# CSC311 Machine Learning Challenge

Team Members: Aditya Kumar, Sharat Krishnan, Ishaan Jamdar, Jay Patel

UTORIDS: kumar168, krish506, jamdaris, pate1679

# Data

## Exploration and Data Representation (Data Split)

The data exploration aspect of the challenge was based on the lab sections and the processes explained in the lecture slides. The first step was to extrapolate the data into a format that can be viewed and edited (i.e. CSV → dictionary/list). We decided to stick with a dictionary for the cleaning process as we arranged our dictionary to have each key represent a column in the CSV. For our data to be used as input features in our models, we had to convert Q5 and Q6 into their own features to use the data inputted by the user. For Q6 specifically, this includes having each of "Skyscrapers", "Sport", "Art and Music", "Carnival", "Cuisine" and "Economic" as its own feature with its according value. For Q5, we had "Partner", "Friends", "Siblings" and "Co-worker" as separate features with 0 or 1 being its value (1 representing the field being selected and 0 representing the field is not selected).

The process for cleaning and formatting the quote field was slightly more complicated. Our initial idea was to ignore the quote field entirely as we wanted to use models that did not require the quotes (e.g. decision trees). However, we decided to use the quote field and we realized that it does provide valuable information and thus will improve our accuracy significantly. We use bag semantics to split each word in our vocabulary into a feature where the value for each word represents the frequency of that word occurring in the quote for a single data point. (i.e. 100 vocabulary words = 100 additional features) By accounting for the frequency, it gives the word a higher weight. Therefore, if a word is used more than once to represent a city, it means a higher correlation. We also ignore stop words as they do not carry significant information.

Problems that arose when using the data for the models were NaN values and incomplete fields. We discovered that columns Q1, Q2, Q3, Q4 and Q8 all had NaN values. Through debugging, we found that 9 rows had some sort of NaN value where 6 had 0s and NaN for every data field and 3 had only 1 NaN value for a specific column. Our thought process was that the 6 rows with no information should be deleted as replacing the NaN values with the mean of each column would give us no information. However, for the remaining 3 data points that already had a lot of partial information, it is more plausible to include the average for the NaN value column.

We split our data for model training purposes using the classic train, test, validation split which we commonly used in our labs throughout the term. We initially divided the data using train_test_split() from the sklearn library, which effectively split our data into a training set and test set. We chose a split size of 0.2, meaning 80% of the data was assigned to the training set whilst the other 20% was assigned to the test set.

It was crucial we did this, as the 20% test set would not impact the model, meaning it could be used to test how our models performed against unseen data. After this we split the remaining 80% training set again by 0.2 into a final training set and a validation set. The reason we required a validation set was to verify any instances of overfitting or underfitting occurring as we used models like k-nearest neighbors and decision trees where this would be a design concern. Also the fact that the sklearn train_test_split() was used allowed us to quickly test many different randomized splits for a final verification of accuracy.

## Feature Distributions

Feature Correlation (Empirical Evidence through Crosstab). The Following Figures represent the relationship between the features and the target as this is plotted numerically via tables. This was done using the pandas crosstab in order to get the cross tabulation of each feature. This compares the distributions of the features for each label, showcasing how the responses vary across different labels providing insights on identifying correlation nations between the features and cities. These distributions are key towards offering empirical evidence for analyzing correlation between specific features and determining to utilize them or not.

### Figure 1

This figure represents the distribution of the Label with Q1, showcasing the distributions across Dubai, New York City, Paris, and Rio de Janeiro as New York City is perceived to be the more popular city. Q1 here is the feature that represents popularity with 1 being least popular and 5 being most popular for each Label.

```
Crosstab for Label and Q1:
Q1              1.0  2.0  3.0  4.0  5.0
Label
Dubai             5    5   42  162  153
New York City     2    5    6   35  317
Paris             1    5   12   72  275
Rio de Janeiro    7   24  145  136   53
```

### Figure 2

This figure represents the distribution of the Label with Q2 (feature representing social media popularity with 1 least viral and 5 most viral). Dubai is perceived to be the most social media viral city as the feature fairly showcases the social media virality between all Labels

```
Crosstab for Label and Q2:
Q2              1.0  2.0  3.0  4.0  5.0
Label
Dubai            22   58  103  117   67
New York City     4   18   23   83  237
Paris            17   65  121   94   68
Rio de Janeiro   25  119  119   82   19
```

Figure 3

This figure represents the distribution of the Label with Q3 (feature representing architectural uniqueness with 1 being least unique and 5 being most unique). Paris is perceived to be the most architecturally unique city as the feature is fairly showcased its uniqueness between all Labels

```
Crosstab for Label and Q3:
Q3              1.0  2.0  3.0  4.0  5.0
Label
Dubai            14   25   51  124  153
New York City    19   58   98  112   78
Paris             8   12   48  119  178
Rio de Janeiro   17   86  140   84   37
```

Figure 4

This Figure represents the distribution of the Label with Q4 (feature representing Enthusiasm for Street parties with 1 being least and 5 being most). Rio De Janeiro is perceived to be the most Street party enthusiast as the feature showcases the Enthusiasm for street Parties fairly for each Label.

```
Crosstab for Label and Q4:
Q4              1.0  2.0  3.0  4.0  5.0
Label
Dubai            60  106   97   67   37
New York City    30   78   94   81   82
Paris            29   99  108   82   47
Rio de Janeiro    3    8   40   82  232
```

Figure 5

This Figure represents the distribution of Label with Q7 (feature representing Average temperature over the Month of January). With Paris being the Hottest (Greater

Distribution) and New York City is the Coldest in the month of January (Greater Distribution).

Crosstab for Label and Q7:

| Q7 | -1 | -10 | -12 | -15 | -2 | -3 | -4 | ... | 5 | 5.5 | 6 | 7 | 8 | 89 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Label | | | | | | | | ... | | | | | | | |
| Dubai | 0 | 0 | 0 | 0 | 0 | 0 | 1 | ... | 4 | 0 | 0 | 0 | 1 | 0 | 0 |
| New York City | 8 | 20 | 1 | 4 | 3 | 6 | 4 | ... | 62 | 0 | 5 | 6 | 6 | 1 | 0 |
| Paris | 0 | 6 | 0 | 1 | 5 | 1 | 1 | ... | 84 | 1 | 9 | 11 | 16 | 0 | 6 |
| Rio de Janeiro | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 4 | 0 | 0 | 0 | 1 | 0 | 1 |

[4 rows x 58 columns]

Figure 6

This figure represents the distribution of Label with Q8 (The feature representing the Languages Overheard). New York city has the most languages overhead as it has the greater distribution while Dubai has the least distribution for languages overheard.

Crosstab for Label and Q8:

| Q8 | 1.0 | 2.0 | 3.0 | 4.0 | ... | 87.0 | 100.0 | 200.0 | 800.0 |
|---|---|---|---|---|---|---|---|---|---|
| Label | | | | | ... | | | | |
| Dubai | 7 | 60 | 107 | 67 | ... | 0 | 0 | 0 | 0 |
| New York City | 63 | 56 | 56 | 34 | ... | 0 | 1 | 1 | 1 |
| Paris | 41 | 136 | 95 | 32 | ... | 1 | 1 | 0 | 0 |
| Rio de Janeiro | 28 | 120 | 144 | 37 | ... | 0 | 0 | 0 | 0 |

[4 rows x 25 columns]

Figure 7

This figure represents the distribution of Label with Q9 (The feature representing the Spotted Fashion Styles). New York city has the most fashion styles as it has the greatest distribution while Dubai has the least distribution for Fashion Styles.

Crosstab for Label and Q9:

| Q9 | 0 | 1 | 1,000 | 10 | 100 | 11 | 12 | ... | 78 | 8 | 80 | 81 | 9 | 90 | 99 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Label | | | | | | | | ... | | | | | | | |
| Dubai | 1 | 19 | 0 | 41 | 4 | 0 | 1 | ... | 0 | 11 | 0 | 0 | 2 | 1 | 0 |
| New York City | 0 | 6 | 0 | 64 | 5 | 0 | 2 | ... | 1 | 18 | 0 | 0 | 7 | 1 | 1 |
| Paris | 0 | 9 | 1 | 53 | 4 | 1 | 6 | ... | 0 | 21 | 1 | 1 | 2 | 0 | 0 |
| Rio de Janeiro | 0 | 16 | 0 | 26 | 1 | 2 | 0 | ... | 0 | 7 | 0 | 0 | 1 | 0 | 0 |

```
[4 rows x 41 columns]
```

Figure 8

This Figure represents the distribution with Label and Q5 (The feature representing Company for travel, which is 0 for no Partner, and 1 For Partner). Paris is the city to have Partner for travel as it has the greatest distribution while Dubai has the least.

```
Crosstab for Label and Q5_Partner:
Q5_Partner        0    1
Label
Dubai           171  196
New York City   134  233
Paris            51  316
Rio de Janeiro  153  214
```

Figure 9

Similar to the last figure, this now represents feature 5, but with No friends or Friends. Rio De Janeiro can be seen as the city to travel with the friends the most and Dubai Paris being the least.

```
Crosstab for Label and Q5_Friends:
Q5_Friends        0    1
Label
Dubai            85  282
New York City    78  289
Paris           183  184
Rio de Janeiro   62  305
```

Figure 10

Similar to the previous figures, this represents the distribution with Label and the Q5 Feature with just siblings (No for Sibling or Yes for Siblings). New York City has the greatest distribution to travel with Siblings, while Rio de Janeiro has the least.

```
Crosstab for Label and Q5_Siblings:
Q5_Siblings       0    1
```

```
Label
Dubai           240  127
New York City   161  206
Paris           248  119
Rio de Janeiro  250  117
```

Figure 11

Similar to the previous figures, this represents the distribution with Label and the Q5 Feature with just CoWorkers, (No for Coworkers, Yes For Coworkers). New York City has the greatest distribution for Coworkers, while Rio De Janeiro has the least.

```
Crosstab for Label and Q5_Co-worker:
Q5_Co-worker      0    1
Label
Dubai           279   88
New York City   155  212
Paris           325   42
Rio de Janeiro  335   32
```

Figure 12

This Figure represents the distribution of Label and Q6 (The feature representing Skyscrapers most relatable to city, 0 for Not relatable, 6 for most). Skyscrapers are seen to be most relatable with Dubai and least relatable with Rio de Janeiro.

```
Crosstab for Label and Q6_Skyscrapers:
Q6_Skyscrapers  0    1   2   3   4   5    6
Label
Dubai           0   16   1   4  10  53  283
New York City   2   16  13   9  29  90  208
Paris           2  113  94  50  48  27   33
Rio de Janeiro  2  218  80  36  15   3   13
```

Figure 13

This Figure represents the distribution between Label and Q6 (The feature representing Sports most relatable to city, 0 for Not relatable, 6 for most). Sports are seen to be most relatable with Rio de Janeiro and least with Dubai due to the distributions.

```
Crosstab for Label and Q6_Sport:
```

| Q6_Sport | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| Label | | | | | | | |
| Dubai | 0 | 107 | 95 | 88 | 45 | 22 | 10 |
| New York City | 2 | 93 | 81 | 77 | 65 | 31 | 18 |
| Paris | 2 | 82 | 71 | 84 | 66 | 41 | 21 |
| Rio de Janeiro | 2 | 6 | 18 | 20 | 49 | 88 | 184 |

Figure 14

This Figure represents the distribution between Label and Q6 (The feature representing Art and Music most relatable to city, 0 for Not relatable, 6 for most). Art and Music are seen to be most relatable with Paris and least with Dubai due to the distributions.

Crosstab for Label and Q6_Art and Music:

| Q6_Art and Music | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| Label | | | | | | | |
| Dubai | 0 | 65 | 119 | 116 | 46 | 11 | 10 |
| New York City | 2 | 13 | 74 | 80 | 89 | 59 | 50 |
| Paris | 2 | 12 | 9 | 19 | 26 | 94 | 205 |
| Rio de Janeiro | 2 | 5 | 13 | 42 | 111 | 133 | 61 |

Figure 15

This Figure represents the distribution of Label and Q6 (The feature representing Carnivals most relatable to city, 0 for Not relatable, 6 for most). Carnivals are seen to be most relatable with Rio de Janeiro and least relatable with New York City due to the distributions.

Crosstab for Label and Q6_Carnival:

| Q6_Carnival | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| Label | | | | | | | |
| Dubai | 0 | 108 | 90 | 88 | 51 | 16 | 14 |
| New York City | 2 | 132 | 94 | 56 | 45 | 19 | 19 |
| Paris | 2 | 76 | 76 | 84 | 83 | 35 | 11 |
| Rio de Janeiro | 2 | 9 | 19 | 27 | 64 | 84 | 162 |

Figure 16

This Figure represents the distribution of Label and Q6 (The feature representing Cuisines most relatable to city, 0 for Not relatable, 6 for most). Cuisines are seen to be most relatable with Paris and least relatable with Dubai due to the distributions.

Crosstab for Label and Q6_Cuisine:

| Q6_Cuisine | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| Label | | | | | | | |

```
Dubai           0  36  51   86  125   44   25
New York City   3  28  57  108  102   44   25
Paris           2  11  17   24   53  126  134
Rio de Janeiro  2  20  39  166   83   38   19
```

Figure 17

This Figure represents the distribution of Label and Q6 (The feature representing Economy most relatable to city, 0 for Not relatable, 6 for most). Economy is seen to be most relatable with New York City and least relatable with Rio de Janeiro due to the distributions.

```
Crosstab for Label and Q6_Economic:
Q6_Economic     0   1    2   3    4    5    6
Label
Dubai           0   9   18  19   37  163  121
New York City   2  17   15  19   25  105  184
Paris           2  33   69  97  104   47   15
Rio de Janeiro  2  67  178  73   30   11    6
```

Figure 18

This Figure represents the distribution between Label and the ID. Since this distribution has no real relevance, the Id is not required as part of the features.

```
id              5978    14231   24137   40536   ...  633733  633746  634513
727099
Label                                           ...
Dubai                1       1       1       1  ...       1       1       1
1
New York City        1       1       1       1  ...       1       1       1
1
Paris                1       1       1       1  ...       1       1       1
1
Rio de Janeiro       1       1       1       1  ...       1       1       1
1

[4 rows x 367 columns]
```

Figure 19 & 20

This figure shows the distribution between Label and Q10 (Which is the feature representing the choices of quotes that comes to mind of each city, but is now broken down into a bag of words). Thus each word becomes a feature as part of Q10. The following are sample distributions of two words, tennis and ice showing the frequency or occurrence of that relative to Label/City.

```
Crosstab for Label and tennis:
tennis            0  1
Label
Dubai           366  1
New York City   367  0
Paris           367  0
Rio de Janeiro  367  0

Crosstab for Label and ice:
ice               0  1
Label
Dubai           367  0
New York City   366  1
Paris           367  0
Rio de Janeiro  367  0
```

## Determining Input Features

In determining the input features for our analysis, we followed a systematic approach that leveraged both logic and empirical evidence to determine which input features to include. This involved identifying the relevant attributes identifying and collectively capturing the city under the questions. Based on the empirical evidence derived from the dataset, we observed significant variations in the distributions of the responses across the identified features (the questions) suggesting its relevance in characterizing the unique attributes of the city. Each of the features from Q1 to Q9 had variations in distributions which helped portray the essence of the city using unique attributes. Additionally, using the bag of words provided additional insights into the connections with each city enhancing the relevance of the feature. Thus as a result, we translated each relevant attribute of the city as a feature to ensure no important aspect of a city was being overlooked. Splitting each word in our vocabulary into a feature in order to quantify the frequency of occurrence of each word helped to form correlation between each word in the quote to a city. Furthermore, when characterizing a city, it's important to include attributes relevant to the city such as street parties, popularity, social media virality, type of travel place, Relatable Factors in the City, City enthusiasm, City Architecture, temperature, fashion, language and

quotes are all important factors to consider logically to differentiate the cities. Additionally, given by the Labels that had the highest and lowest distributions varied, backs the relevance of including the features Q1-Q10 (10 being broken down into more features) while accounting for outliers to ensure a city is characterized accurately based on empirical data. Overall, our determination of input features was guided through the logical and comprehensive understanding of the nature of cities (how these can vary between cities), including empirical evidence to ensure an insightful analysis for choosing input features.

# Model

We initially trained the data on each model using sklearn before implementing pred.py.

K Nearest Neighbour Model

We implemented a k Nearest Neighbour Classifier Model which would classify the given data of the features to one of the four cities. Prior to using the training, validation, and test sets, the data is normalized using the mean and standard deviation. Normalization is done because without scaling, the larger scales can dominate in the distance calculation leading to biased results due to higher numerical values. While representing the data, we ensure every feature is a numerical value which allows for easy use of the kNN model. We use uniform weights so all points in each neighborhood are weighted equally. There is also the option to give closer points a higher weight, however we need to be able to implement this without sklearn library as well so that was taken into consideration.

Decision Tree

We implemented a Decision Tree Classifier Model which will classify the given data of the features to one of the four cities. The data is not normalized because the decision tree depends on splitting based on feature values and the actual values do not affect the model.

Multi-Layer Perceptron

We implemented a Multi-Layer Perceptron Classifier model which will classify the given data of the features to one of the four cities. We set the maximum iterations of the classifier to be 5000 so that the model could be as close to convergence as possible. While training and testing the model we used the default solver for sklearn, adam which is a stochastic gradient based optimizer. We decided to choose the default solver because we wanted to gauge how effective a MLP model is in this scenario. We took into consideration the fact that a MLP model made with

python without the sklearn library would be a much simpler model which may not be the most optimal.

## Model Choice and Hyperparameters

The evaluation metrics are comparable because the training, validation and test sets are the same for all the models. We run all the models in one program so that they can use the training, validation, and testing set. Thus, we can compare the validation accuracy among the variations of one model to pick the best combination of hyperparameters. Similarly, we can compare the testing accuracy between the models to pick the model for the data. Please see attached the testingModels.py file to see how we set up each model and got its training, validation, and testing accuracies.

The evaluation metric used to evaluate the three models with tuned hyperparameters was the testing accuracy which was the determinant to decide which model is the best. This is used as the evaluation metric because we care most about the accuracy of the model, as given the context, we will be judged on how our model will perform against an unknown test set. The other accuracy metrics like validation accuracy and training accuracy do not apply here as they will inflate the accuracy of the model since validation accuracy is used to tune the hyperparameters and training accuracy is used to train the model. The confusion matrix evaluation metric does not apply here as an incorrect prediction from our model will not have any impact. Thus, the false positive rate and false negative rate is not a significant factor to determine which model is better.

For the kNN model, we tuned the hyperparameters, k (the number of nearest neighbors) and the metric to measure distance which is either euclidean or cosine similarity. We only look at values of k from 1 to 10 because it is known that large values of k can lead to underfitting. After, looping through all the combination and comparing the validation accuracies, the best values were:

k = 7
metric:‘euclidean’
validation accuracy: 0.811965811965812

For the MLP model, we tuned the hyperparameters, activation function, number of hidden layers, and number of neurons. These were the hyperparameters because there were multiple possible values for these parameters that can affect how accurate the model is. The options for activation functions are identity, logistic, tanh, and relu. The options we kept for hidden layer were 1,2, and 4 because it would be difficult to implement a large number of hidden layers in python (without

sklearn). The options we kept for the number of neurons were 5, 20, 50, and 100. After looping through all the combinations and comparing the validation accuracies, the best values were:

Activation function: ‘tanh’
Hidden layer: 2
Number of neurons: 100
Validation accuracy: 0.8717948717948718

For the Decision Tree model, we tuned the hyperparameters, criterions to measure quality of split, maximum depth of the tree, and minimum samples required to split. These were the hyperparameters because there were multiple possible values for these parameters that can affect how accurate the model is. The options for criterion are gini, entropy, and log-loss. The options for maximum depth of the tree are 1, 5, 10, 15, 20, 25, 30, 50, 100. We capped it at 100 because it would be difficult to implement an extremely big tree. The options for minimum samples required to split are 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024. We capped this at 1024 because there are only 1469 samples. After looping through all the combinations and comparing the validation accuracies, the best values were:

Criterion: ‘entropy’
Maximum depth of three: 5
Minimum sample split: 2

In order to find the best model out of these three, we compared the testing accuracies of these 3 models with tuned hyperparameters. The testing accuracies were:

kNN model:  0.797945205479452
MLP model: 0.8082191780821918
Decision Tree model:  0.8527397260273972

Based on the testing accuracies, Decision Tree Classifier is the best model.

Please see attached the testingModels.py file to see how we set up each model and got its training, validation, and testing accuracies.

In order to implement this model in python for pred.py we need to visualize the Decision Tree and find out its conditions. We visualized the Decision Tree by using Google Colab to display the decision tree and its classifications. Please see attached Google Colab file for how the decision tree looks like and the code behind it.

Then, we were able to implement our pred.py through a series of if-else statements replicating the decision tree. In pred.py, we also organize the data from the csv file and put it in the form that we can use our model on (as explained in the Data section of the report).

## Prediction

Prediction

We predict that there will be a testing accuracy of 0.8143877551020406.

We obtained this prediction by generating 100 random splits of our data using the train_test_split function that uses a random seed every time. For each split, we trained our Decision Tree model (with the tuned hyperparameters) on the training set and then tested it with our testing set. We accumulated the testing accuracy for every split and then took the average of all the testing accuracies (average of 100 samples). The code for this is at the bottom of the attached Google Colab file.

## Workload Distribution

- Jay Patel
    - Handling formatting and input feature conversion of the data to be used by the model.
    - Responsible for handling incomplete/outlier data.
- Sharat Krishnan
    - Handling formatting and input feature conversion of the data to be used by the model.
    - Handled empirical evidence for distributions of features to determine which features to include based on logic and empirical evidence.
- Aditya Kumar
    - Worked on training the model and analyzing model accuracy
    - Explored other model families and evaluated them.
- Ishaan Jamdar
    - Worked on training, building and testing the final model for predictions.
    - Explored other model families and evaluated them.