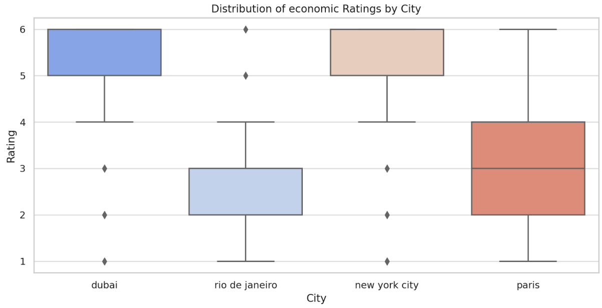


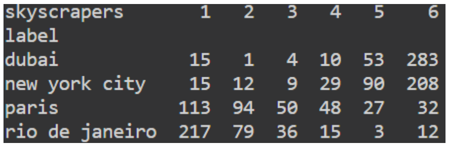




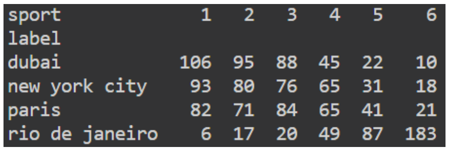




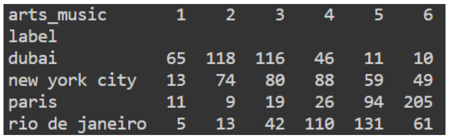




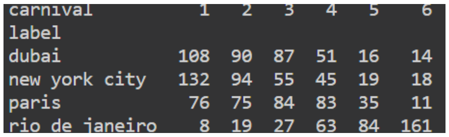
```
skyscrapers      1   2   3   4   5    6
label
dubai           15   1   4  10  53  283
new york city   15  12   9  29  90  208
paris          113  94  50  48  27   32
rio de janeiro 217  79  36  15   3   12
```



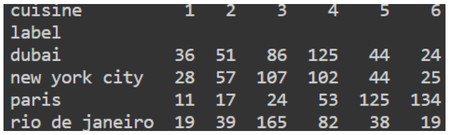
```
sport            1   2   3   4   5    6
label
dubai          106  95  88  45  22   10
new york city   93  80  76  65  31   18
paris           82  71  84  65  41   21
rio de janeiro   6  17  20  49  87  183
```



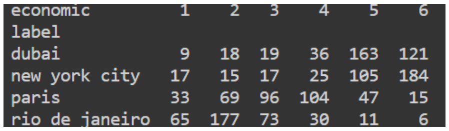
```
arts_music      1    2    3    4    5    6
label
dubai          65  118  116   46   11   10
new york city  13   74   80   88   59   49
paris          11    9   19   26   94  205
rio de janeiro  5   13   42  110  131   61
```



```
carnival         1   2   3   4   5    6
label
dubai          108  90  87  51  16   14
new york city  132  94  55  45  19   18
paris           76  75  84  83  35   11
rio de janeiro   8  19  27  63  84  161
```



```
cuisine         1   2    3    4    5    6
label
dubai          36  51   86  125   44   24
new york city  28  57  107  102   44   25
paris          11  17   24   53  125  134
rio de janeiro 19  39  165   82   38   19
```



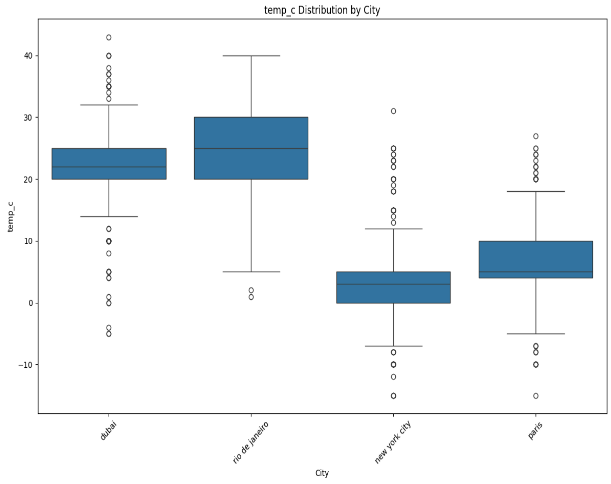
```
economic        1    2   3    4    5    6
label
dubai           9   18  19   36  163  121
new york city  17   15  17   25  105  184
paris          33   69  96  104   47   15
rio de janeiro 65  177  73   30   11    6
```

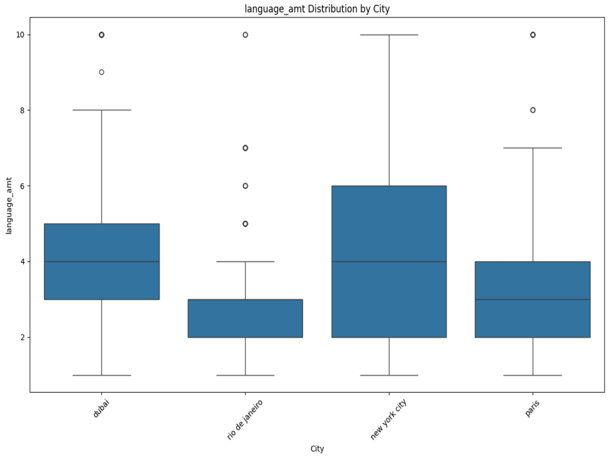
Frequency distribution of rankings of each keyword by city

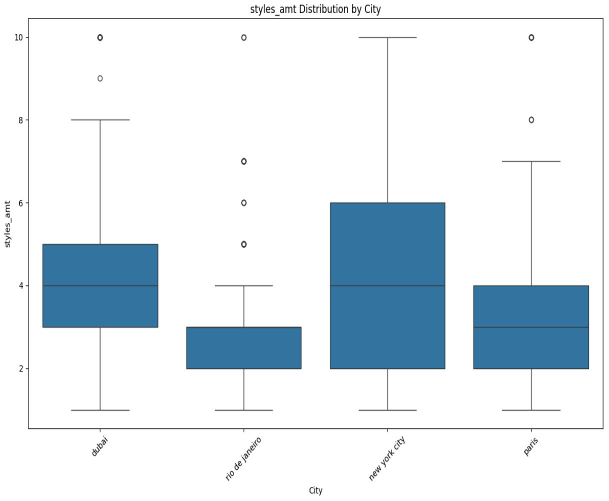
1.4 Responses to Questions 7-9

Responses for questions 7 to 9 suffered from extreme values. The answers to question 7 (temperature) range from -15 to 43 degrees, apart from 3 respondents who answered 89, 1000 and 10000 respectively. The highest temperature ever recorded on Earth, on the other hand, is only 56.7 degrees.

Answers to Q8 (number of languages) range from 1 to 50, apart from 4 respondents who answered 87, 100, 200 and 800 in 5 occurrences. Answers to Q9 (number of fashion styles) range from 0 to 100, apart from 3 respondents who answered 200, 300 and 1000 in 4 occurrences. While these extreme values are apparently unrealistic, they nevertheless reflect some repondants' approach to answering the questionnaire. The graphs below show the distribution of answers to questions 7 (top-left), 8 (top-right) and 9 (bottom), with values outside the 9th-95th percentile range removed.







We see that Average Temperature for the cities for the month of January provides a good insight and helps us to determine and predict the city. Number of languages (Q8) and fashion styles (Q9) don't provide as strong a proof as Q7 for one specific city prediction.

4 respondents did not provide any answer to questions 7, 8 or 9 in 7 of their responses.

1.5 Responses to Question 10 (Free Text Response)

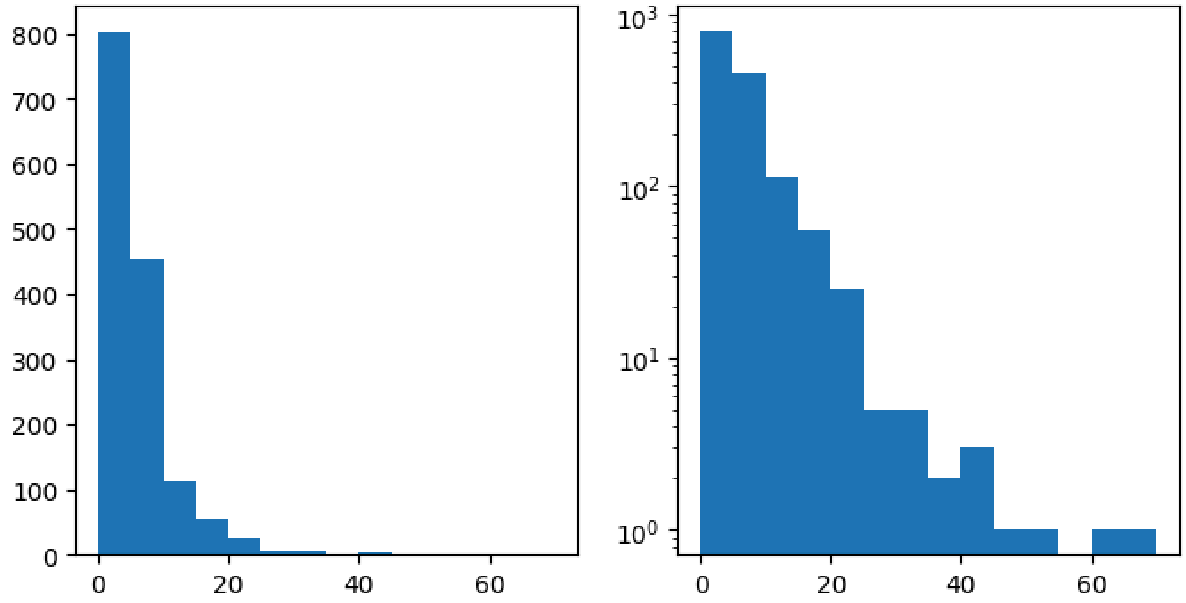
Due to encoding issues, the responses to Question 10 of the survey in the clean_dataset contain non-ASCII characters (such as ©, ¿, §, Œ) that does not form meaningful words. These are relatively rare and we decide to remove them from the dataset. In normalizing the text, we also:

1) Strip the leading and trailing single and double quotation marks and whitespaces.
2) Replace punctuation marks and non-ASCII alphanumeric symbols with spaces, and replace consecutive whitespaces with one space.
3) Convert all letters to lowercase.

Note that single quotation marks in the middle of a sentence are retained to account for words such as "I'm" and "o'clock".

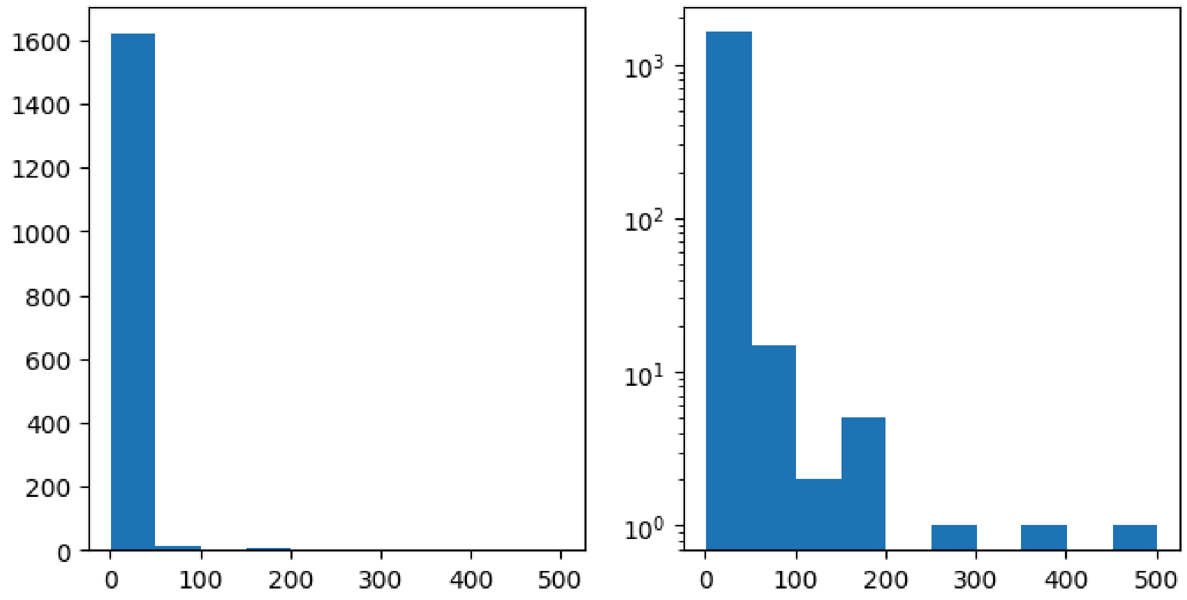
We further split the text into a list of words for each response.

After normalization, the average length of responses is 5.74 words, with the longest responses having 69 words. There are 43 responses that are either empty or contain no ASCII alphanumeric characters. The figure below shows the number of responses in the y-axis against response length (in words) in the x-axis, in both linear (left) and logarithmic (right) scale.



Number of responses against length of response (word count), in linear (left) and logarithmic (right) scale

There are 1646 unique words among the responses. The most common ones are the so-called ‘filler-words’ ('the', 'of', 'a', 'is', 'in', 'and', 'to'). The most common non-filler word is the word ‘city’ with 296 occurrences due to the topic of the survey, unfortunately it does not provide useful information about any particular city. The figure below shows the number of words in the y-axis against their occurrences in the x-axis, in both linear (left) and logarithmic (right) scale.



Number of words in the vocabulary against word frequency, in linear (left) and logarithmic (right) scale

The distribution shows the sparsity of the text data as almost all words are rare, with only 25 words having 50 or more occurrences. It is then not advisable to use the full vocabulary to encode the bag-of-words, due to both (i) the various curses of dimensionality during training, and (ii) that it is very unlikely that the test set will contain the same set of words. Hence, we need a reduced vocabulary that is both reasonably compact in size and easy to 'hit' by the test set.

For each word and each city, we calculate the posterior of that word (i.e. the probability of the label being that city given the presence of that word in the response). Ideally, we want a set of words whose presence most strongly indicates the label being each city. However, most words with high posterior are extremely rare, often occurring just once in the entire corpus (such words will have a posterior of 1 for the corresponding city to the response the word is in, and 0 for other cities). These words are not helpful for prediction due to low probability of having such evidence. To balance evidence and posterior, we calculate the "score" of each word (with respect to a particular city) as posterior + λ * number of word occurrences in the corpus, parameterized by λ. With λ=0.01, we take the union of the top 20 words for each city as our reduced vocabulary. In this way, we obtain a set of words that have both strong correlation to each city and relatively high probability of occurrence. The list of words selected for each city, together with their posterior, number of occurrences and scores is detailed in Appendix A.

The top 20 words for each city happen to be disjoint, so our reduced vocabulary is of size 80. With only 4.9% the size of the entire vocabulary, it covers 23.2% of the word occurrences and 66% of the responses (968 out of 1468 responses have at least 1 word in the reduced vocabulary).

2. Data Representation as Input Features

Each response is converted to a vector of length 97, of which the first 4 fields correspond to the response to questions 1-4, the following 4 fields correspond to the response to question 5, the next 6 fields correspond to the response to question 6, followed by 3 fields corresponding to the response to questions 7-9. The last 80 fields are based on the response to question 10. The following subsections detail the feature extraction process, as well as the treatment for missing data.

2.1 Responses to Questions 1-4

The answers to question 1-4 are represented in the vectors as-is, with an integer value between 1 and 5 inclusive. For responses with missing values, we replace it with an uninformed guess of the mode among the responses with that particular city label. For responses in the test set where the label is not seen, we replace the missing value with the mode among all responses to that particular question.

2.2 Responses to Question 5 (Multiple Select Question)

Responses to question 5 are represented as a binary vector of length 4. The first (second, third, fourth) value being 1 means ‘Friends’ (‘Co-worker’, ‘Siblings’, ‘Partner’, respectively) is selected in the response, and a value 0 indicates that the corresponding term is not selected in the questionnaire.

2.3 Responses to Question 6 (Keyword Ranking)

Each response for this question will be represented as a vector of length 6, corresponding to the number ranking for each of the following keywords: “skyscrapers”, “sport”, “art and music”, “carnival”, “cuisine”, “economic”. The value for each field is an integer between 0 and 6 inclusive, where 0 represents missing ranking.

2.4 Responses to Questions 7-9

The answers to questions 7-9 are represented in the vectors as-is, except that:

1) For question 7, values above 50 are capped at 50, and values below -50 are replaced by -50.
2) For question 8, values above 24 (the 99th-percentile value) are capped at 24, and values below 0 are replaced by 0.
3) For question 9, values above 100 (the 99th-percentile value) are capped at 100, and values below 0 are replaced by 0.

Similarly to questions 1-4. the missing values in each question will be replaced with an uninformed guess of the mean among all the responses to that question (after the extreme values have been capped).

2.5 Responses to Question 10 (Free Text Response)

Using the reduced vocabulary in section I.1.4, we construct a bag-of-words for each response. The feature only concerns the presence or absence, rather than the frequency of each word. Hence, the feature is a binary vector of length 80. A value of 1 in the i-th position of the feature vector means the i-th word (in alphabetical order) of the reduced vocabulary is present is the response, and a value of 0 means the absence of the i-th word. Missing values are treated the same as empty responses, with a vector of all 0's.

3. Training/Validation Set Splitting

As the test set consists of survey responses from the instructors and the TAs whereas the clean_dataset provided are responses from the students, the model needs to generalize the feature-label relationship from the response given by one set of people to that given by other people. Hence, we split the given dataset into training and validation sets based on the ID o f the respondents (the 'id' column) only, so that for any ID x, either all four responses given by respondent x (one for each city) are in the training set, or all four responses are in the validation set.

We use 10-fold cross validation to tune the hyperparameters of our models. To do so we first randomly split the set of all respondent IDs into 10 subsets. In iteration i where 1≤i≤10, subset i is the validation set whereas the union of all but the i-th subset is the training set. The accuracy of the model is the average accuracy on the validation set over all 10 iterations. To account for the influence of the random process in splitting the subsets, the above process is repeated for 10 rounds and we further average the accuracy over the 10 rounds as the overall accuracy of the model, parameterized by the chosen hyperparameters.

To ensure a fair comparison between models and hyperparameters, the data splitting is done only once and the same 100 training/validation splittings is used across all models and hyperparameter configurations.

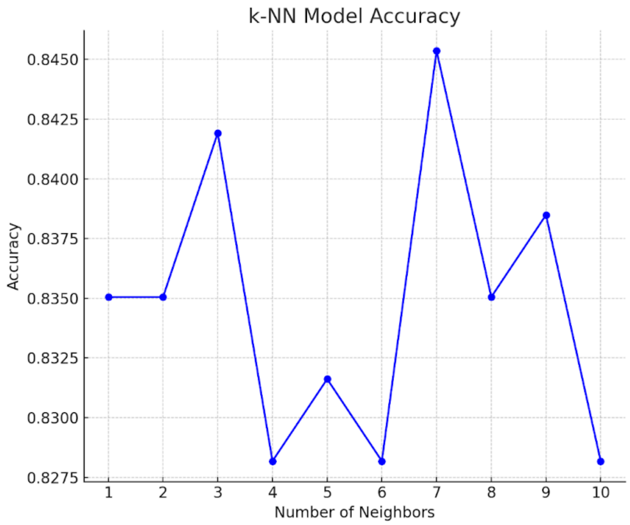
II. Model Exploration

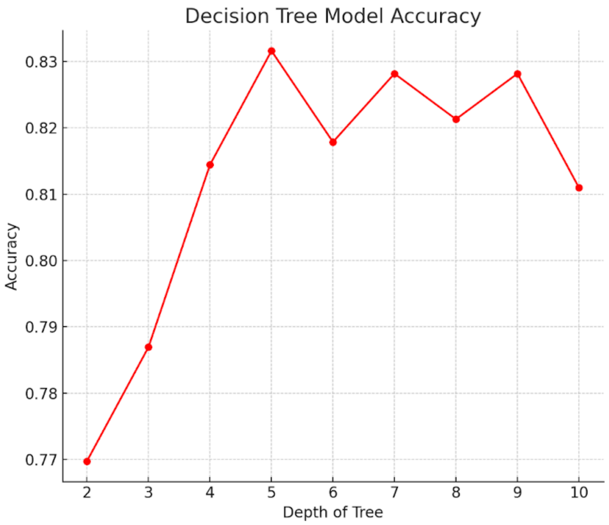
1. kNN and Decision Tree

For the first 2 model exploration, we explored kNN and Decision tree(Best Decision Tree and Random Forest) . Using the Sklearn library we simulated models starting with kNN. For kNN we tried different numbers of neighbors like ranging 1-10. The best testing accuracy we got was for the number of neighbors = 7 which was 84.5%. We also explored the model by trying out different sets of features but the final results were lower. For example by just using Q6, which includes ratings of cities by Skyscrapers, Fashion,Economy etc, we got an accuracy of 60%.

For the Decision Tree we started with Sklearn and made the basis of the tree which gave us 83.16% accuracy. We tried out different depths but the accuracy approximately stops increasing around depth 5. So for the finalization we chose the Accuracy value that the decision tree with depth 5 gives which is 83.16%. Similarly, trying out different sets of features like Q6 gave us an accuracy of 53.2% and we also tried Q7-9 which was around 40%.

Although Random Forest is not reproducible just for the sake of comparison we did choose to explore how the Random Forest performs on our processed data vs non processed data. For the original dataset we were getting around 60% accuracy which increases to 87% after scrubbing through the data. Although other models did not show this much of a significant increase in accuracy before and after processing data, they still did show quite significant increase, like the kNN model went from 60% accuracy to 83.16% and Decision Tree with depth 5 went from 51.5% to 81.7%. This gave us confidence that the steps taken in processing data are on the right track, and helped us to understand how and which features give us the best predictions and also how classifying them differently can help with increasing the model's overall accuracy.



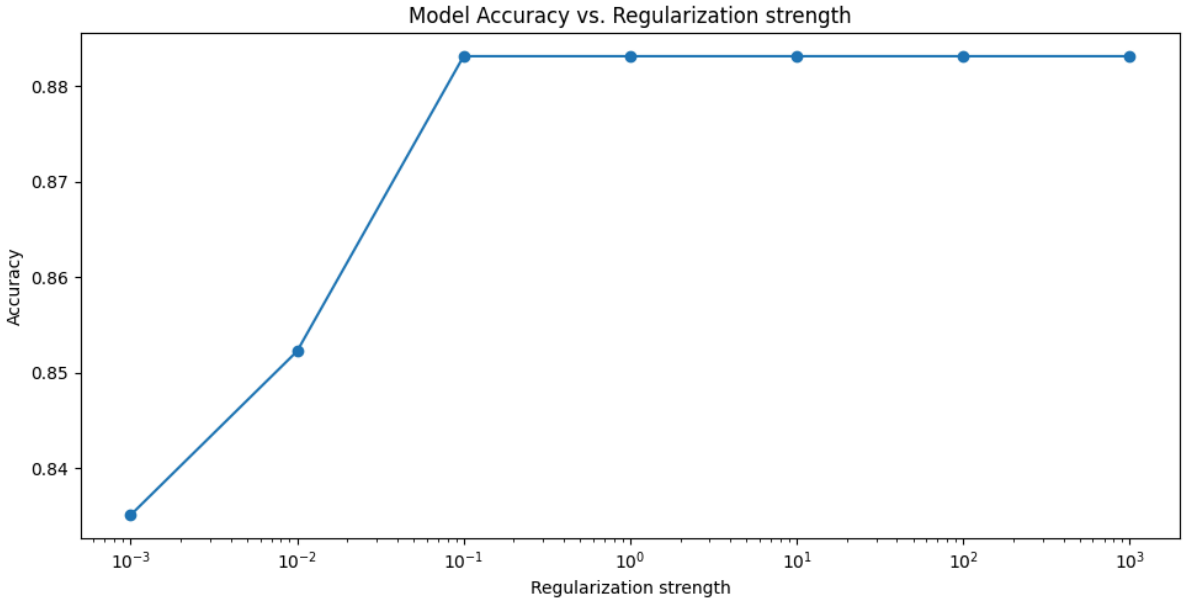


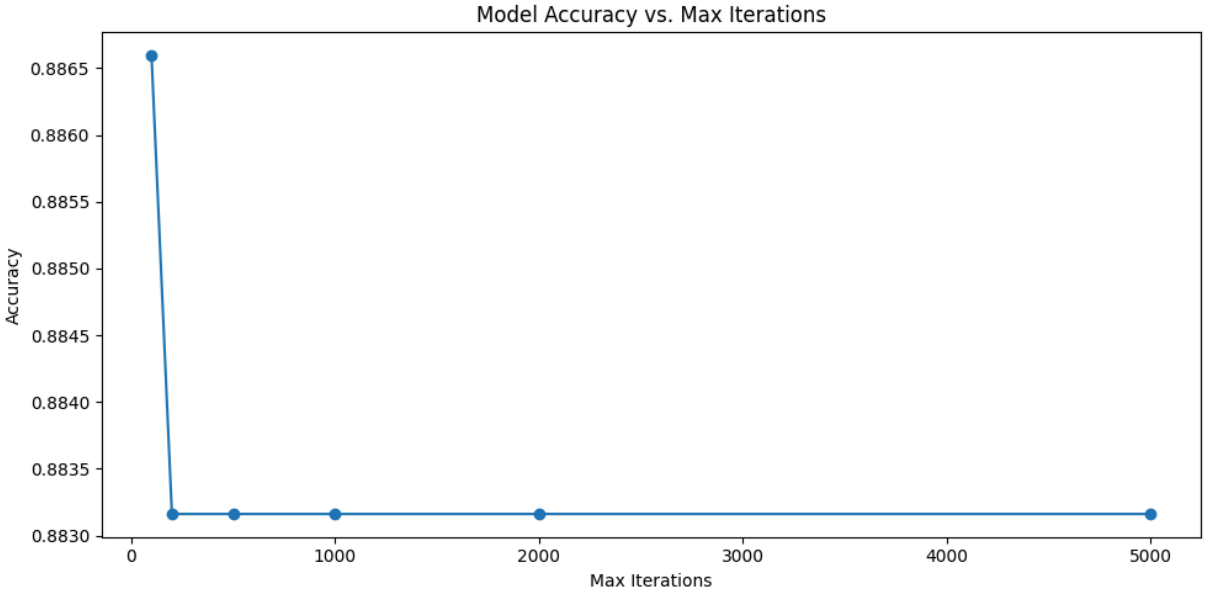
2. MLP

We explored MLP classifiers using the scikit-learn library, specifically using a model with two layers of 100 neurons each and default hyperparameters, the test accuracy is 0.84 when q10 is not included. After including q10, the test accuracy reaches 0.87, and upon further processing of q10, the test accuracy improves to 0.92.

3. Logistic Regression

Explored Logistic Regression using Sklearn's Logistic Regression module, we got 88% accuracy at the best possible tuning. It was difficult to integrate Q10 so this best accuracy did not include Q10. After trying many vectorial methods to try and integrate Q10 in this model it did not work. Hence by using top 5 words(excluding basic grammar words) used for each city bag of words approach, and integrating that into the csv file (it added 20 columns) we were able to get an accuracy of 98%, though it made the csv file and the whole model really large and really hard to replicate without using the Sklearn Library. As for the logistic model, the accuracy becomes stagnant after 500 max number of iterations and by using L2 regularizer of lower or equal to 1 strength. Due to the difficulty in integrating Q10 and making the data fit for use for Logistic regression we chose not to use this model.





III. Model Choice and Hyperparameters

1. Evaluation Metrics

We used accuracy over the validation set (the ratio of the number of correctly classified points to the total number of points in the validation set) as the measure to evaluate and compare the performance of various models and hyperparameter settings. We chose accuracy as the measure in this project for the following two reasons:

1) The validation sets are balanced. This is ensured by splitting the training/validation sets based on respondent ID only, where each respondent has to provide one response for each label. Similarly, we know that the test set will be balanced, because each instructor/TA has to provide one response for each city.
2) The (real-life) cost of all cases of misclassification is expected to be the same. As we are only predicting cities, the potential applications of this project are unlikely to involve crucial decisions (e.g. medical diagnosis) in real life; this project also does not involve classification based on characteristics of people which could lead to potential discrimination. Hence, we expect that the cost incurred by a misclassification of data point x as y will be the same, regardless of which cities x and y are.

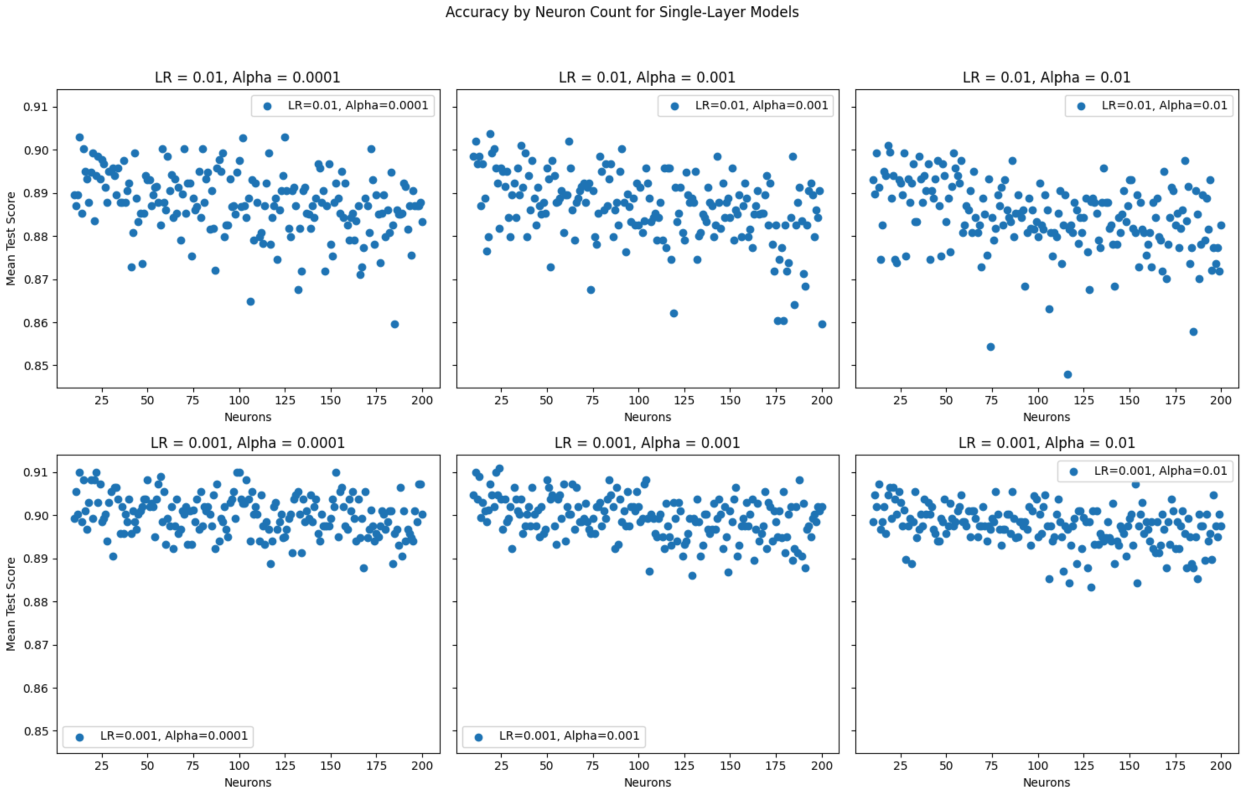
The evaluation metrics for different models and hyperparameter settings are comparable because the same training set and validation set are used in each round of evaluation. The influence of the random process in splitting the training and validation set is further addressed through multiple rounds of cross-validation, as detailed in Section I.3.

2. Model Hyperparameters

In our evaluation of the MLP model, we employed gridsearchcv from the scikit-learn library to explore a wide range of hyperparameters. The configurations we tested included:

1) Hidden layer sizes: Configurations ranging from 1 to 3 layers, with each layer containing 10 to 200 neurons.
2) Alpha (L2 regularization): Tested values were 0.01, 0.001, and 0.0001, aimed at minimizing overfitting.
3) Initial learning rate: 0.01 and 0.001.
4) Maximum iterations: Consistently set at 500 to allow sufficient training without premature stopping.

We used a 10-fold cross-validation on the grid search, the best-performing model configuration reached an accuracy of 0.91. The best configuration featured two hidden layers with 21 neurons each, an alpha of 0.001, and a learning rate of 0.001.



Accuracy by Neuron Count for Single-Layer Models

LR = 0.01, Alpha = 0.0001

LR = 0.01, Alpha = 0.001

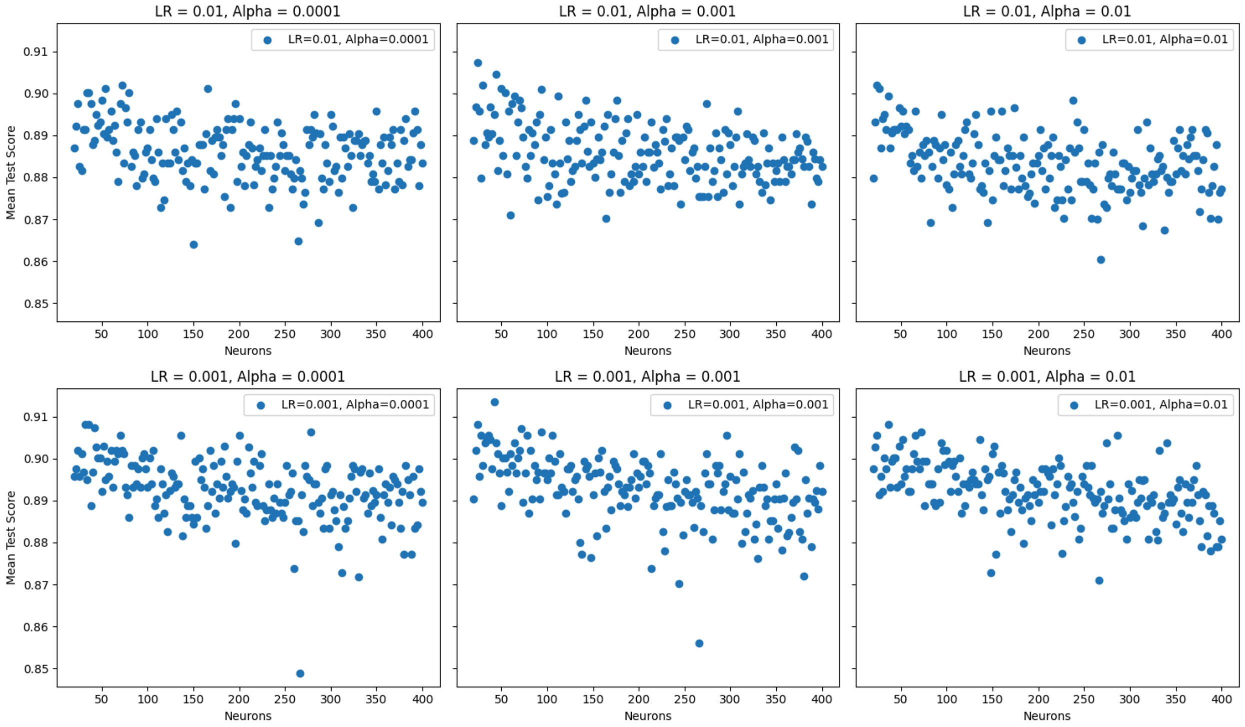
LR = 0.01, Alpha = 0.01

LR = 0.001, Alpha = 0.0001

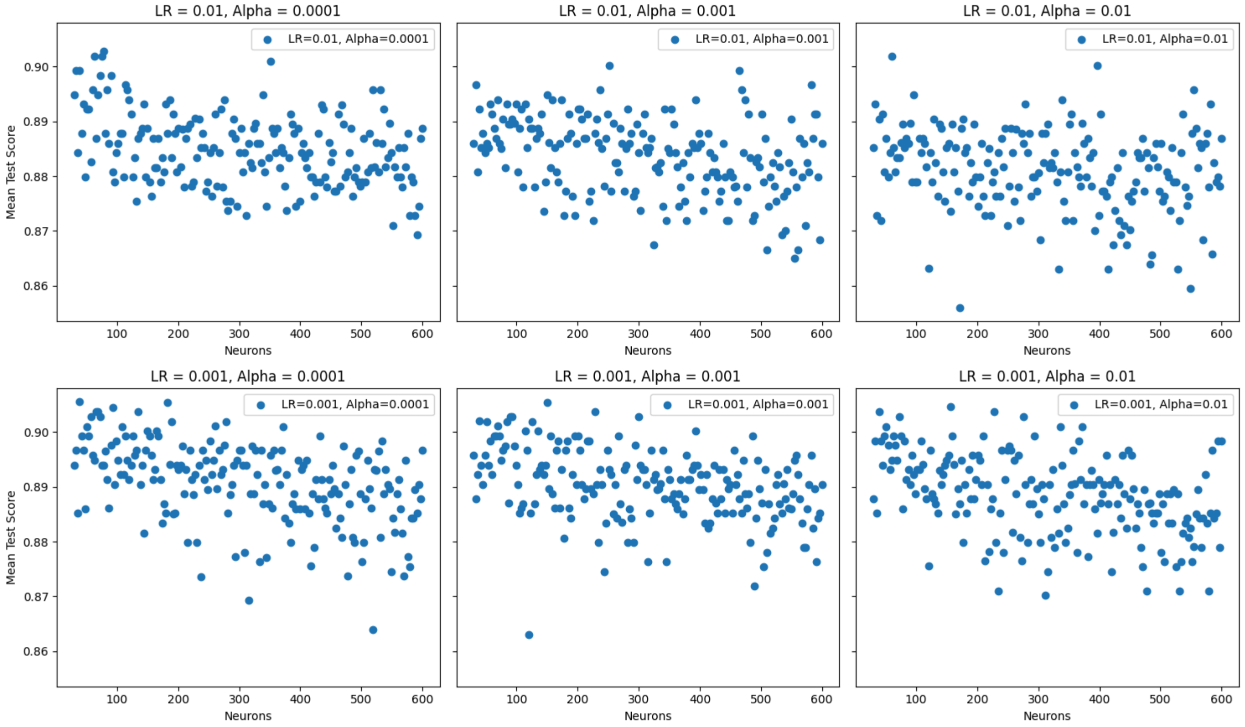
LR = 0.001, Alpha = 0.001

LR = 0.001, Alpha = 0.01

Accuracy by Total Neuron Count for Double-Layer Models



Accuracy by Total Neuron Count for Triple-Layer Models



3. The Model in pred.py

The predict_all method in our pred.py file performs the following tasks:

1) Read the data from the given CSV file to a numpy 2D matrix of size [N, 97], where N is the number of responses (rows) in the CSV file. This is done by calling the read_data method. The content in each row of the matrix is documented in Section I.2.
2) Make a prediction for each response. This is done by calling the predict method, which runs the MLP model and returns the city with the highest output. The MLP model has two hidden layers with 21 neurons each. It uses ReLU in each hidden layer and softmax in the output layer. The weights of each connection and the biases of each neuron are listed at the top of the pred.py file.

predict_all returns a list of city names of length N.

IV. Prediction

We expect our model to achieve an accuracy of around 0.87 on the test set, which is a bit lower than the accuracy on the validation set.

The training and validation sets come from the responses provided by the same cohort of students taking the same class. They only represent a unique group of people with a lot of common characteristics, and are not necessarily a fair representation of the source of the test set, namely the instructors and TAs. Therefore, we expect our model to have limited ability to generalize to the responses given by people from other backgrounds, and hence will have lower accuracy on the test set than on the validation sets.

V. Workload Distribution

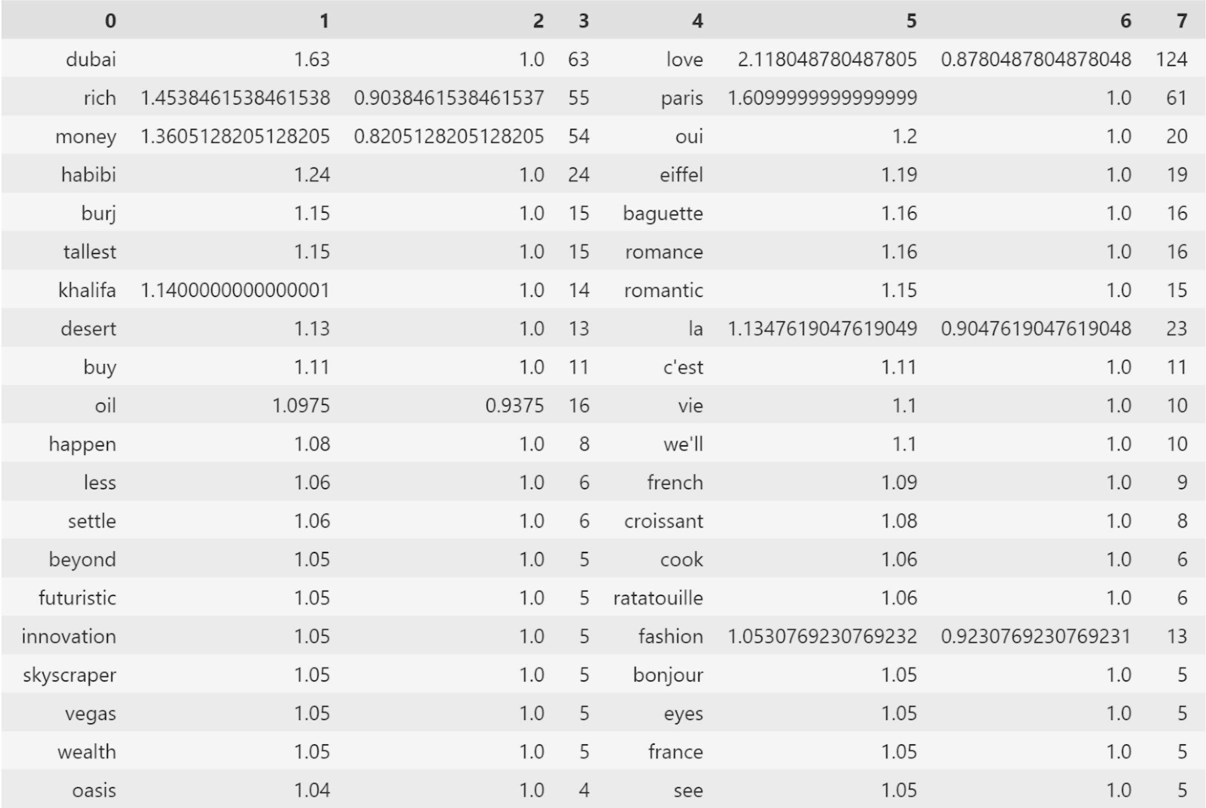
Haofei performed data exploration on Q6 and Q10, coded the data preprocessing (converting responses into input features) part of pred.py, and wrote Sections I.1.3 to I.3, III.1, III.3, IV, and Appendix A of the report.

Arindam performed data exploration visualizing data as boxplots ,correlation matrices and bar graphs where required. Coded data processing to remove outliers and fill missing values. Wrote sections of Q1-4,Q5 and Q7-9. Performed Model exploration on kNN, Decision Trees and Logistic Regression

Tara performed data cleaning and generated an analysis dataset representing each description token as a one hot vector. Built the MLP model without utilizing outside frameworks such as tensorflow or Sklearn. Components such as back propagation, front propagation, and optimization were implemented using numpy.

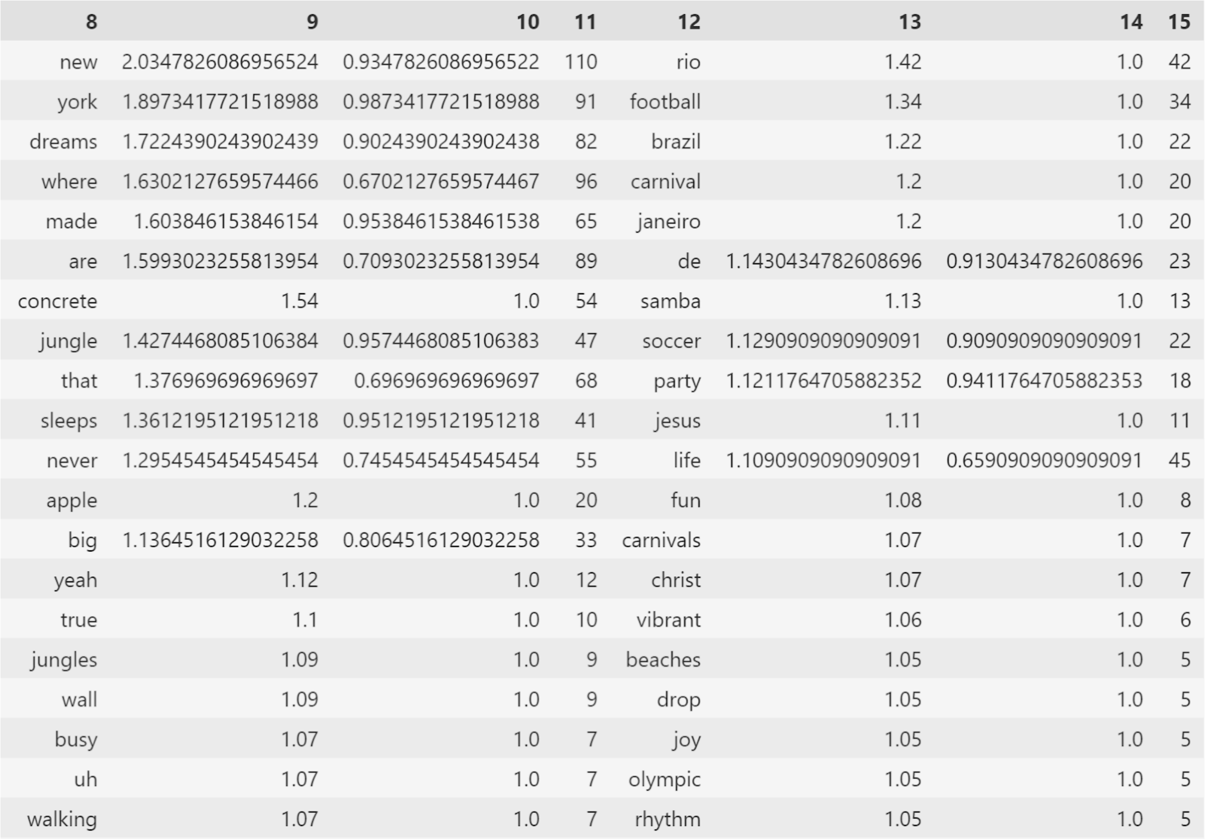
Sizhe performed model exploration on MLP. Fine-tuning the hyperparameter for the MLP model. Extracted the weights and biases from the best-performing model and apply it in the final model in pred.py.

Appendix A. Reduced Vocabulary for Question 10



0	1	2	3	4	5	6	7
dubai	1.63	1.0	63	love	2.118048780487805	0.8780487804878048	124
rich	1.4538461538461538	0.9038461538461537	55	paris	1.6099999999999999	1.0	61
money	1.3605128205128205	0.8205128205128205	54	oui	1.2	1.0	20
habibi	1.24	1.0	24	eiffel	1.19	1.0	19
burj	1.15	1.0	15	baguette	1.16	1.0	16
tallest	1.15	1.0	15	romance	1.16	1.0	16
khalifa	1.1400000000000001	1.0	14	romantic	1.15	1.0	15
desert	1.13	1.0	13	la	1.1347619047619049	0.9047619047619048	23
buy	1.11	1.0	11	c'est	1.11	1.0	11
oil	1.0975	0.9375	16	vie	1.1	1.0	10
happen	1.08	1.0	8	we'll	1.1	1.0	10
less	1.06	1.0	6	french	1.09	1.0	9
settle	1.06	1.0	6	croissant	1.08	1.0	8
beyond	1.05	1.0	5	cook	1.06	1.0	6
futuristic	1.05	1.0	5	ratatouille	1.06	1.0	6
innovation	1.05	1.0	5	fashion	1.0530769230769232	0.9230769230769231	13
skyscraper	1.05	1.0	5	bonjour	1.05	1.0	5
vegas	1.05	1.0	5	eyes	1.05	1.0	5
wealth	1.05	1.0	5	france	1.05	1.0	5
oasis	1.04	1.0	4	see	1.05	1.0	5

(word, score, posterior, occurrence) for Dubai (columns 0-3) and Paris (columns 4-7)



8	9	10	11	12	13	14	15
new	2.0347826086956524	0.9347826086956522	110	rio	1.42	1.0	42
york	1.8973417721518988	0.9873417721518988	91	football	1.34	1.0	34
dreams	1.7224390243902439	0.9024390243902438	82	brazil	1.22	1.0	22
where	1.6302127659574466	0.6702127659574467	96	carnival	1.2	1.0	20
made	1.603846153846154	0.9538461538461538	65	janeiro	1.2	1.0	20
are	1.5993023255813954	0.7093023255813954	89	de	1.1430434782608696	0.9130434782608696	23
concrete	1.54	1.0	54	samba	1.13	1.0	13
jungle	1.4274468085106384	0.9574468085106383	47	soccer	1.1290909090909091	0.9090909090909091	22
that	1.376969696969697	0.696969696969697	68	party	1.1211764705882352	0.9411764705882353	18
sleeps	1.3612195121951218	0.9512195121951218	41	jesus	1.11	1.0	11
never	1.2954545454545454	0.7454545454545454	55	life	1.1090909090909091	0.6590909090909091	45
apple	1.2	1.0	20	fun	1.08	1.0	8
big	1.1364516129032258	0.8064516129032258	33	carnivals	1.07	1.0	7
yeah	1.12	1.0	12	christ	1.07	1.0	7
true	1.1	1.0	10	vibrant	1.06	1.0	6
jungles	1.09	1.0	9	beaches	1.05	1.0	5
wall	1.09	1.0	9	drop	1.05	1.0	5
busy	1.07	1.0	7	joy	1.05	1.0	5
uh	1.07	1.0	7	olympic	1.05	1.0	5
walking	1.07	1.0	7	rhythm	1.05	1.0	5

(word, score, posterior, occurrence) for New York City (columns 8-11) and Rio de Janeiro (columns 12-15)