SF 311 Cases

Serena Leung Sruthi Rayasam Sujin Song

Serena Contribution: Question 1 and 2
Sruthi Contribution: Question 5 and 6
Sujin Song Contribution: Question 3, 4, and 7

Introduction

The “311 Cases” dataset contains non-emergency cases from the city and county of San Francisco. The dataset contains 4.8 million observations, in which each case has information like CaseID, request type, and location. Our project attempts to answer these questions:

Q1: Based on the category of the request and neighborhood, how long does it take for a case to be closed?
Q2: Given 1 day has passed, what is the predicted case status?
Q3: Given the request type, what is the predicted time of day?
Q4: Are specific 311 cases associated with a certain price level of cars?
Q5: What categories of cases have the best success rate?
Q6: Is the frequency of cases reported related to either the population or median income of a neighborhood?
Q7: Do cases that have not been closed share similar attributes?

Methodology

Q1: Perform LASSO, ridge regression, decision tree, and random forest to see how well each model can predict the time elapsed to close a case given the request category and neighborhood.

Q2: Perform Logistic Regression, Decision Tree, and Random Forest Classifier to see how accurately our models can classify the status of each observation. At the end, we plot ROC curves to directly compare the accuracy of each classification model.

Q3: Create and compare logistic regression results of two models, with or without reducing the dimension (PCA), predict and calculate accuracy scores to compare the two models.

Q4: Extract car brands from request details, perform TF-IDF to transform text into numbers that represent overall relevance of the words, perform PCA to see which request type is the most common.

Q5: Made dummy variables for the category columns, performed LASSO and Ridge regression and transformed the data for better accuracy. Found the most significant variables/categories.

Q6: We approached this problem by grouping together the neighborhoods and then finding the frequency of occurrences per neighborhood. We found income and population levels divided into low, medium, and high and plotted pca plots. We made pca plots for training and testing data, used random forest classifiers, confusion matrices, and explained variance ratios. Used LASSO and Ridge regression and transformed the data to achieve higher testing and training accuracy. We found the coefficients for population and income.

Q7: Extract the unclosed cases and construct a PCA plot to see if they share similar attributes.

Implementation Details

Q1: To predict the time elapsed, we performed LASSO and ridge regression because they better handle correlated predictors compared to linear regression. Since we had hundreds of dummy variables from converting categorical data, LASSO seemed fitting in this case. However, since our data is mostly categorical, LASSO and Ridge regression may not be the best option because they may be more suited for data with continuous attributes. Thus, we performed Random Forest because the structure of this model can more easily deal with categorical data. The Default Random Forest serves as a base model for the Random Forest models with tuned hyperparameters. RandomSearchCV was performed to get a general idea of hyperparameter range. Then, GridSearchCV was used to iterate through a narrower range of parameters based on our results from RandomSearchCV.

Q2: We performed logistic regression on the data by using the default values in the LogisticRegression(), then testing the accuracy with different C values (regularization strength), and finally performing GridSearchCV on the logistic regression. The default logistic regression serves as a base model for other logistic model variations. For tree-based models, we performed a single Decision Tree that acted as the base model comparison for the Random Forest model.

Q3: For logistic regression, we are trying to predict if the request was made during the day or night time based on the attributes. We added a “day_night” column that labeled each row (day = 0 & night = 1) looking at the requested time. We set X as all the columns except the “day_night” column and y as the “day_night” column. We performed logistic regression with two different types of data: original and PCA transformed data. The data is imbalanced, so we set the weights parameter to “balanced.” We reduced the dimensions with PCA because the new dataset is high-dimensional due to dummy variables. For both models, we split the data into testing and training sets and obtained the accuracy rates. We then predicted the testing group using Logistic Regression and found the confusion matrix and accuracy rates of the predicted values for comparison.

Q4: We took the “Request Details” column and extracted the car brands written in car-related requests. Car brands were separated into different price levels (high, medium, low) based on the classification list. We used the TfidfVectorizer function on the request type column to transform the text into numbers that consider the overall relevance of the words. PCA was performed to reduce the dimension, and we plotted the PCA results by price level to see if certain 311 cases are associated with the price. Density is shown through the size of the points, so we identified the request type that corresponded to the largest point.

Q5: We created dummy variables for the ‘Category’ column of the data. Our X variable was a dataframe of all the dummy variables and y was the Time Elapsed. We then used LASSO and Ridge regression to see which option would yield the best testing and training accuracy. We noticed that changing the alpha values did not yield much change to the testing or training accuracy so we used the default parameter and normalized the data. We also plotted the variables against the coefficients to see the best performing category that had the strongest relationship with the amount of time elapsed.

Q6: We found the frequency of each neighborhood occurring in the dataset. Then, we found the income and population of each neighborhood and combined it with our previous data. We divided the income and population into levels of low, medium, and high and used those as the target for our pca plots. We made two pca plots, one with frequency and income and the other with frequency and population. We then split the data into training and testing and fit pca for those data. We also found the explained variance ratio using the standard scalar fit, and made a confusion matrix using the random forest classifier with a maximum depth of 2. We thought 2 would be the perfect parameter for this data to ensure that variance would not increase and the data would not be overfitted. We then fit the three variables to a LASSO and Ridge regression and transformed the data for better accuracy. Again, we normalized the data and used the default alpha parameter. We also plotted the coefficients of population and income.

Q7: We extracted all the rows containing NA values in the “Closed” column and subsetted seven most common request types to observe if they had similar characteristics. We scaled the variables to normalize the distribution and fit a PCA plot to see if they share similar attributes.

Results and Interpretation

Q1: To gauge our regression models, we used the metric MAE, MSE, and R^2. MAE (Mean Average Error) is the average log hour difference between predicted and true value.

In general, the results for LASSO and Ridge Regression were fairly close, centering at around 0.37, with LASSO having a slightly higher MSE (Q1: Table 1). For the Random Forest Regression models, the Default Random Forest performed fairly well, compared to the other models with SearchCV. In general the model with GridSearchCV outperformed the model with RandomSearchCV, but it had a similar performance with the Default Random Forest model (Q1: Table 2).

The "Q1: Data vs Predicted" plot shows the difference in fluctuation between the true data and the predicted results: the true data has much more extreme fluctuations in time elapsed, whereas the predicted results has a more consistent variation. However, this is not surprising because our data contains outliers, so our model should best fit the non-outlier data points.

Q2: Between the tested C values, C = 1 appeared to be the best value because the model with GridSearchCV also chose C = 1. The test accuracy for logistic regression models center around 0.72, with the GridSearchCV model having the highest test accuracy.

Surprisingly, the Decision Tree performed almost as well as the Random Forest model. There was 0.01 improvement in test accuracy (Q2 Table 2), possibly because our Decision Tree is relatively stable. Our data mainly consists of dummy variables, so classification is simple. Thus, aggregating Decision Trees to create a Random Forest doesn't significantly improve the accuracy as the submodels already return a similar prediction.

ROC curves are used to measure and compare binary classification model performance. Ideally, a model with stellar separation capacity has AUC=1 and a model with no separation capacity (ie: a model that makes random guesses) has AUC=0.5. Based on the models in Q2, we have a consistent AUC of around 0.78 ~ 0.79 among all classifiers, which indicates that they have an acceptable and similar classification performance (Q2 Plot).

From this experience, I learned that tuning hyperparameters is time-consuming and difficult. When setting the range of possible hyperparameters to test, the range must include the unknown optimal hyperparameter or else the results will likely be the same (if not worse) then the default model.

Q3: Results obtained from logistic regression after PCA yielded lower training and test accuracy score compared to the logistic regression on the original, dummy variable data (Q3 Table 1). We can conclude that PCA could not take every variable into account when separating the components. We calculated the cumulative percentage of the explained variance, and the percentages are low (~7.22% for first and ~13.83% for second component). Therefore, we can conclude that the PCA components do not reasonably explain the variance, and the variables show weak correlation with one another, making it difficult to group based on commonalities and condense the data into reduced dimensions. We tried to change the number of components, but we didn't observe significant changes unless n > 100. Overall, we learned that it is not efficient to reduce the dimension into a number of groups without knowing the general correlation between the variables and that some components better explain the variance than others.

Q4: We found that both high and medium priced cars are most associated with "Blocking_Driveway_Only" and the low price is with "Other_Illegal_Parking." All three PCA plots seem to follow a parallel pattern of the point in the middle being the smallest while the rest roughly were of the same size. Though we identified which cases are most common with coding, it is difficult to identify the largest point on all the graphs due to similarity in sizes. Consequently, it is hard to explicitly pinpoint which 311 cases are linked to the specific price level of cars.

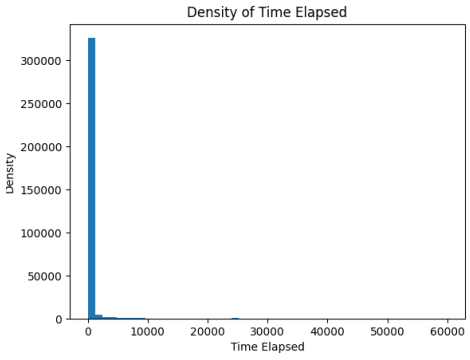
Q5: Improving the accuracy for this data proved to be difficult. When the Ridge Regression fit was transformed by squaring y (Time Elapsed), the training and testing accuracy was reduced by almost half. When y was transformed to y^(1/16) of the original value however, the accuracy improved about three-fold. This transformation yielded the best results between all of the Regression fits. The best LASSO fit was when the model was left undisturbed. Squaring y for the LASSO fit halved the accuracy. Transforming X did not prove to

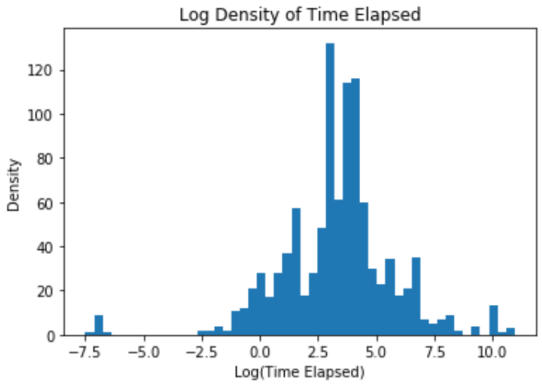
be helpful either. We then plotted the coefficients against the variables and found the significant variables. The most significant category in terms of the highest coefficient is the ‘General Request Type’. Both the Ridge and LASSO Regression plots displayed this.

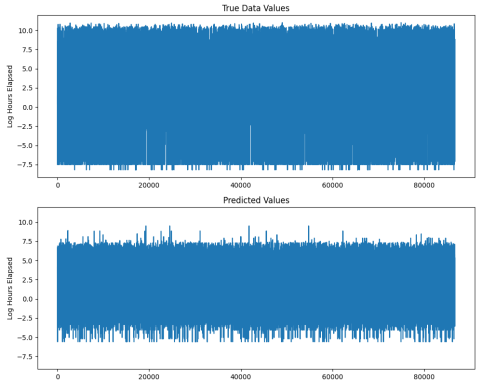
Q6: The pca plot for frequency and income showed all the income levels clustered together. There was a clear distinction of each group and the high income level was clustered together the closest, while the medium income level had some outliers, and the low income level was clustered the least closely. Because the values are so distinct, we can conclude that the data representation is most likely sufficient for a classification model. After fitting pca and a standard scalar fit to the data, the explained variance ratio for the first component was at 62.7%. This shows that both principal components are necessary to explain the data. The confusion matrix yielded good results with the true positives making up 64% of the matrix. The accuracy score was much better than anything the Ridge or LASSO models and their corresponding transformations yielded, proving that this was the best fit for the model. The second pca plot’s clusters also fit very distinctly. While the high population level was a little scattered, the medium and low levels of the population were packed very closely. After fitting the pca and standard scalar fit to the training and testing data, we concluded that frequency and population variables yielded the best results. 92.7% of the variance was explained in the first principal component, 85% of the confusion matrix was of true positives, and the accuracy of the fit was at 85%. The Ridge and Lasso regression accuracies and the transformed accuracies also were much better for frequency vs population than for frequency vs income. However, the Ridge and LASSO regression accuracies again, were not as high as the accuracy obtained after the standard scalar fit. We initially theorized that income vs frequency would yield the best accuracy, especially after frequency was transformed, because of the discrepancies in both frequency and population, but this theory was clearly wrong. Population clearly has a much better relationship with frequency. The coefficient versus variables plots for income and population versus frequency also showed that population had a stronger relationship with frequency.

Q7: The variables for the most part are scattered very closely. They are not distinct or separate in the plot which implies that the data is not sufficient for this model. With large disparities between values, it is clear that the variance for PC1 and PC2 are not optimal. The groups have no distinguishing characteristics between them and do not contribute to the request type.

Question 1 and 2:

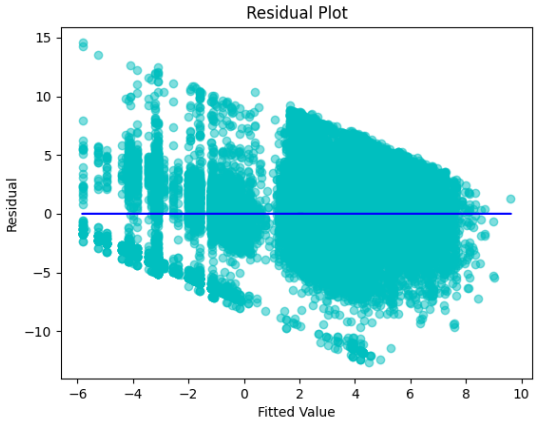






Q1:Table 1

	Score	MSE
LASSO	0.3707	5.4342
Ridge	0.37069	5.3833



Q1: Table 2

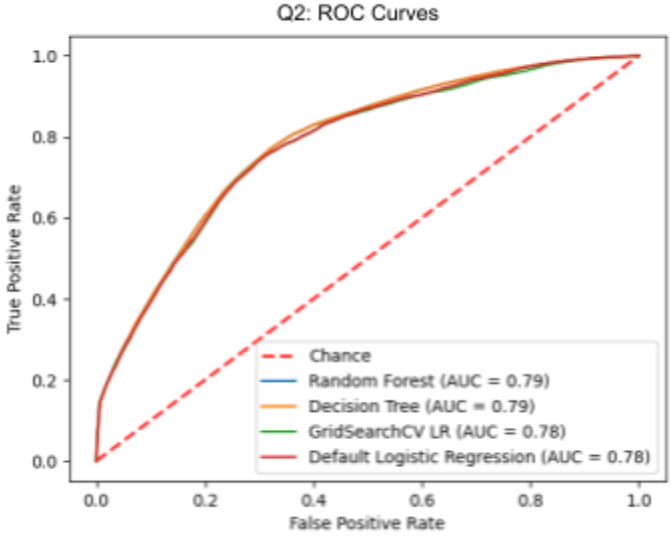
	Default Random Forest	RandomSearchCV	GridSearchCV
Best Parameters with Search CV		{'max_depth': 5, 'max_features': 'auto', 'min_samples_split': 3, 'n_estimators': 85}	{'max_features': 4, 'n_estimators': 80}
MAE	1.7075 log(hrs)	1.7520 log(hrs).	1.6970 log(hrs)
MSE	5.189977	5.4414715583	5.12431202
R2	0.395929	0.366400641	0.40431437

Q2: Table 2 Tree-Based Models

	Test Accuracy	Confusion Matrix	Get Params or Best Params
Decision Tree	0.72823	[[45737 24908] [22216 80542]]	{'criterion': 'gini', 'max_depth': None... 'min_samples_leaf': 1, 'min_samples_split':2...}
Random Forest Classifier (GridSearchCV)	0.73656	[[44303 26342] [19339 83419]]	{'max_features': 20, 'n_estimators': 30}

Q2: Table 1 Logistic Regression

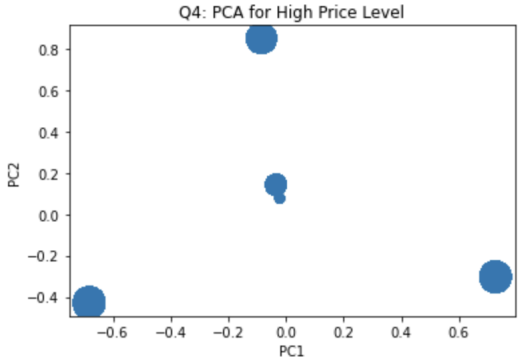
Logistic Regression Iterations				
	C	Training Accuracy	Test Accuracy	Confusion Matrix
	10	0.728678	0.728822	[[44205 26669] [20354 82175]]
	1	0.728667	0.728782	[[44180 26694] [20336 82193]]
	0.1	0.728298	0.728707	[[44111 26763] [20280 82249]]
	0.001	0.723673	0.725218	[[49786 21088] [26560 75969]]
GridSearchCV				
	1	NA	0.7288916	[[46054 24660] [22351 80338]]

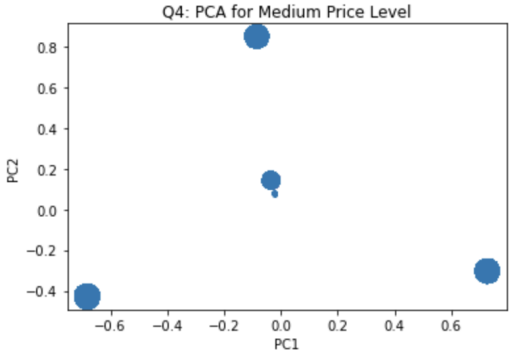


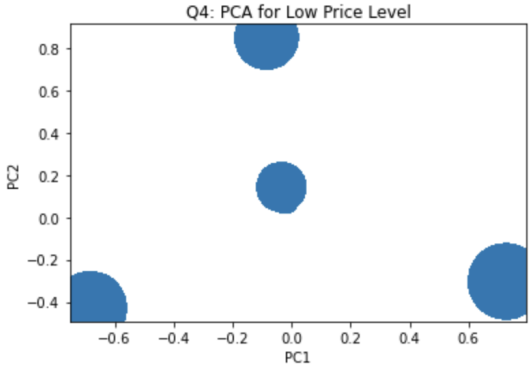
Question 3 and 4:

Q3 Table 1: Accuracy and Confusion Matrix for Logistic Regression Iterations

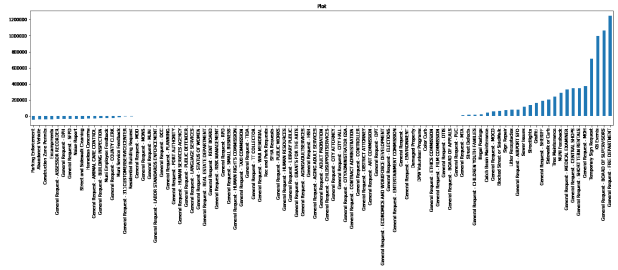
Methods	Training	Test	Confusion Matrix
PCA	0.525686864110063	0.5279830842423952	[[99493 100273] [14876 28128]]
Without PCA (Original)	0.6342587634386456	0.6270748233937239	[[127499 72267] [16524 26480]]

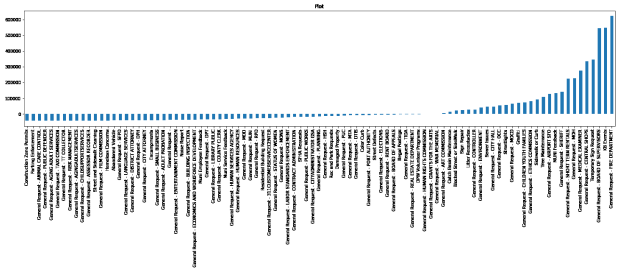






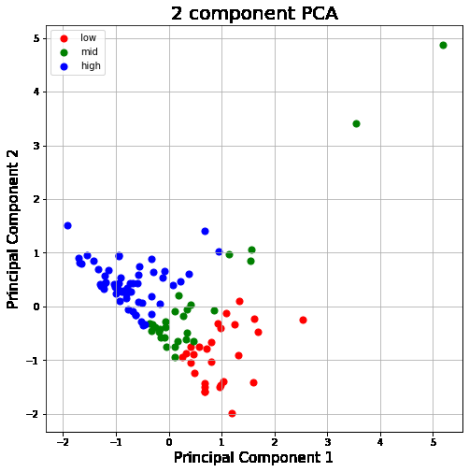
Question 5 and 6:





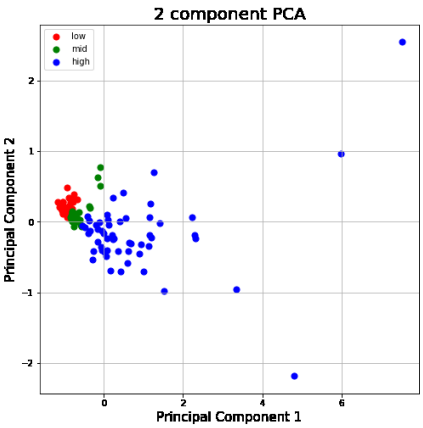
Q5:Table 1: Accuracy

LASSO/Ridge	Testing accuracy	Training accuracy	Transformation
Ridge	0.08620076453862358	0.08632223834540487	-
Ridge	0.0391314517332817	0.03983087143386243	square(y)
Ridge	0.275442904462024	0.27556554389367427	y^(1/16)
LASSO	0.11250924501536252	0.11163413874226946	-
LASSO	0.05185337150177371	0.05202694418372156	square(y)

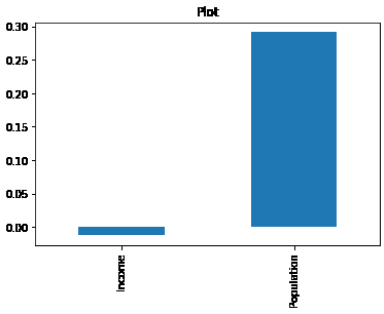


PCA for Income

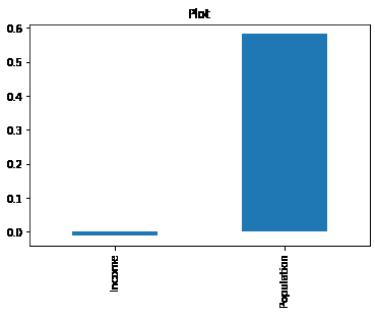
PCA for Population



coefficients vs variables Ridge



coefficients vs variables LASSO



Q6: Table 2: Accuracy of Frequency and Population

LASSO/Ridge	Testing accuracy	Training accuracy	Transformation
Ridge	0.25586108061	0.5475282343659	-
Ridge	0.120900879522	0.42937268528916	log(Population)
Ridge	0.401005872428	0.34687868607665	log(freq)
Ridge	0.262490713883	0.30042624354109	Remove outliers from frequency
Ridge	0.0	0.0	Remove outliers from log(freq)
LASSO	0.492512587913	0.7300359515829	-
LASSO	0.256769416148	0.57249521948054	log(Population)
LASSO	-0.271995220419	0.0	log(freq)
LASSO	-0.110307621671	0.0	Remove outliers from log(freq)
LASSO	0.0	0.0	Remove outliers from frequency

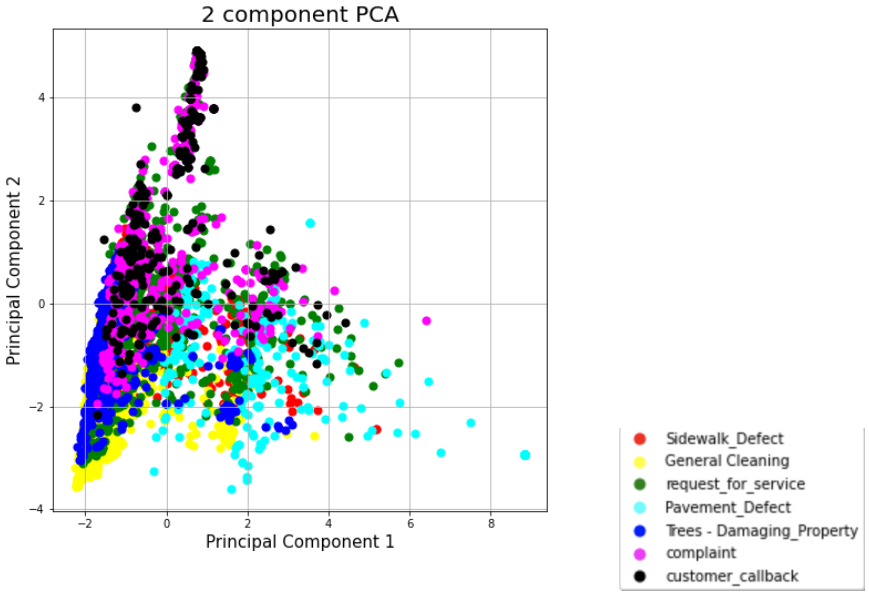
Q6: Table 3: Variance Ratio and Confusion Matrix

Variables	Explained Variance Ratio	Confusion Matrix	Accuracy
Frequency, Income	[0.62732165 0.37267835]	[12 3 2] [0 8 0] [1 6 2]	0.6470588235294118
Frequency, Population	[0.9272112, 0.0727888]	[20 0 0] [0 3 2] [2 1 6]	0.8529411764705882

Q6: Table 4: Accuracy of Frequency vs Income and Population

LASSO/Ridge	Testing accuracy	Training accuracy
Ridge	0.2550488240914113	0.5613168484799629
LASSO	0.4892485457736159	0.7364917272450566

Question 7:


2 component PCA
Principal Component 2
Principal Component 1
4
2
0
−2
−4
−2
0
2
4
6
8
Sidewalk_Defect
General Cleaning
request_for_service
Pavement_Defect
Trees - Damaging_Property
complaint
customer_callback