CS4563 - Introduction to Machine Learning

Final Project Report

Written by Qilei Cai and Tasnim Nehal

Introduction

The dataset, obtained from kaggle.com, contains 48895 Airbnb listings located in New York City. The dataset can be found in the file *AB_NYC_2019.csv*. The features provided for each listing include the neighborhood group (Manhattan, Brooklyn, etc.), latitude, longitude, price and more. Other features of statistical significance include the minimum number of nights, the total number of reviews received by the host, the average number of reviews received per month, the host's total number of listings, and the number of days in a year for which the accommodation is available.

Many of the features provided are relevant to the pricing of an accommodation, such as the room type and the neighborhood. The goal of the project is to experiment with multiple machine learning models and train them to predict the price range of a particular accommodation, given a list of features.

Unsupervised Analysis

Unsupervised analysis was performed on the dataset by creating a k-means model with five clusters. The model was created using sklearn. The program for unsupervised analysis can be found in the file *unsupervised_analysis.py*.

Some of the features in the dataset were deemed irrelevant to the task and excluded from the training of the model. The following features were preserved: latitude, longtiude, price, minimum number of nights, total number of reviews, number of reviews per month, number of listings belonging to the host, availability in a calendar year, room type, and neighborhood group. Since room type and neighborhood group were written as strings in the dataset, the program converts each type into a number. For example, the room type "Entire home/apt" was converted to the number 1, and Brooklyn was converted to the number 2. The model was fit with these features of the dataset. Clustering predictions were made using a method of sklearn.

The clustering predictions were graphed using matplotlib. Shown below is a graph of longitude versus latitude.

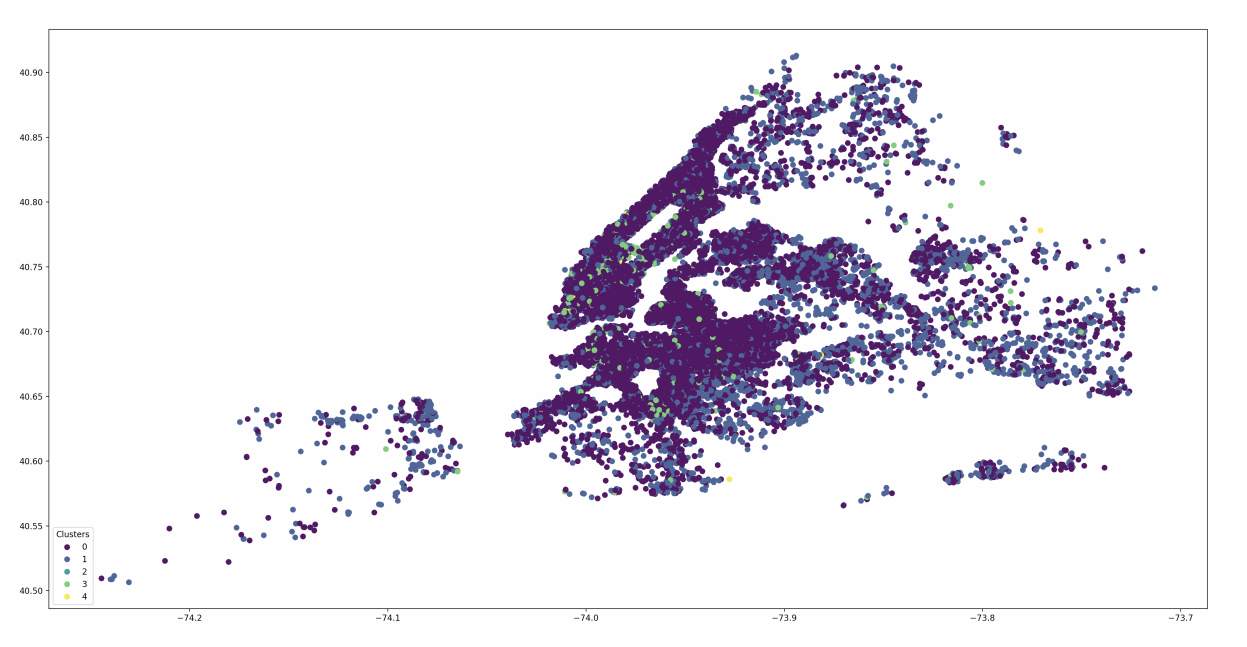


Figure 1: Longitude vs. Latitude (Original)

The x-axis is longitude, and the y-axis is latitude. This way, the data points are rendered as they would show on a map. Comparing the distribution of the points to a map of NYC, the five boroughs are identified in the figure below. You can also tell that the blank space in densely-dotted Manhattan is Central Park.

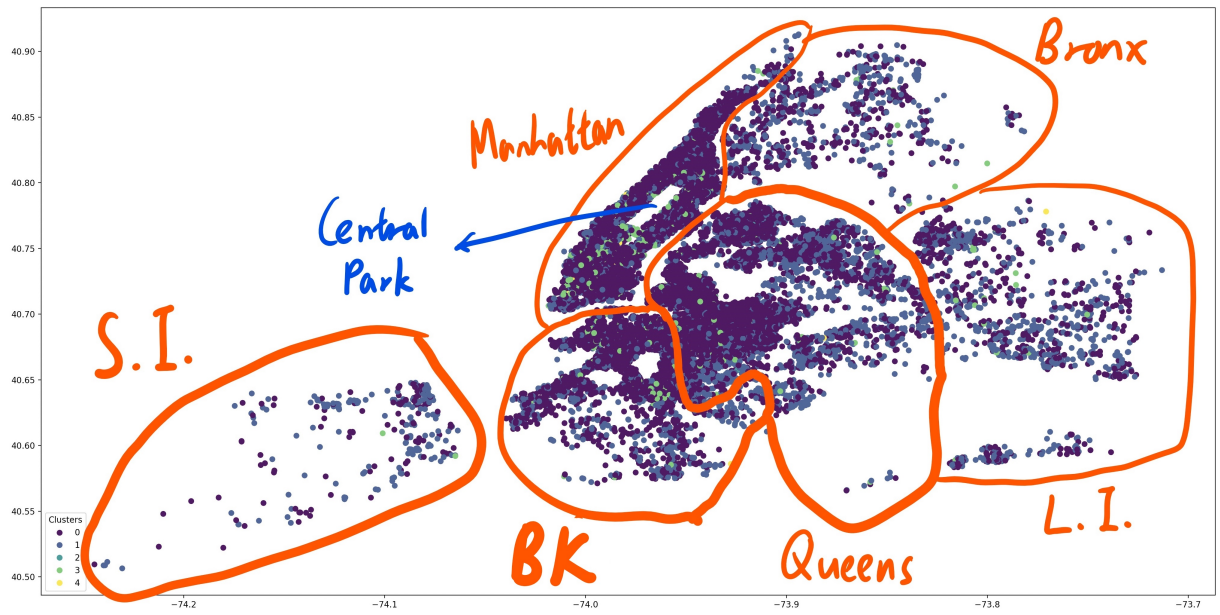


Figure 2: Longitude vs. Latitude (Annotated)

These graphs give a good idea of where the listings are most densely located. The entire Manhattan is covered by data points, except for Central Park; Brooklyn and Queens lag behind, but the points are dense as well. The Bronx, Staten Island and Long Island are on the bottom rung here.

The data points are rendered in five colors, one for each of the five clusters that the k-means model identified. At first glance, there is no clear correlation between geography and the clustering. The purple dots and blue dots are the most common and are strewn all over the place. Despite that, it is worth noting that most of the green dots are located in Manhattan. Specifically, many of them circle Central Park, and the rest are mostly in Midtown and Downtown. At this point, not enough analysis has been done to offer an explanation. More clues should come to light as the other features are analyzed.

Next, a graph of the minimum number of nights (the minimum required for a booking) versus the total number of reviews is made. The x-axis is the number of nights, and the y-axis is the number of reviews.

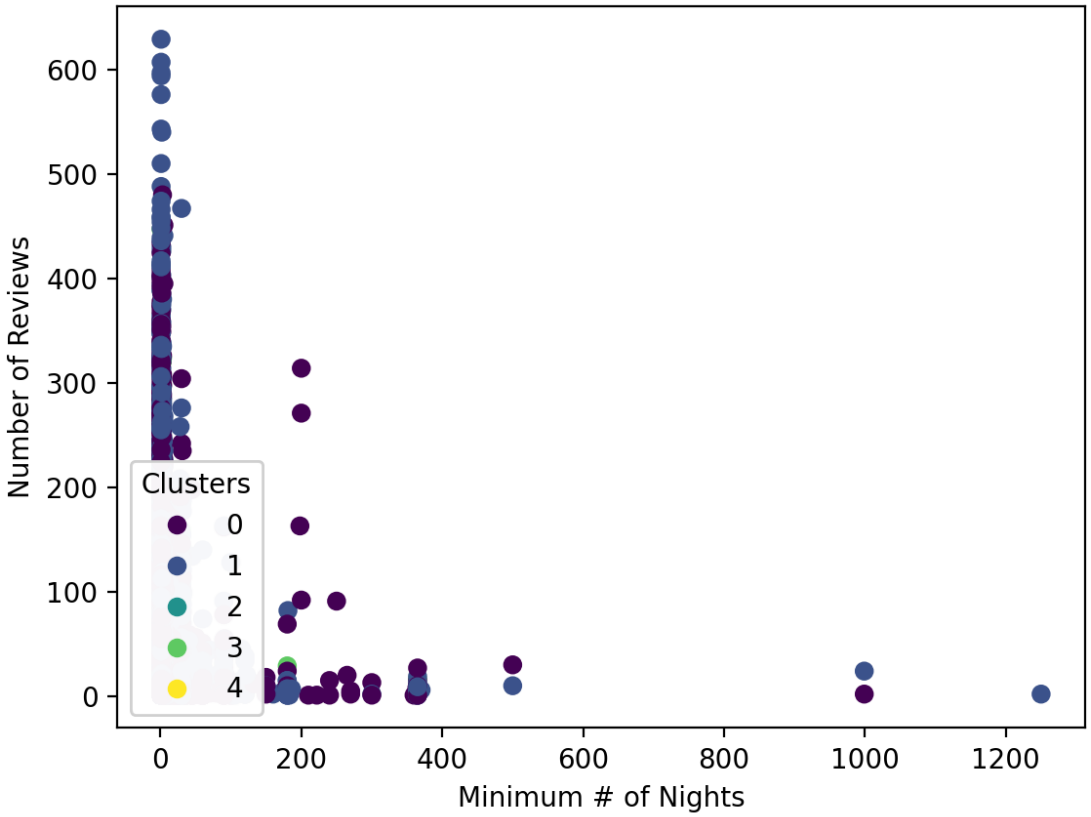


Figure 3: Minimum Number of Nights vs. Number of Reviews

The general trend is that the lower the number of nights, the higher the number of reviews. This relationship is intuitive—more people stay at accommodations where they can book fewer nights, so these places receive more reviews. However, there is no clear correlation between the two variables and the clustering. Purple dots and blue dots are present across the entire spectra of reviews and nights, whereas the dots of the other colors can barely be seen. Therefore, neither of the two variables had a heavy weight in how the model made the clustering predictions.

Shown below is a graph of the minimum number of rights versus room type. There are three room types in the dataset: entire home/apartment, private room and shared room. During pre-processing, the three labels were converted to 1, 2 and 3 respectively. The x-axis is the minimum number of nights, and the y-axis is the room type, which is labeled either 1, 2 or 3.

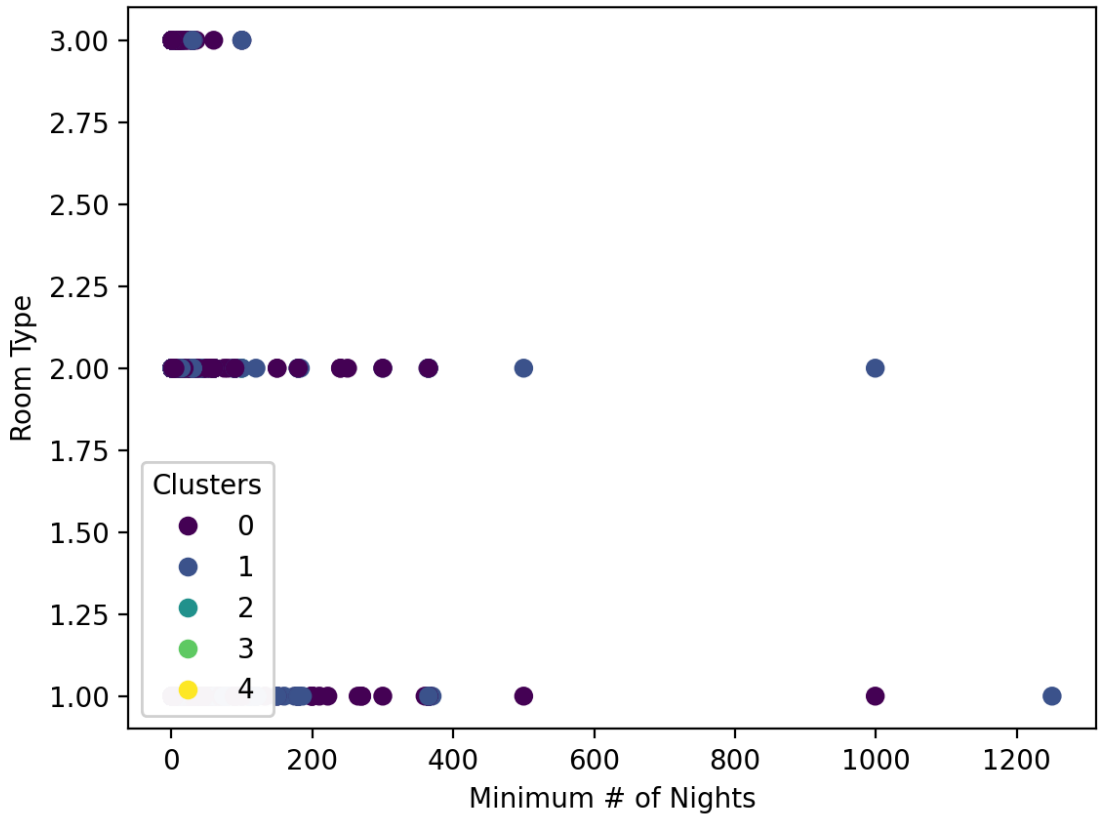


Figure 4: Minimum Number of Nights vs. Room Type

The graph shows that shared rooms (labeled 3) require the least number of nights for a booking, whereas the distribution for the other two room types are similar. Since shared

rooms are targeted toward short-term travelers, it is expected that they require fewer nights per booking. A clear pattern of clustering is still missing—the purple dots and blue dots are present across the spectrum of nights and in the three room types, whereas the dots of the other colors can barely be seen. This indicates that neither of the two variables had a heavy weight in how the model made the predictions.

After a few unsuccessful attempts at looking for a pattern of clustering, the attention was turned toward pricing. A graph of the number of reviews per month versus price was made. The x-axis is the number of reviews, and the y-axis is the price in dollars.

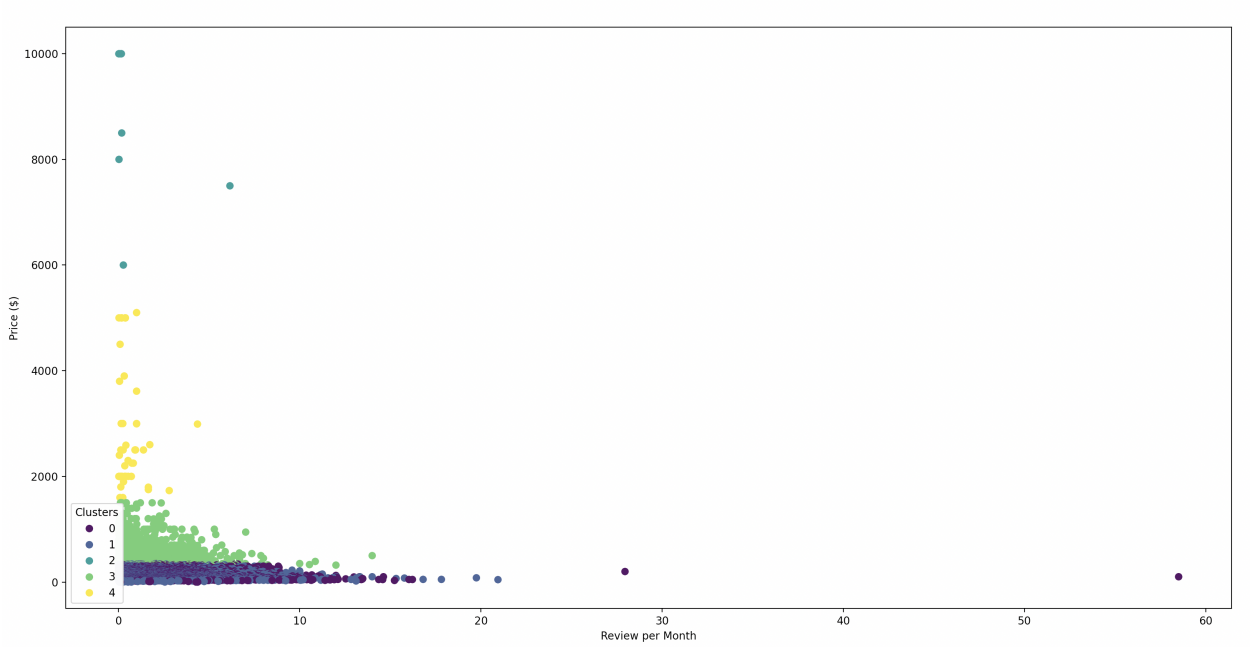


Figure 5: Number of Reviews per Month vs. Price ($)

This attempt finally yielded a clue as to how the clusters were made. The graph clearly shows that dots of a particular color are all clustered within a certain price range. The cyan dots, which refer to Cluster 2, are at the top of the pile in terms of pricing, followed by yellow dots, which refer to Cluster 4. The green dots, which refer to Cluster 3, are ranked in the middle. The purple and blue dots, which refer to Cluster 0 and 1 respectively, overlap each other in the lowest price range. On the other hand, the the graph does not illustrate any correlation between reviews and the clustering, since there are dots of multiple types in the range of 0-10 reviews per month. This pattern indicates that price had a heavy weight in how the model made the clustering predictions.

Another graph was made to analyze the relationship between pricing and room type (labeled 1, 2 or 3 like above). The graph again shows a clear pattern of clustering based on price range. On the other hand, there is no clear correlation between room type and the

clustering, since dots of multiple types are present for each room type.

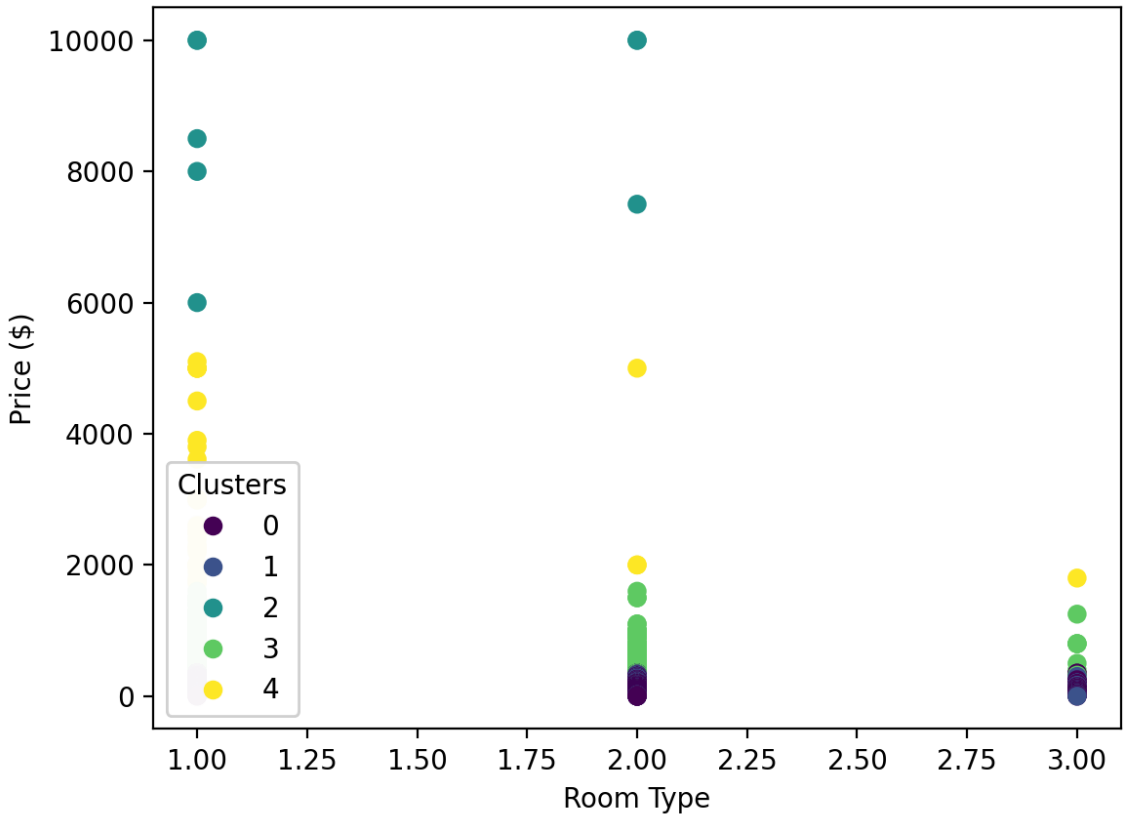


Figure 6: Room Type vs. Price ($)

More analysis was done on the relationship between pricing and the other features. Shown below is a graph of neighborhood group versus price. There are five neighborhood groups in the dataset: Manhattan, Brooklyn, the Bronx, Staten Island and Queens. The five labels were converted to 1, 2, 3, 4 and 5 respectively during pre-processing. The graph illustrates the same pattern as the previous two graphs. In addition, the graph illustrates that the more expensive listings are mostly located in Manhattan and Brooklyn, whereas the other neighborhoods have fewer.

This graph also offers a clue to the question raised on Figure 1, which shows the geographical distribution of the data points. It was noted that most of the green dots were located in Manhattan. Specifically, the green dots circle Central Park, and the rest of them were located in Midtown and Downtown Manhattan. Figure 7 shows that the green dots of Manhattan are roughly in the $500-$1700 price range. Real estate is more expensive in Midtown and Downtown than the other parts of Manhattan, which explains the relatively

high price range of the listings that were grouped in the green cluster (Cluster 3).

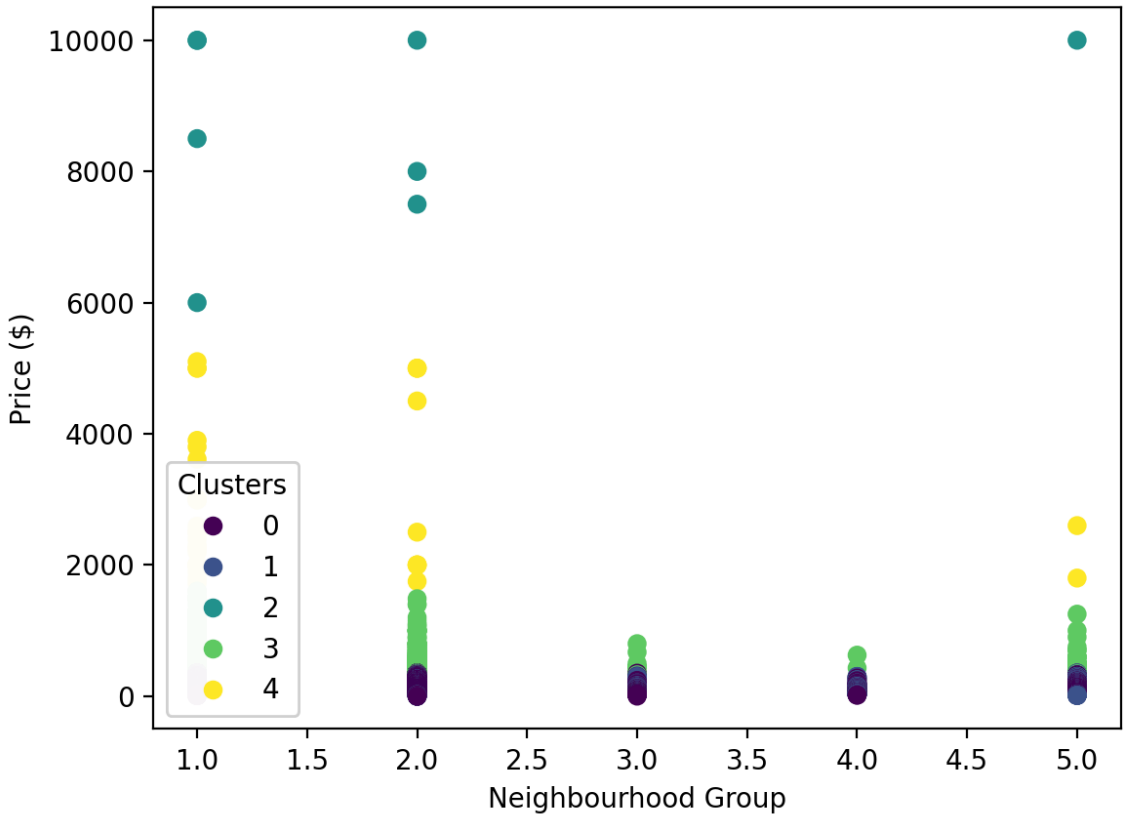


Figure 7: Neighborhood Group vs. Price ($)

Another graph was made to analyze the relationship between price and the minimum number of nights (shown below). The graph illustrates the same clustering pattern as the other graphs of pricing. It also shows that the more expensive listings require fewer nights per booking. This is expected—the more expensive listings offer higher profit margins, so the owners do not have to require too many nights per booking to turn a profit.

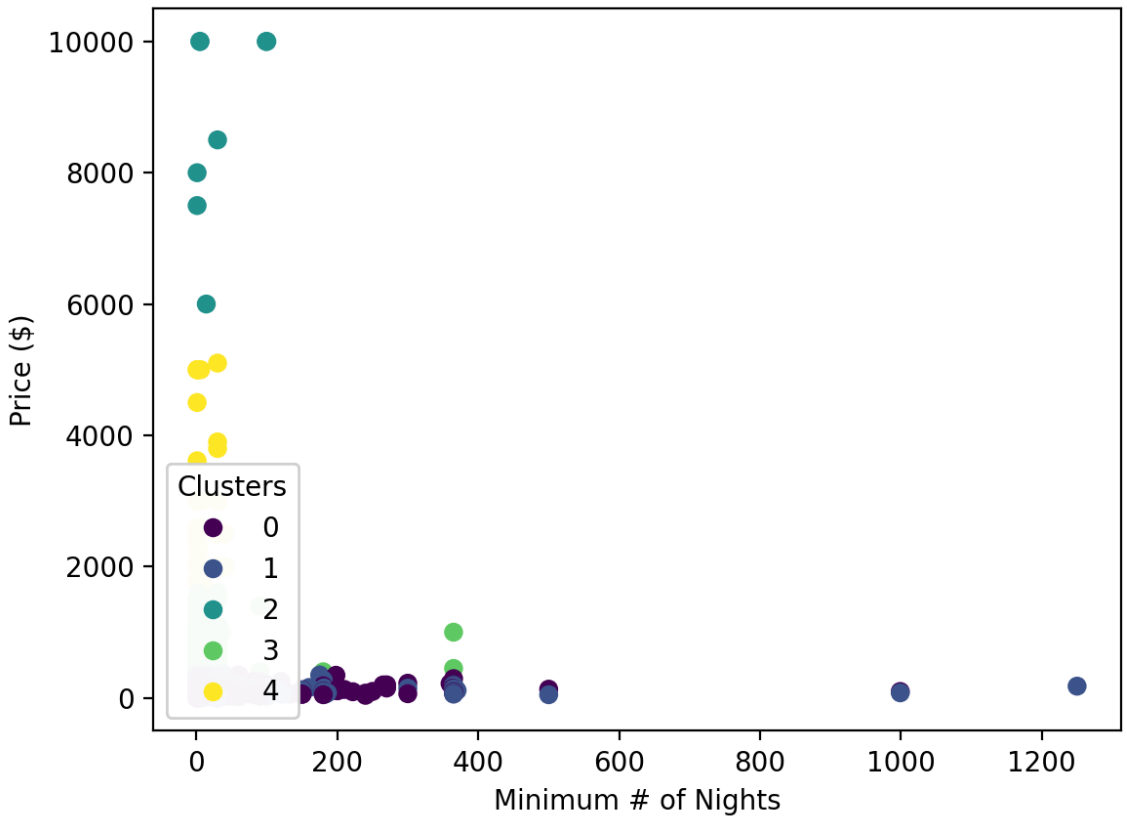


Figure 8: Minimum Number of Nights vs. Price ($)

From the graphs of various relationships between the features, it can be concluded that pricing was a significant factor in how the model made the clustering predictions. The general trends found in the graphs also match the expectations from reality, such as the locations of the more expensive listings.

Supervised Analysis

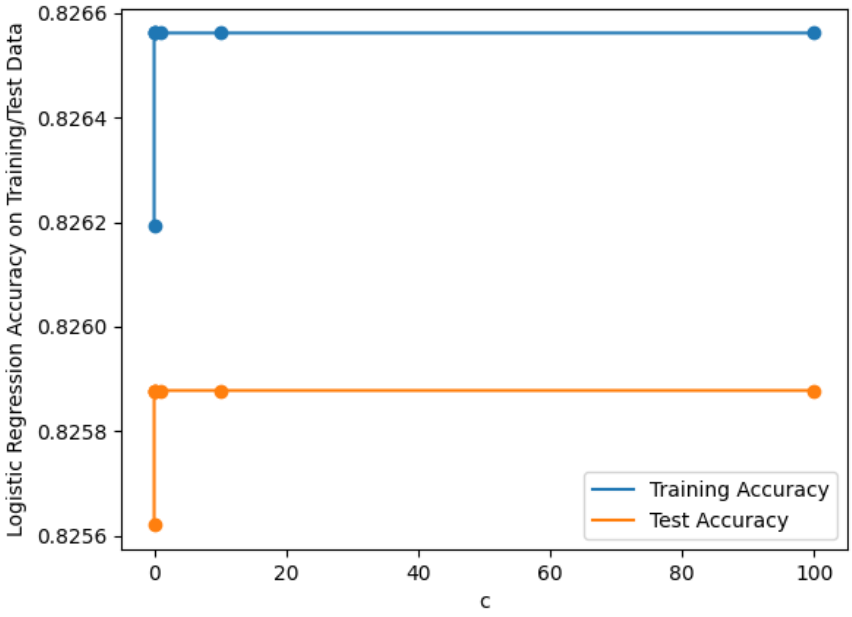
The goal of the supervised analysis was to create models that predict the price range of a listing. Eight price ranges were specified, and prices in the dataset were converted to integers from 0 to 7 in accordance with these ranges, for the purpose of training the models. The price ranges are:

Label 0: $0-$199.99
Label 1: $200-$499.99
Label 2: $500-$999.99
Label 3: $1000-$1499.99
Label 4: $1500-$1999.99
Label 5: $2000-$2999.99
Label 6: $3000-$3999.99
Label 7: $4000 and above

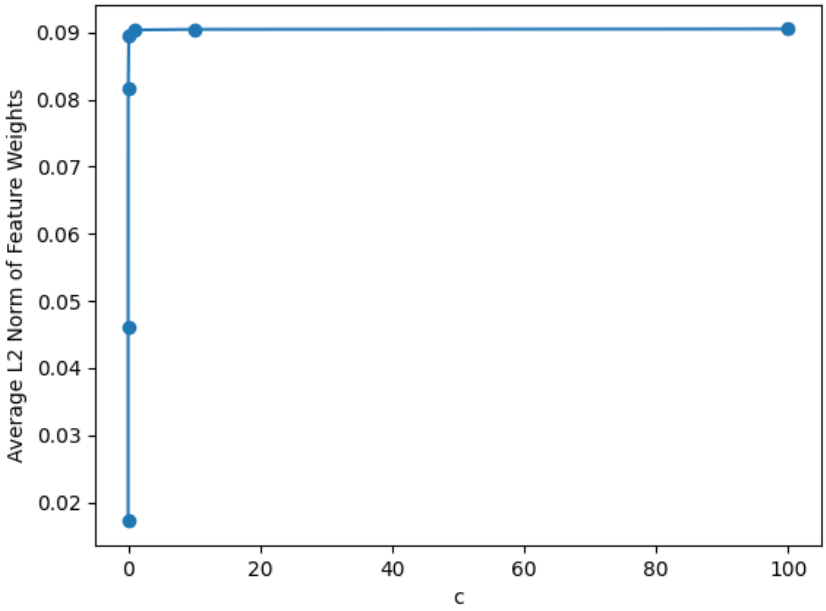
1. Logistic Regression

   Logistic regression models were created and trained on the dataset. L1 and L2 regularization, polynomial feature transformation and MinMaxScaler feature transformation were implemented. The outcomes are shown in the graphs below:

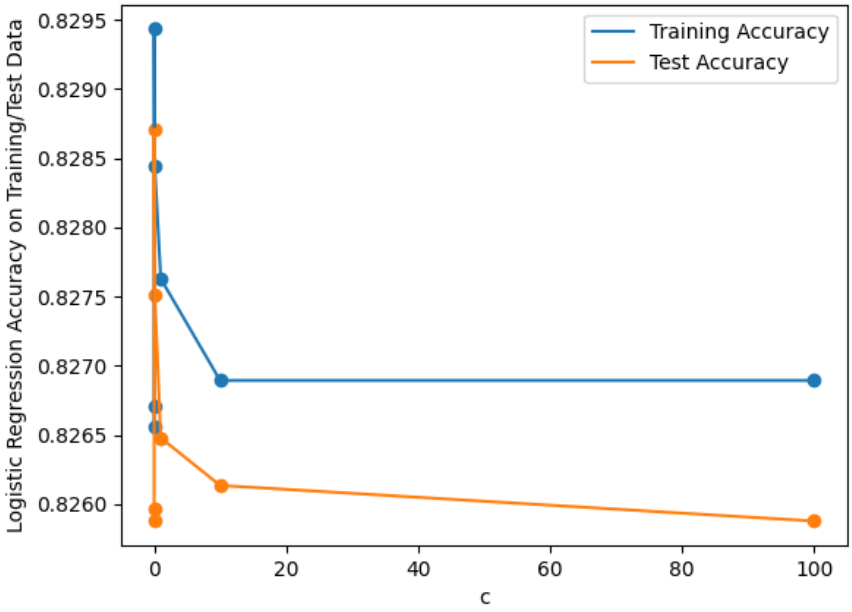
   Logistic regression with L1 regularization



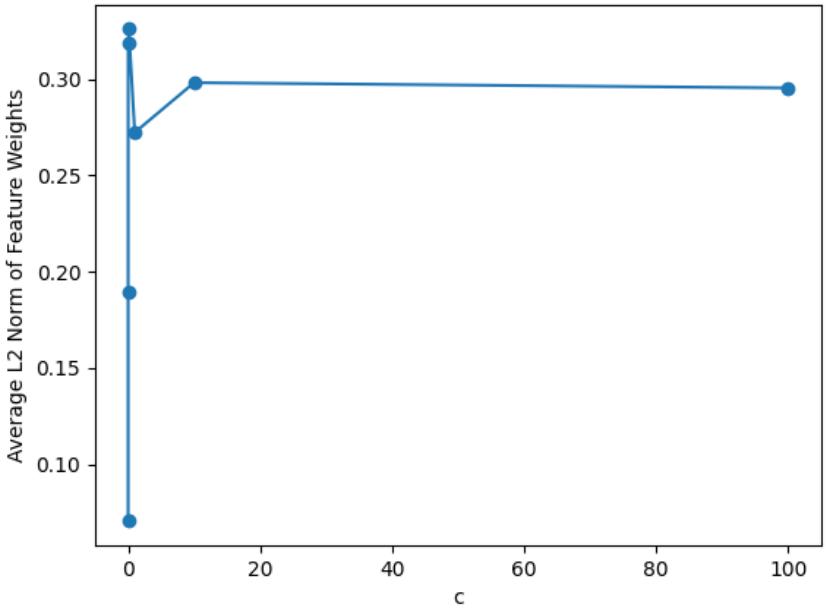
Variations in feature weights in response to variations in the parameter



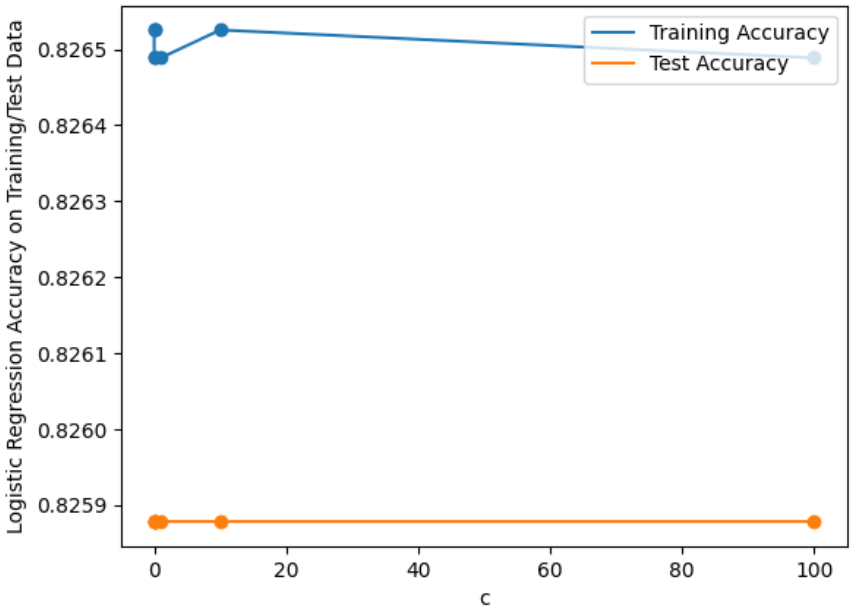
Logistic regression with L2 regularization



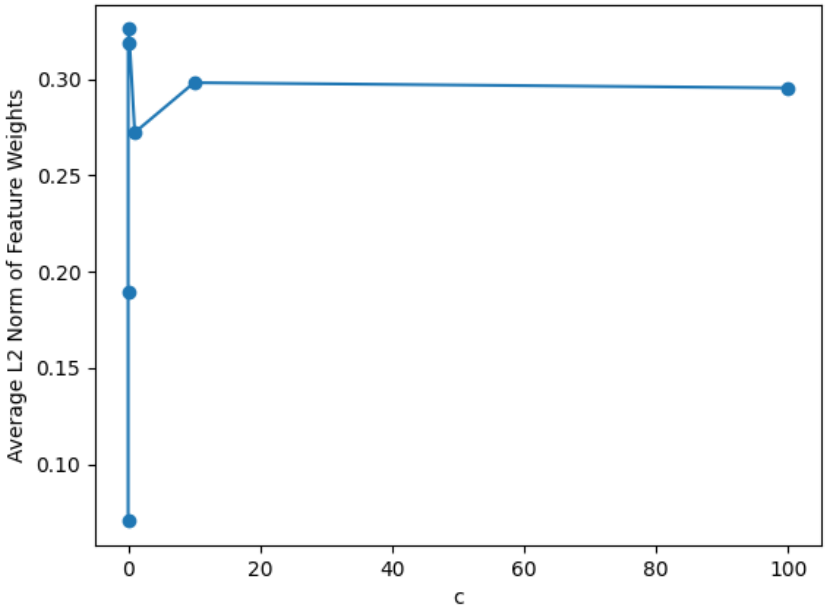
Variations in feature weights in response to variations in the parameter



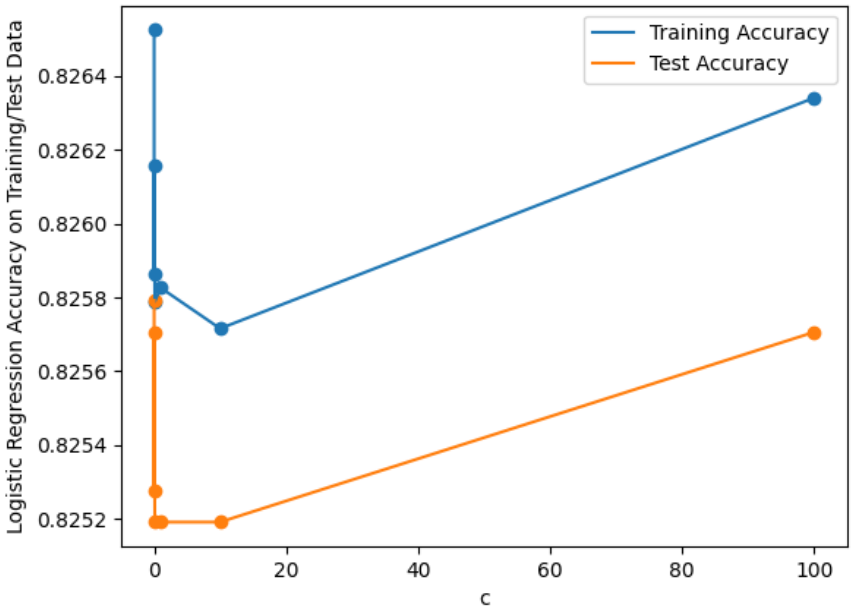
Logistic regression with L1 regularization and polynomial feature transformation



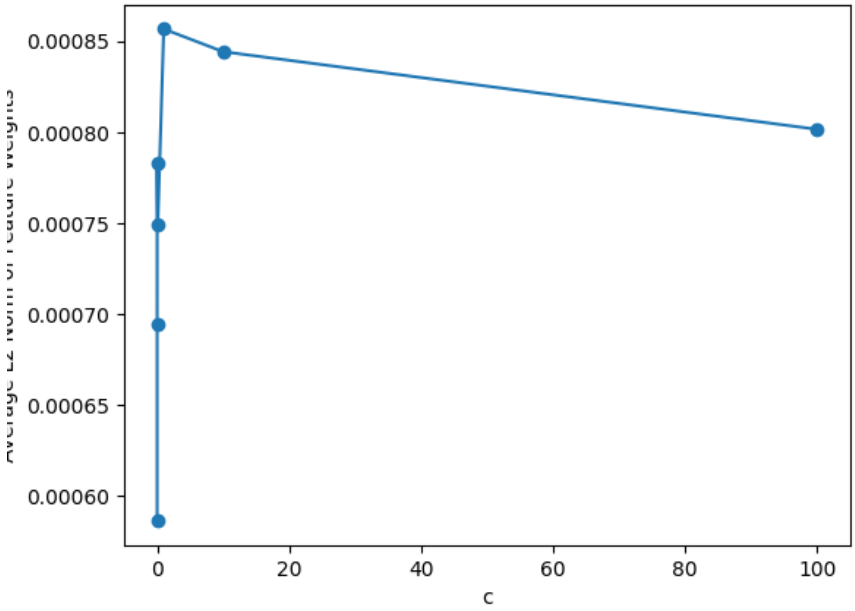
Variations in feature weights in response to variations in the parameter



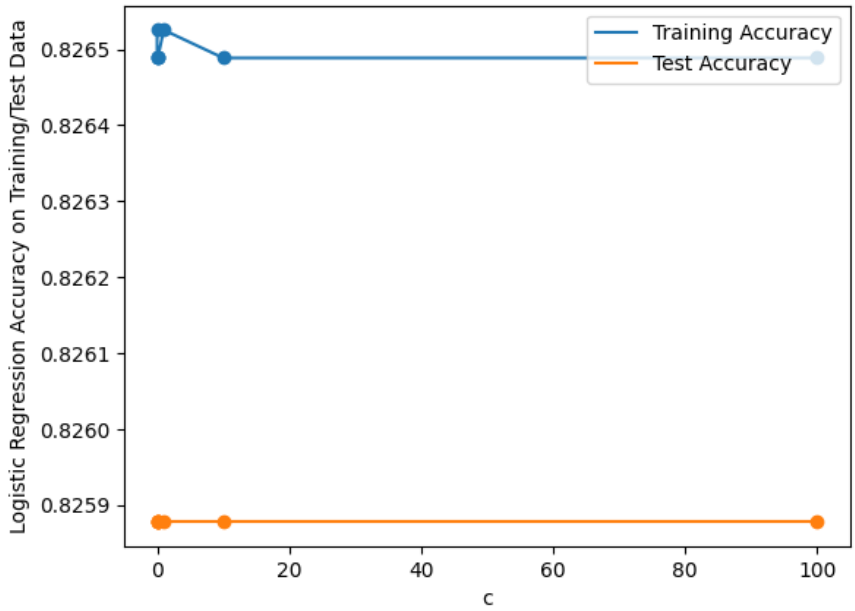
Logistic regression with L2 regularization and polynomial feature transformation



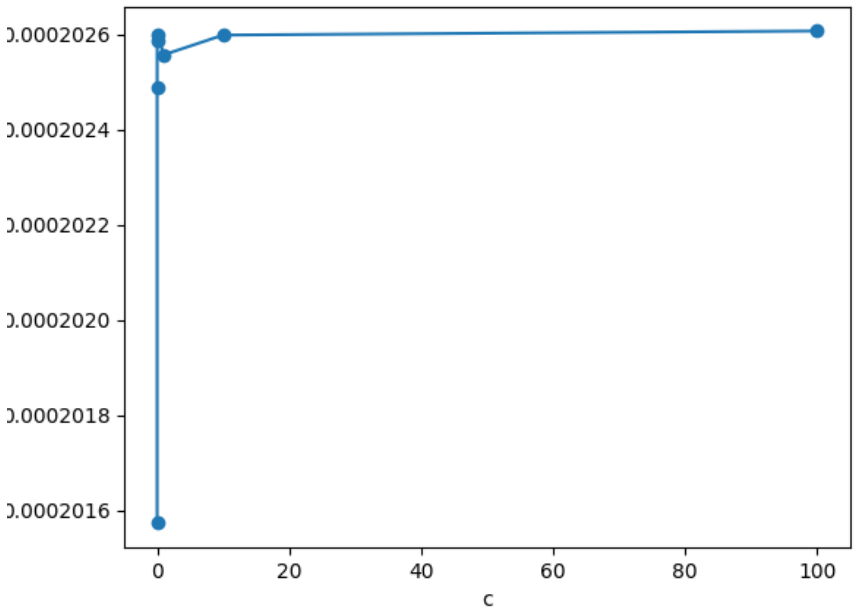
Variations in feature weights in response to variations in the parameter



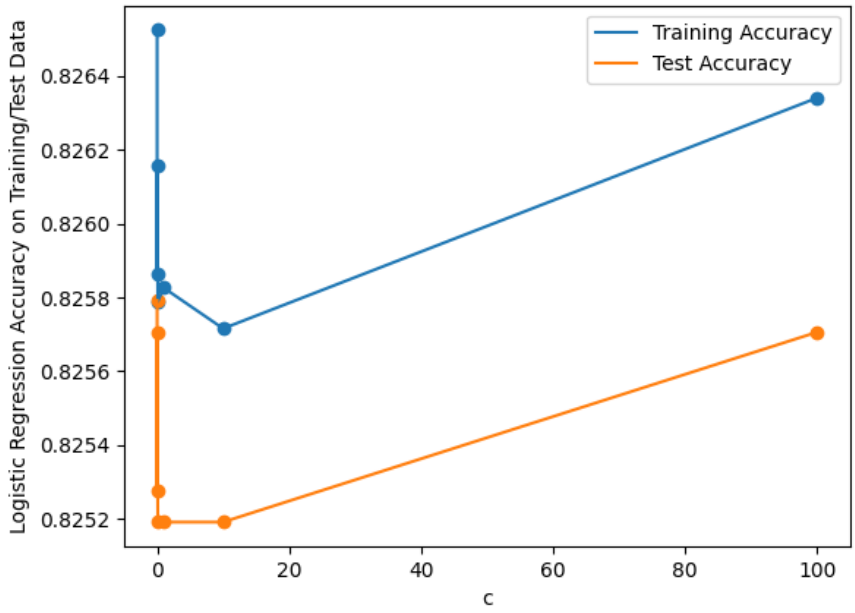
Logistic regression with L1 regularization and MinMaxScaler feature transformation



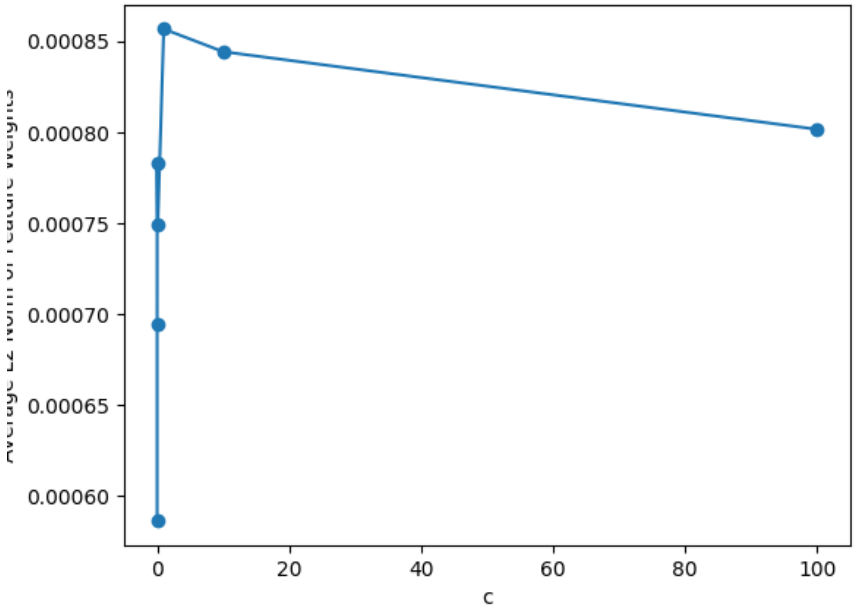
Variations in feature weights in response to variations in the parameter



Logistic regression with L2 regularization and MinMaxScaler feature transformation



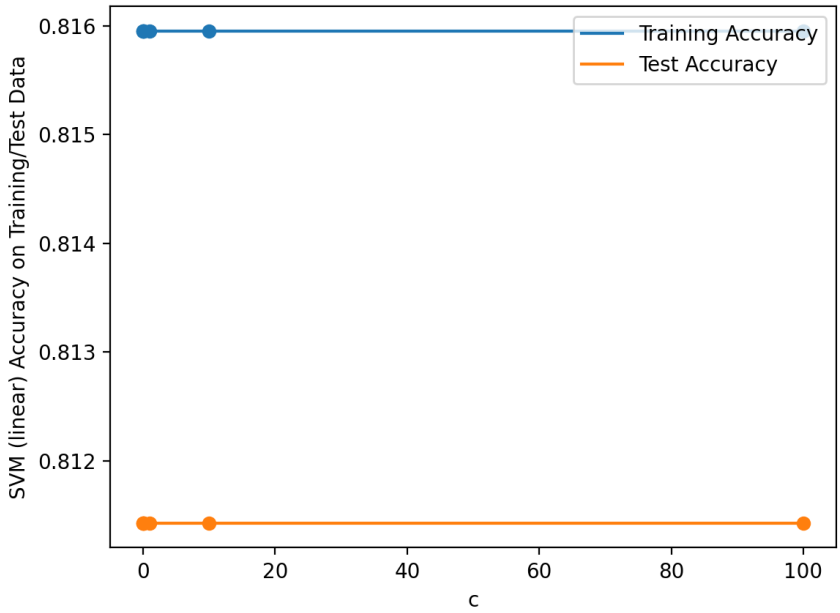
Variations in feature weights in response to variations in the parameter



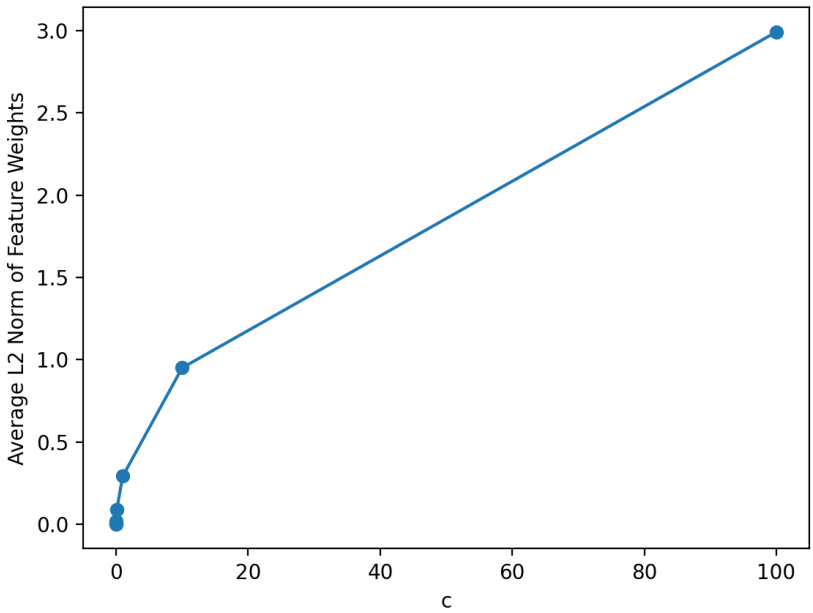
2. Support Vector Machines

   Support vector machine models were created and trained on the dataset. Linear, polynomial and radial-basis function kernels were implemented. The outcomes are shown in the graphs below:

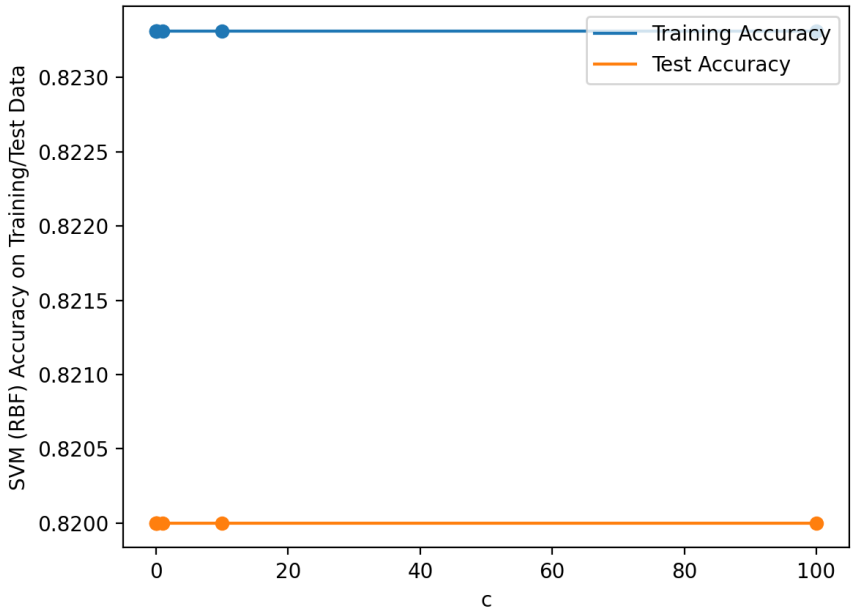
SVM with liner kernel



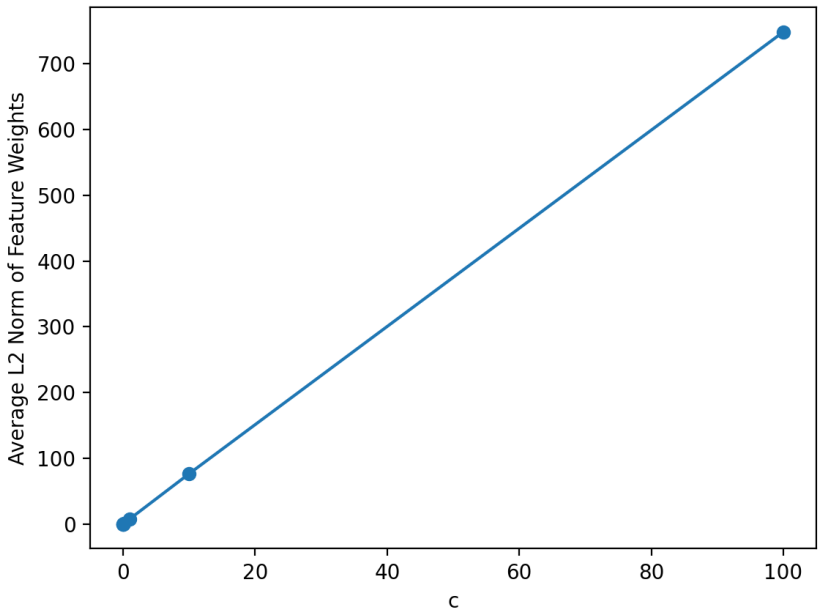
Variations in feature weights in response to variations in the parameter



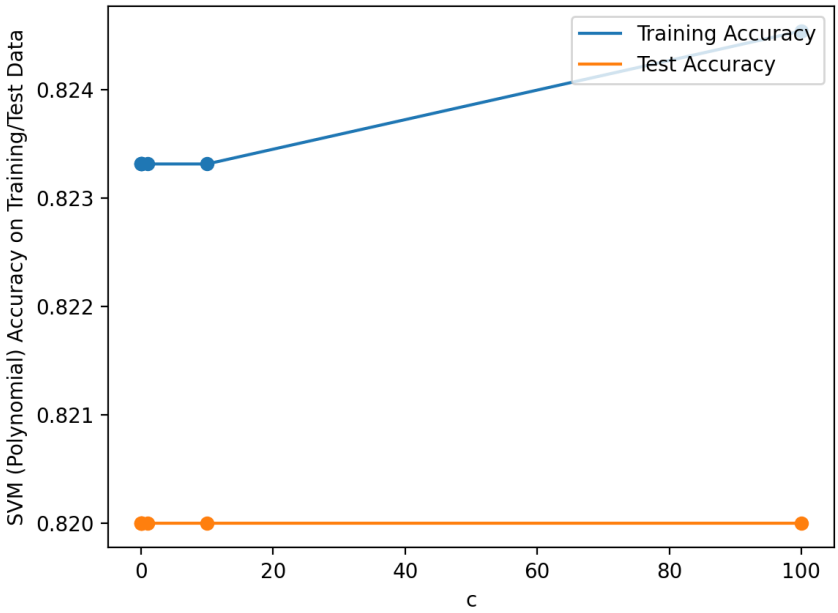
SVM with radial-basis function kernel



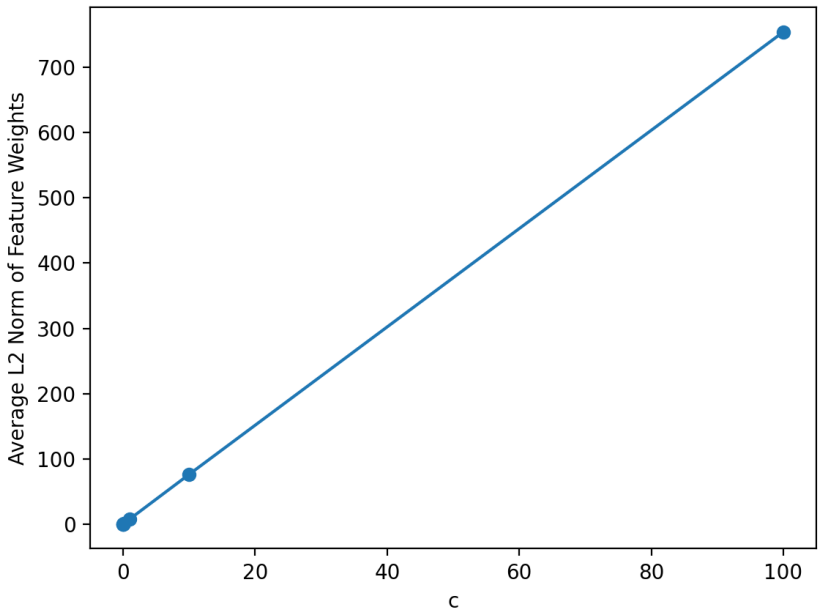
Variations in feature weights in response to variations in the parameter



SVM with polynomial kernel



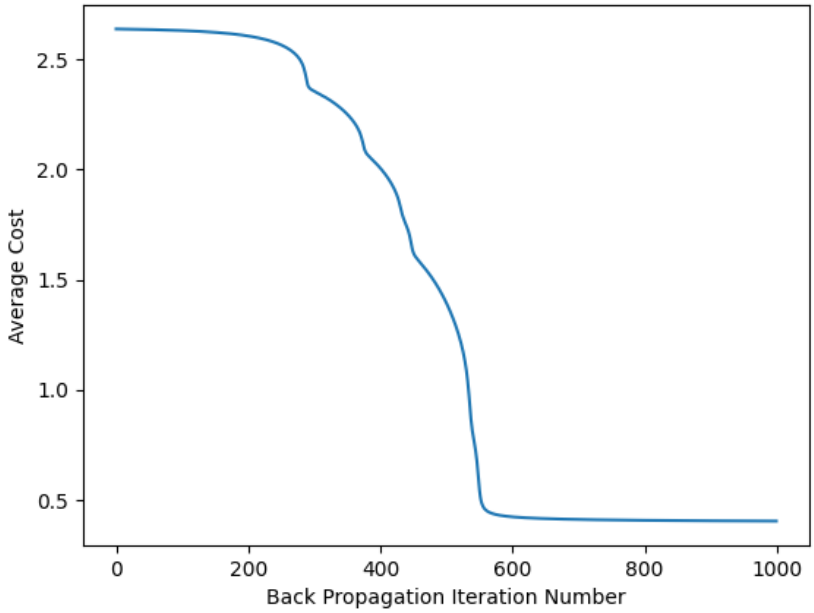
Variations in feature weights in response to variations in the parameter



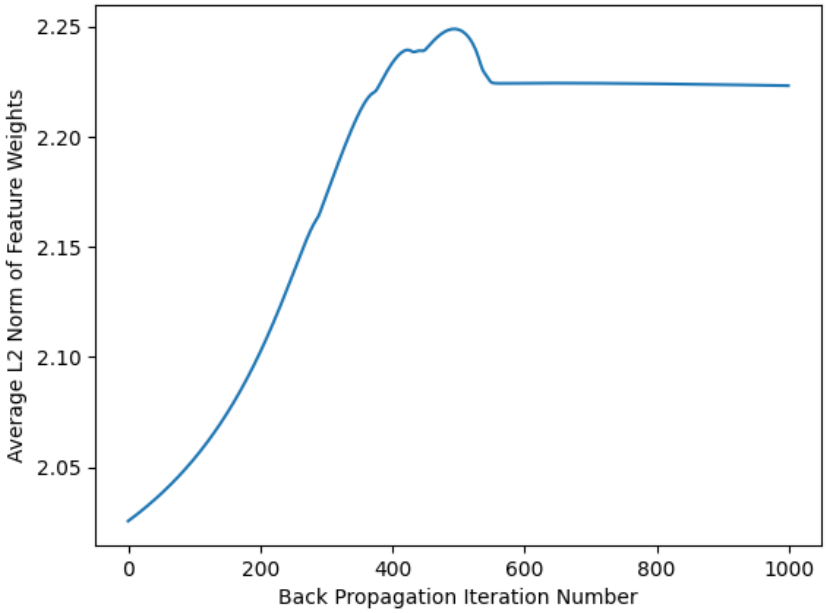
3. Neural Networks

   Neural networks were created and trained on the dataset. L1 and L2 regularization, polynomial feature transformation and MinMaxScaler feature transformation were implemented. The outcomes are shown in the graphs below:

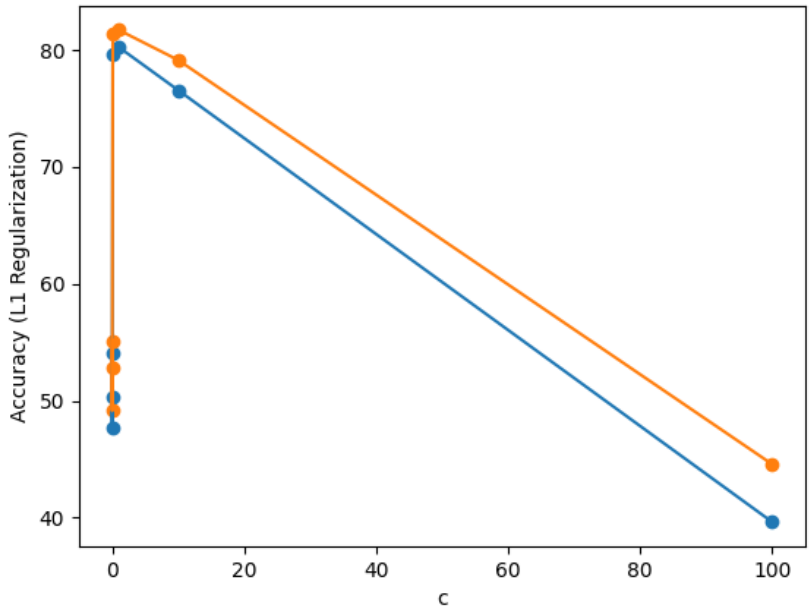
   Neural network without regularization or feature transformation



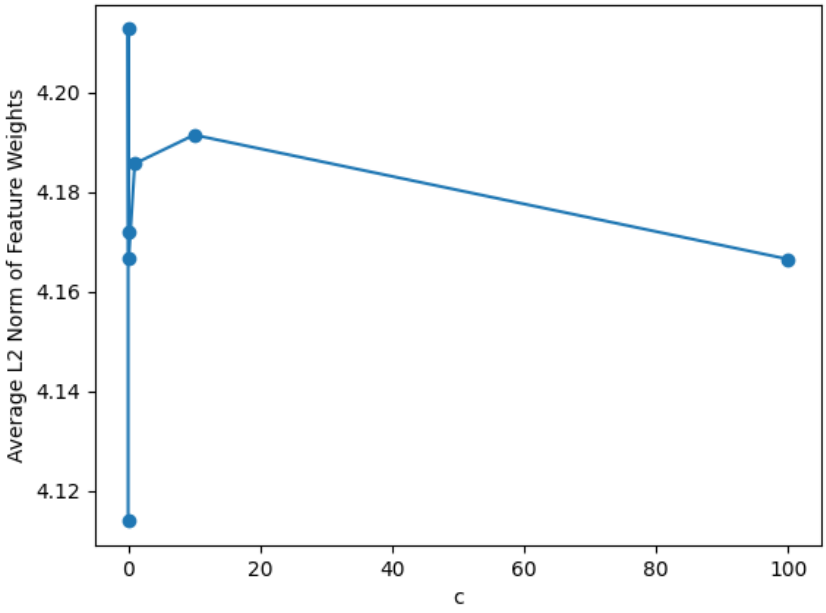
   Variations in feature weights as more iterations are run



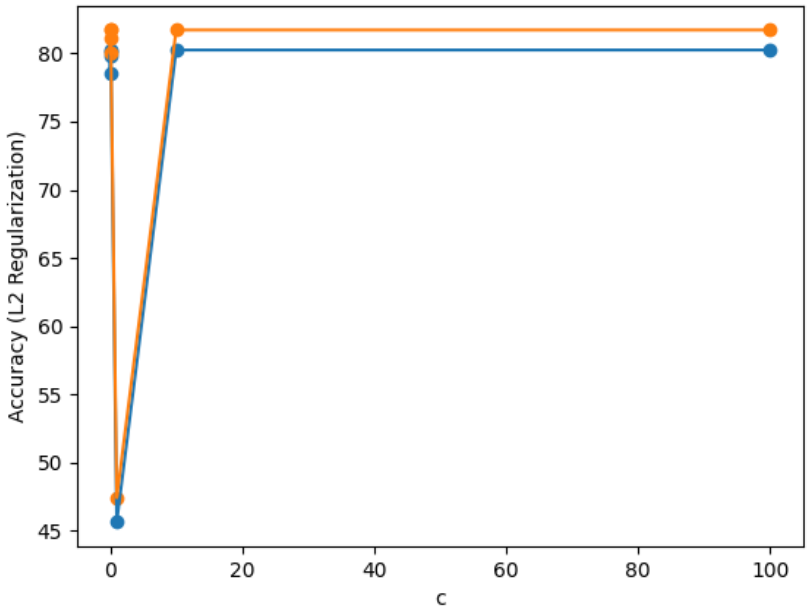
Neural network with L1 regularization and polynomial feature transformation



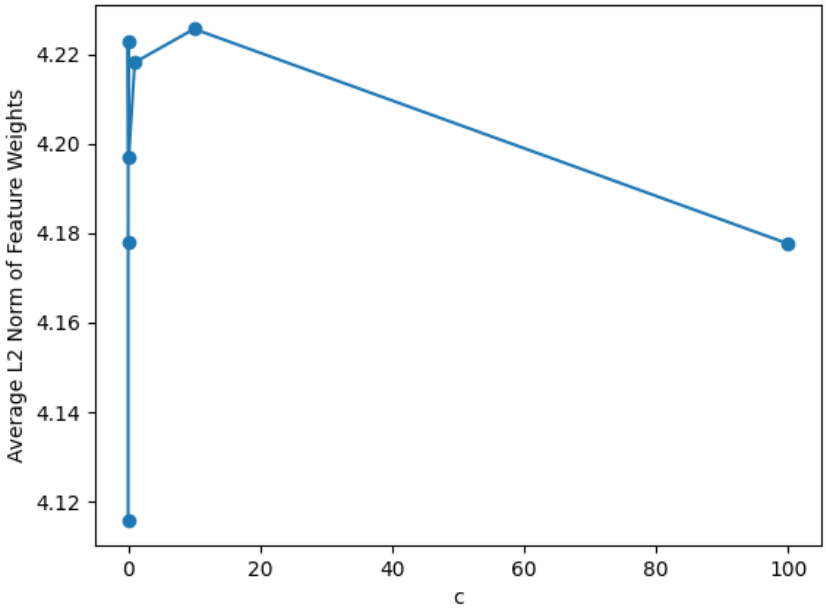
Variations in feature weights in response to variations in the parameter



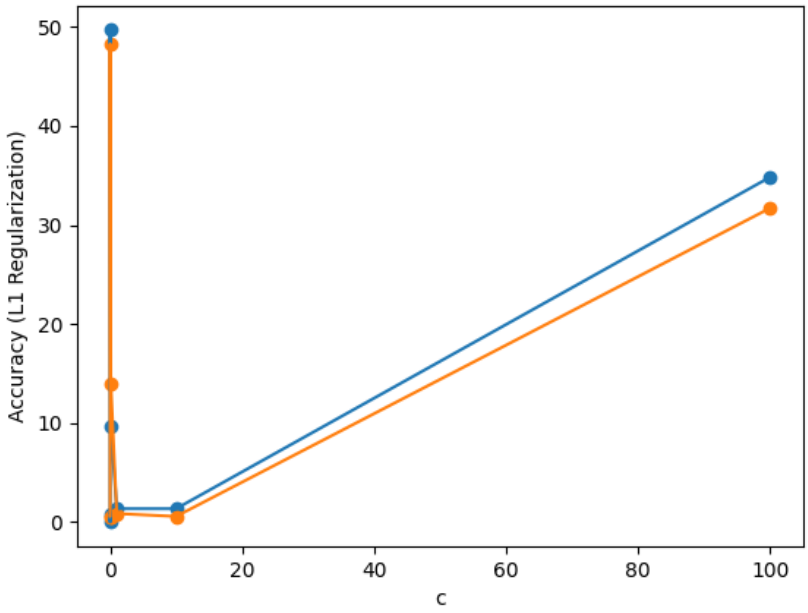
Neural network with L2 regularization and polynomial feature transformation



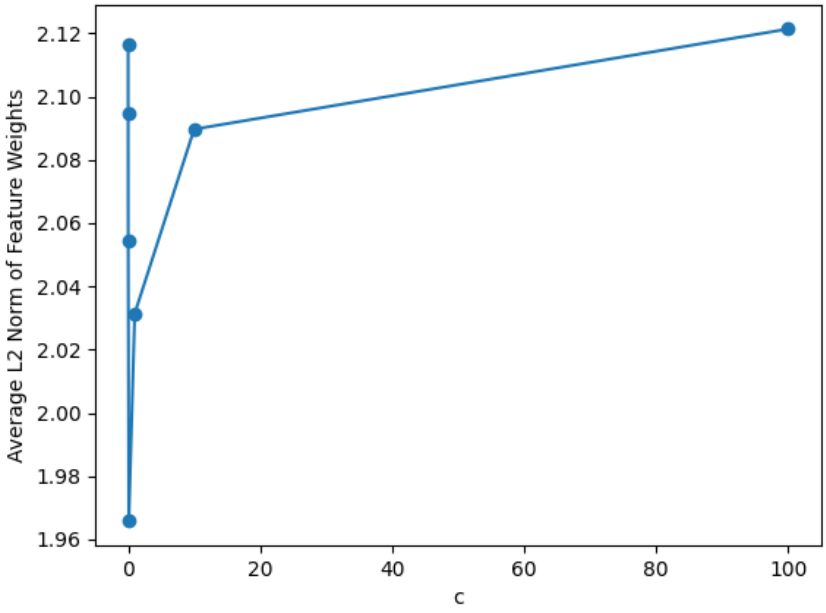
Variations in feature weights in response to variations in the parameter



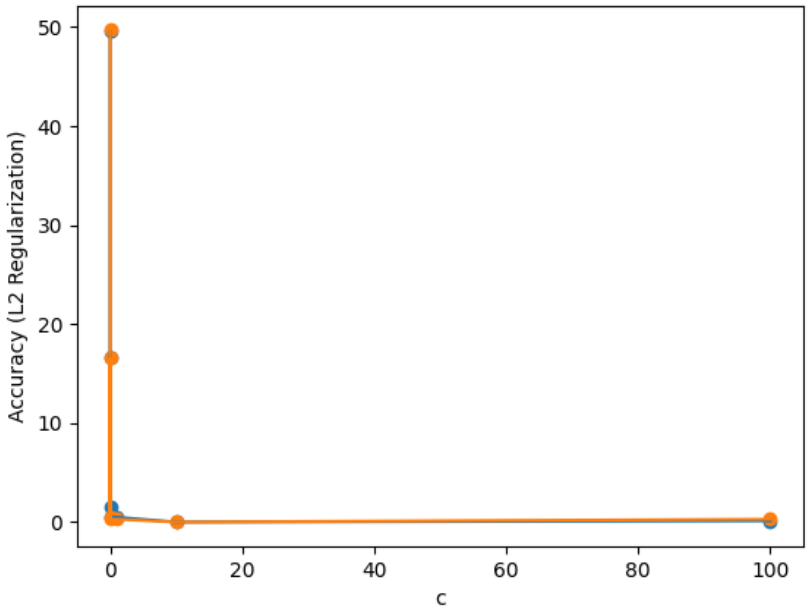
Neural network with L1 regularization and MinMaxScaler feature transformation



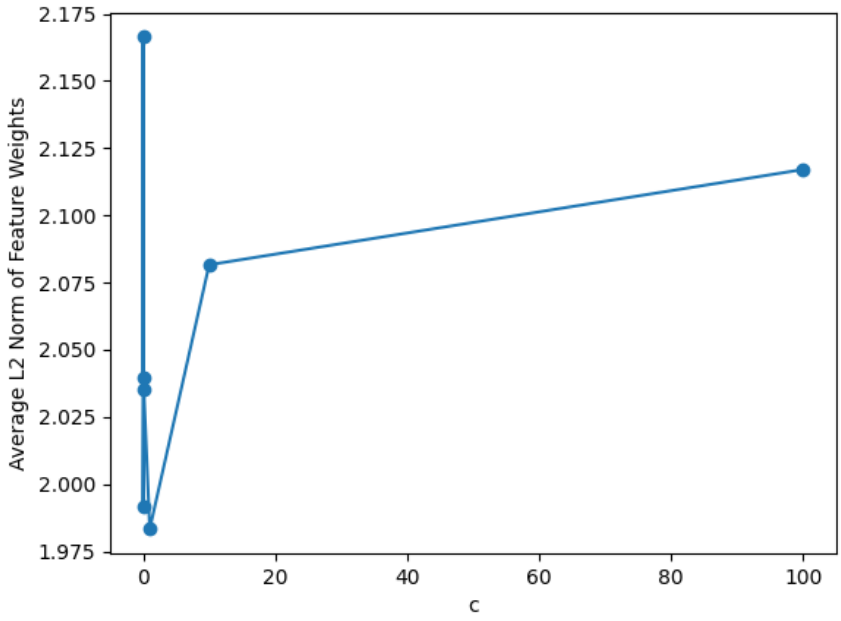
Variations in feature weights in response to variations in the parameter



Neural network with L2 regularization and MinMaxScaler feature transformation



Variations in feature weights in response to variations in the parameter


2.175
2.150
2.125
2.100
2.075
2.050
2.025
2.000
1.975
Average L2 Norm of Feature Weights
0
20
40
60
80
100
c

Tables of Results

1. Logistic Regression

   (a) Model with L1 regularization and polynomial feature transformation

	Training Accuracy	Test Accuracy
c=0.0001	0.826525	0.825878
c=0.001	0.826489	0.825878
c=0.01	0.826489	0.825878
c=0.1	0.826489	0.825878
c=1	0.826489	0.825878
c=10	0.826489	0.825878
c=100	0.826525	0.825878

   (b) Model with L2 regularization and polynomial feature transformation

	Training Accuracy	Test Accuracy
c=0.0001	0.826525	0.825792
c=0.001	0.825863	0.825277
c=0.01	0.826157	0.825706
c=0.1	0.825789	0.825191
c=1	0.825826	0.825191
c=10	0.825716	0.825191
c=100	0.826341	0.825706

   (c) Model with L1 regularization and MinMaxScaler feature transformation

	Training Accuracy	Test Accuracy
c=0.0001	0.826525	0.825878
c=0.001	0.826489	0.825878
c=0.01	0.826489	0.825878
c=0.1	0.826489	0.825878
c=1	0.826489	0.825878
c=10	0.826489	0.825878
c=100	0.826489	0.825878

   (d) Model with L2 regularization and MinMaxScaler feature transformation

	Training Accuracy	Test Accuracy
c=0.0001	0.826525	0.825792
c=0.001	0.825863	0.825277
c=0.01	0.826157	0.825706
c=0.1	0.825789	0.825191
c=1	0.825826	0.825191
c=10	0.825716	0.825191
c=100	0.826341	0.825706

2. Support Vector Machines

(a) Model with linear kernel

	Training Accuracy	Test Accuracy
c=0.0001	0.815951	0.811429
c=0.001	0.815951	0.811429
c=0.01	0.815951	0.811429
c=0.1	0.815951	0.811429
c=1	0.815951	0.811429
c=10	0.815951	0.811429
c=100	0.815951	0.811429

(b) Model with radial-basis function kernel

	Training Accuracy	Test Accuracy
c=0.0001	0.823313	0.820000
c=0.001	0.823313	0.820000
c=0.01	0.823313	0.820000
c=0.1	0.823313	0.820000
c=1	0.823313	0.820000
c=10	0.823313	0.820000
c=100	0.823313	0.820000

(c) Model with polynomial kernel

	Training Accuracy	Test Accuracy
c=0.0001	0.823313	0.820000
c=0.001	0.823313	0.820000
c=0.01	0.823313	0.820000
c=0.1	0.823313	0.820000
c=1	0.823313	0.820000
c=10	0.823313	0.820000
c=100	0.824540	0.820000

3. Neural Networks

(a) Model with L1 regularization and polynomial feature transformation

	Training Accuracy	Test Accuracy
c=0.0001	0.541104	0.551428
c=0.001	0.503067	0.528571
c=0.01	0.477300	0.491428
c=0.1	0.796319	0.814285
c=1	0.802453	0.817142
c=10	0.765644	0.791428
c=100	0.396319	0.445714

(b) Model with L2 regularization and polynomial feature transformation

	Training Accuracy	Test Accuracy
c=0.0001	0.802453	0.817142
c=0.001	0.801226	0.817142
c=0.01	0.785276	0.800000
c=0.1	0.798773	0.811428
c=1	0.456441	0.474285
c=10	0.802453	0.817142
c=100	0.802453	0.817142

(c) Model with L1 regularization and MinMaxScaler feature transformation

	Training Accuracy	Test Accuracy
c=0.0001	0.008588	0.005714
c=0.001	0.000000	0.005714
c=0.01	0.496932	0.482857
c=0.1	0.096932	0.140000
c=1	0.013496	0.008571
c=10	0.013496	0.005714
c=100	0.348466	0.317142

(d) Model with L2 regularization and MinMaxScaler feature transformation

	Training Accuracy	Test Accuracy
c=0.0001	0.495705	0.497142
c=0.001	0.014723	0.005714
c=0.01	0.166871	0.165714
c=0.1	0.004907	0.002857
c=1	0.004907	0.002857
c=10	0.000000	0.000000
c=100	0.001226	0.002857

Conclusion

Of all the models trained, the logistic regression model with L1 regularization and polynomial feature transformation yielded the highest test accuracy, 82.588%. The model was trained multiple times with different parameters, but each parameter yield the same test accuracy. Of all models, the neural networks with MinMaxScaler feature transformation yielded the lowest test accuracy on average. On the other hand, the neural networks with polynomial feature transformation yielded decent test accuracy. It can be concluded that for this dataset, polynomial feature transformation performs better with neural networks than MinMaxScaler feature transformation does.

Notably, MinMaxScaler feature transformation yielded decent test accuracy when it was used on the logistic regression models. This demonstrates that the same type of feature transformation can deliver vastly different performance on different types of models.