# 

# 

# **Machine Learning Project Report**

CSC311H5

April 8, 2024

**Moeez Omair, Adam Tran, Kamil Chaudhry, Daniel Iseoluwa**

# 

# **Table of Contents:**

* **Introduction 2**
* **Data 2**
* **Model 6**
* **Hyperparameters and Model Choice 6**
* **Prediction 7**
* **Workload Distribution X**

# **Introduction**

The world is vast and full of fascinating places, each carrying a unique flavour unlike any other. Perhaps it is possible to examine the uniqueness of these cities through potentially defining characteristics for each of them. To do this, we aimed to train a machine learning model which may help predict the city that is being described by the data provided.

The first step in this process was to conduct a comprehensive survey, designed to determine how people perceive these cities when compared to each other. The survey consisted of a variety of questions such as:

* Rating on a scale of 1 to 5
* Multiple choice
* Short answer

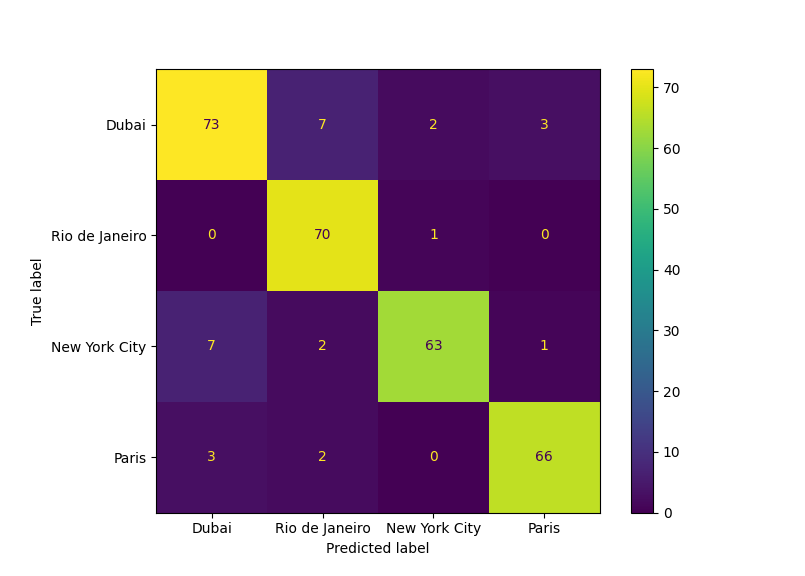
After collecting information from our peers, we split the task amongst ourselves to determine the best model to represent our data. We have tried to use the k-NN model, decision tree model, and neural network, hoping one of them can stand out from the others and we focus our efforts on refining that particular model. The results from our expedition may be found in the following sections.

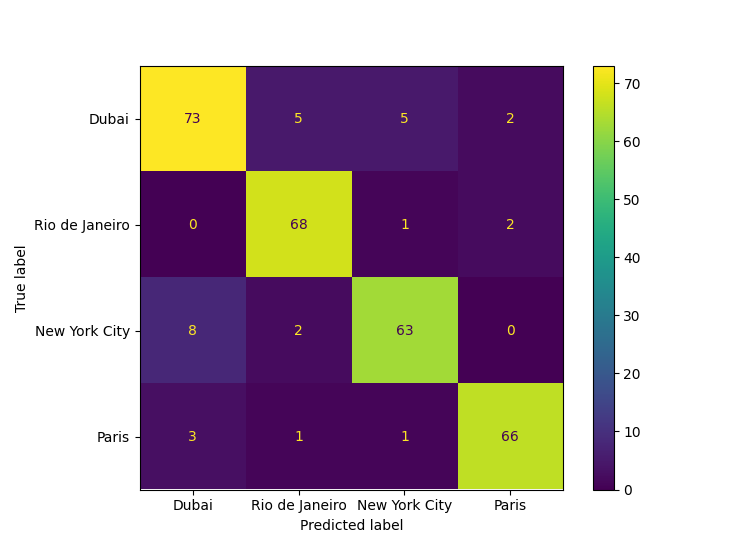
# **Data Exploration**

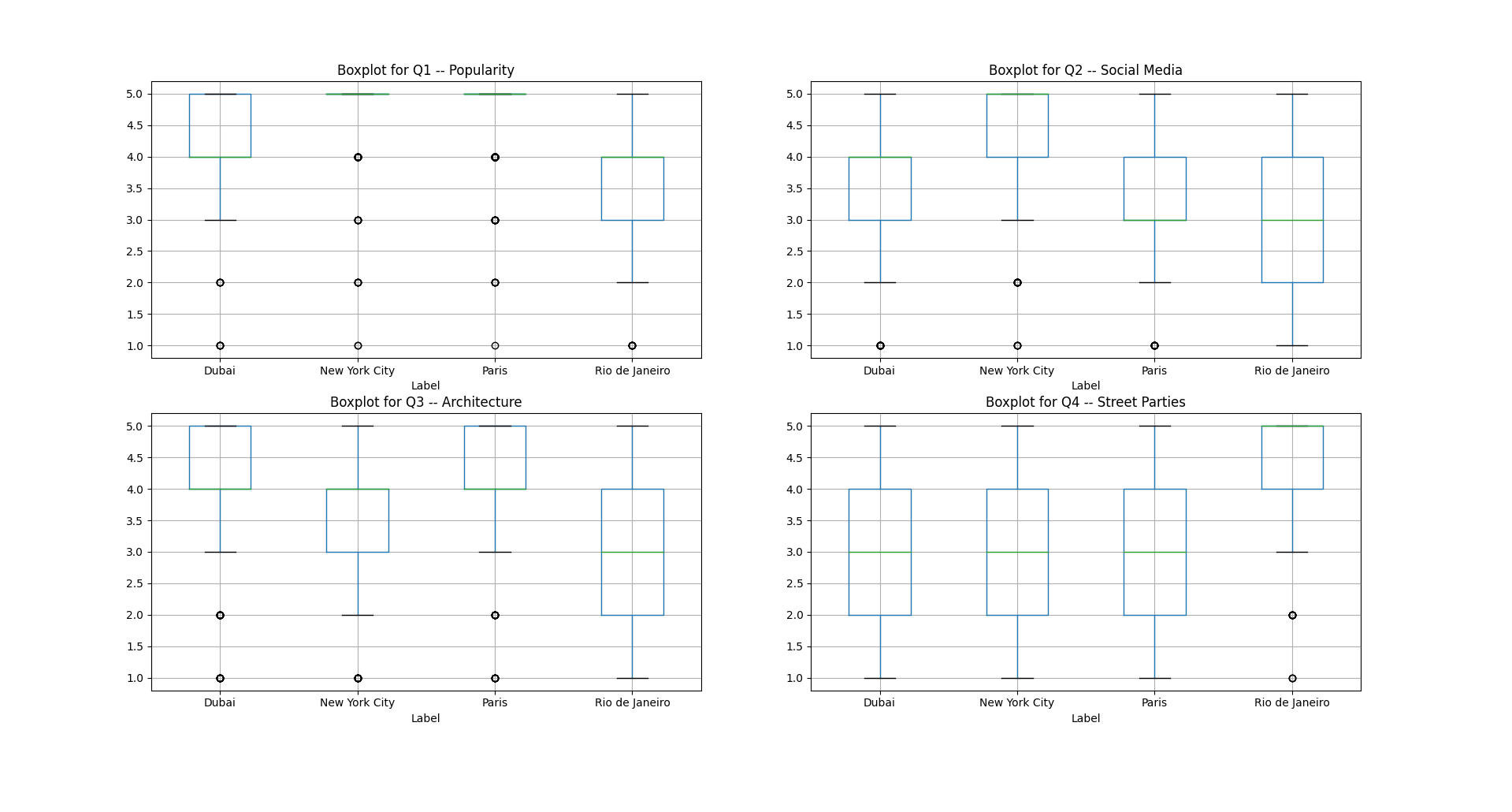
The data was split into 80% training and 20% validation. We represented the data with the categorical selection (Q5) being one-hot encoded and the quote (Q10) being encoded using a \bag-of-words model. The remaining selections were encoded using the numerical value. The vocabulary in the bag of words model is generated from the quotes in the training set and any words of length 2 or less and simple words such as “the'' or “and'' are removed from the vocabulary as these words do not add any additional information for the category of the data. This reduced the dimension of the data from 1456 to 1404 which reduces the noise and redundant features in the data.

We also tried representing the questions that were selections from 1-5 (Q1-Q4, Q6) as a one-hot encoded vector instead of just using the numeric value. This did not have a noticeable change in accuracy but caused the model to take slightly longer to converge (on average about 5 epochs more), which is likely due to the larger feature space.

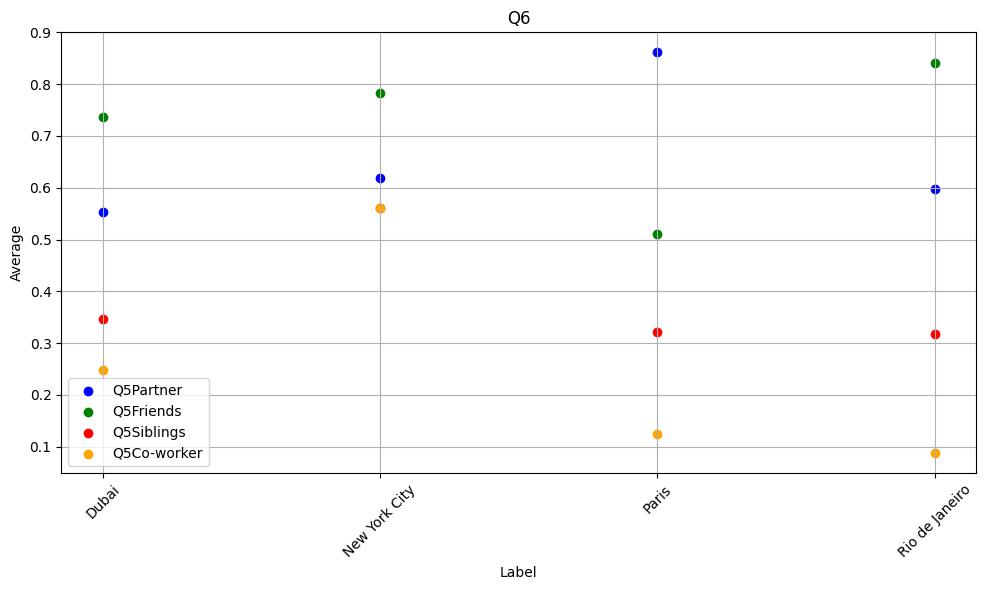
Below are the confusion matrices for the models trained on the two data representations which are evaluated on the validation set:



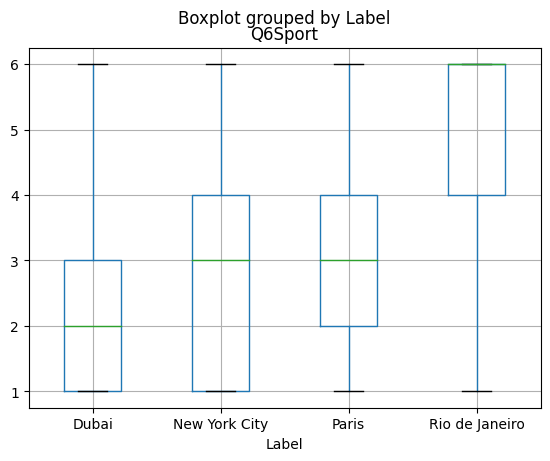
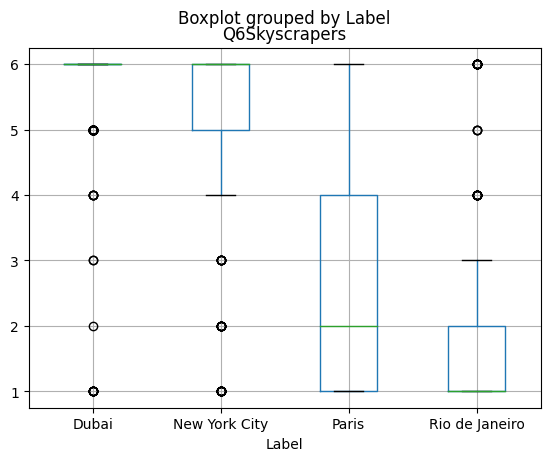


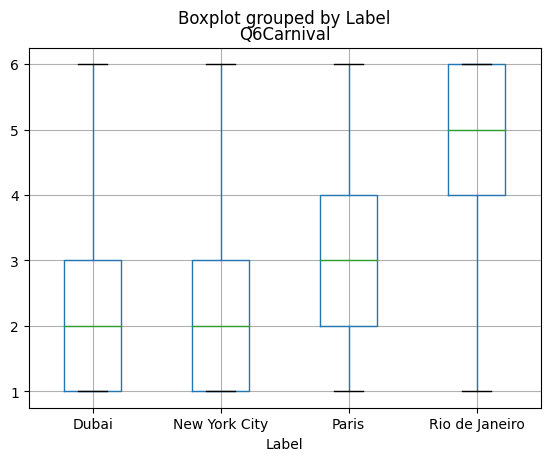
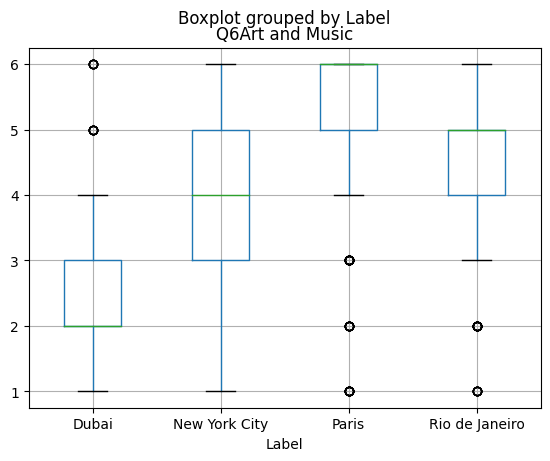


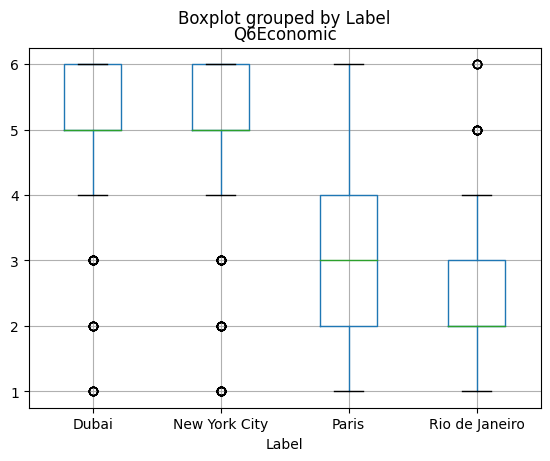
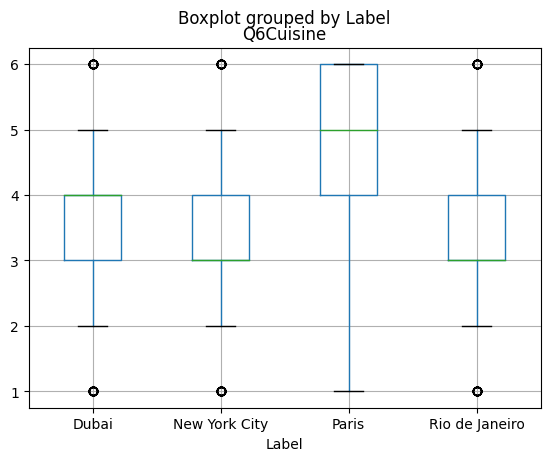
For the distribution of the data, Q1-Q4 is a value on a scale from 1-5. We see from the box plot of Q1 that New York and Paris are the most popular, followed by Dubai and Rio de Janeiro. For Q2, New York is the most popular on social media. For Q3, Dubai and Paris score the highest for architecture, with Rio de Janeiro being less known for architecture. For Q4, Rio scores the highest for street parties. It makes sense to represent these as numerical features rather than one-hot vectors because there is an ordered relationship between the values.



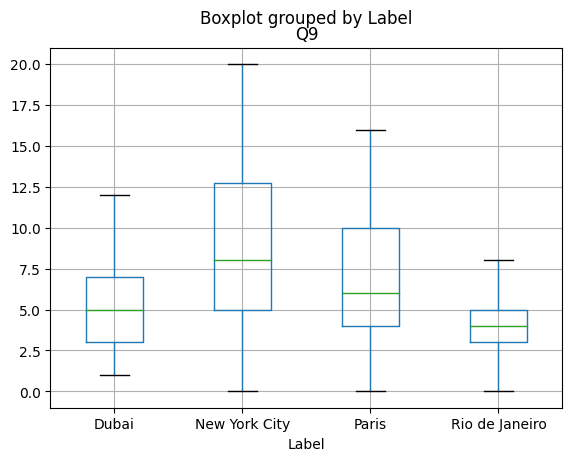
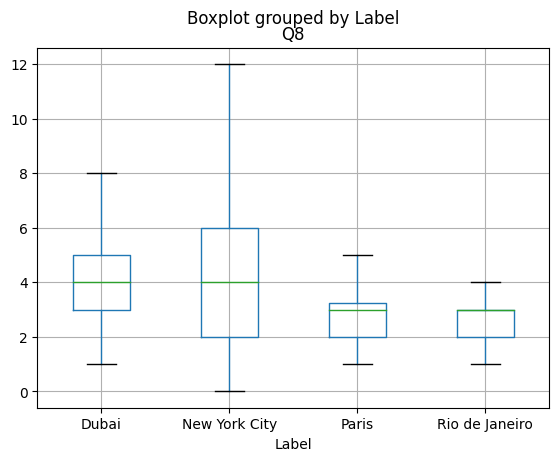
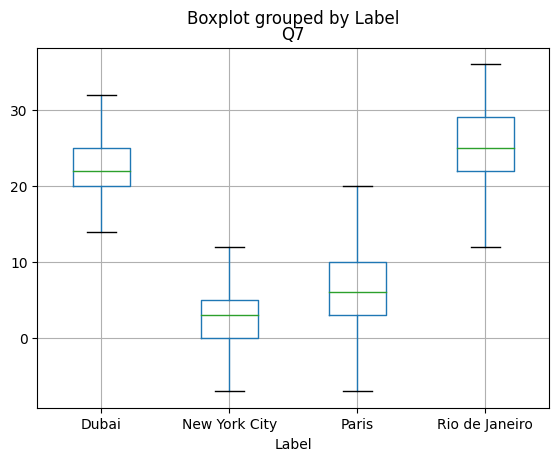
For Q5, the scatter plot above indicates that most people would accompany their partner to Paris, their friends to Rio de Janeiro and Dubai, and their co-workers to New York City. The scatter plot is the average number of selections per city as each of these 4 features are stored as indicator columns. It makes sense to put these features together as they could provide information about each other, for example most people would accompany their friends to Dubai and New York city, but not many people would accompany their co-workers to Dubai as compared to New York.







For Q6, we split the different categories and kept their numerical values as is. Q6 is quite informative compared to the other questions. Dubai and New York seem to have high values for the skyscraper section, Rio de Janeiro has high values in the sports and carnivals section, Paris has high values in the cuisine and art sections, and New york has high values in the economic section. Dubai and New York seem to have similar values for most of these sections, but nevertheless there is still a lot of useful information from these features.



For Q7 - Q9, the range of values are not restricted so we are ignoring outliers in our box plots. The box plot of Q7 indicates that Dubai and Rio de Janeiro have the highest temperature in January, while New York has the lowest temperature. For Q8, they are all roughly centered, with New York and Dubai spanning a bigger range for the number of languages one might hear. For Q9, New York and Paris seem to have a larger number of fashion styles that one may encounter followed by Dubai and Rio de Janeiro. Q8 and Q9 are not particularly helpful as the cities have very similar values for them making it hard to discern between them using this piece of data.

# **Model**

As decided upon by our group, the models we are evaluating are k-NN, decision tree and a neural network.

For the neural network we are using one hidden layer with 200 hidden units. We are only using one hidden layer and a low number of hidden units to prevent overfitting to the training set. There are 4 output units and the softmax function is applied to the output to interpret each output unit as a probability.

For the k-NN model, we are using Euclidean distance to compare distances between features. During the process, we found that a k value of 5 presents the highest validation accuracy when compared to other values.

As for the decision tree, our branches split the data based on the max information gain using an entropy loss function until we reach max depth or until every entry in the resulting split is of one category. The model builds the tree recursively during training and each node automatically searches for the split that results in the highest information gain. To predict new data points, we just traverse the tree according to the data and classify it based on the majority label at our leaf node (if more than one appears).

# **Hyperparameters and Model Choice**

We maintain a consistent training and test set across all 3 models, and are comparing them based on the validation accuracy after tuning the hyperparameters. For evaluation we are using accuracy which is the number correct out of total predictions and a confusion matrix to see where the model is making incorrect predictions, and then we chose the model with the least incorrect predictions. This was how we decided to use the neural network as the final model as it has the highest accuracy across the training and validation set while not overfitting to the training set.

TODO: clear description of what the neural net looks like in pred.py

For the k-NN model, the hyperparameter we are working with is the value k. This value represents the number of neighbors to consider when making predictions with our k-NN algorithm. Initially, our default value for k is 3, however after setting up the initial design of this model, we created a loop to test out other values of k between 1 and 9 as well. Eventually, we discovered the sweet spot for our k value to be when k = 5. The r Intuitively, this value makes sense as it provides a balance between accuracy based upon the training data provided as well as eliminating noise from outliers.

There were two hyperparameters to work with in the decision tree mode, being the information gain criteria, and the max depth. We compared the accuracies of different combinations with sklearn’s decision tree model in order to get a target accuracy. Using gini resulted in the accuracies being slightly worse than entropy across both models, so we decided to use entropy instead. As for the max depth, we again compared it to sklearn’s model and experimented with different values prioritizing accuracy and training runtime. The optimal balance seemed to a max depth of 7 for both models, with both of their validation accuracies hovering around 78%.

Based upon testing we have currently done on our code, these are the best accuracies we have achieved on our models:

| Model | Training Accuracy | Validation Accuracy |
| --- | --- | --- |
| Neural Network | 91.52% | 90.6% |
| k-NN (k = 5) | 85.79% | 82.99% |
| Decision Tree | 93.32% | 78% |

# **Prediction**

TODO: how well will the model perform (revise what I wrote)

Since we settled upon the neural network, we predict that the final test will similarly reflect the accuracy found in the previous section. This means we predict that our neural network will predict the label with an accuracy of approximately 90%. This is a tremendous final outcome as it proves that our model can generalize very well to unseen data.

We are confident this would be the outcome because we ensured a balance in training our data, resulting in the highest validation accuracy after tuning our hyperparameters while not overfitting to the training set.

# **Workload Distribution**

Moeez led the collaboration effort for the group. He was also responsible for creating the k-NN model and worked on the final report, specifically the ‘introduction’, ‘data’, and ‘model and hyperparameters’ of the report for the k-NN model as well as the ‘prediction’ section hypothesizing the final outcome

Adam was responsible for creating the neural network model, which was the model we eventually chose. He also contributed to the overall presentation and outline of the report, more specifically to the ‘data exploration’, and ‘model and hyperparameters’ sections for the neural network.

Kamil designed the decision tree model. He also contributed to the ‘data exploration’ section of the report, more specifically the data distributions and figures, and ‘model’, and ‘hyperparameters’ section of the report for the decision tree. At the very end, he edited all of the sections for greater flow.