BASKIN SCHOOL OF ENGINEERING - TIM 245

Final Project Phase

Panos Karagiannis
Nehal Bengre
Shiqi Wen
Aakash Thakkar

September 24, 2017

CONTENTS

1 Executive Summary

This project shows the utilization of data mining on several postal office services across four data mining tasks: classification, prediction, cluster analysis and association analysis. The techniques proposed in the next paragraphs contain models that accurately recognize diversified handwritten characters and digits, extract organized customer behaviors for further investigation and also predict the revenue of the postal services based on recent data.

More precisely, the project considers using classification to solve handwritten digit recognition. To be specific, the common computer vision methods such as thresholding and Histogram Oriented Gradients based feature extraction are utilized to form a data vector for an SVM classification model. Moreover, neural networks are explored as an alternative supervised learning algorithm.

Clustering analysis shows its true power in allowing us to dig out the underlying connections among mass data and explore distinct "groups" of customers. In this project we consider using several related algorithms such as hierarchical clustering, db-scan, gaussian mixture and k-means models. The quality of our clustering strongly depends on the number and quality of observations we can gather.

Prediction is linked with the maximizing the economic prosperity of the postal office. It can be accomplished by either depending on linear or nonlinear regression models. Additionally, some statistical methods, such Bayesian linear regression could also be expected to get accurate results depending on the kind of prior distribution we choose to assume. In this project, we consider linear regression models such as Lasso, Ridge and Elastic Net methods to predict the profit of the U.S. postal services.

Finally, the association analysis aims to explore useful patterns about items that were lost together. It uses the Apriori Algorithm and the FP-Growth Algorithm to produce strong association disciplines which can be utilized to extract feature values. Then they can act as an input of the algorithm to modify the output for obtaining optimal association rules.

2 Problem Description and Benefits

The main topic of our project is about the application of *Data Mining* in postal office services. In general, there is a variety of functional departments that need to coordinate in order to form a complex postal system. Our project will deal with a large number of postal transmissions each day by recognizing mails with different addresses and zip codes. After attaining accurate mailing information, the postal system needs to classify them into different categories as well as assign the mailing routine and departure time which involves depleting many physical resources. Besides, the font of digits and characters, the underlying rules about frequency, mailing destination and contents of a certain group of customers exists but is hidden in numerous irregular mailing behavior patterns. Finding them can be vital for saving time and financial expenditure. The entire process needs optimal capital distribution and work flow which should be predicted before systematic operation.

An initial approach would be to increase the number of workforce proportionately to the complexity of tasks needed to be solved. However, the financial elements should be controlled under certain limitation. The automatic task-dealing system is an optimal way which conforms to the trend. Nevertheless, the whole system involves a lot of randomness and uncertainty by receiving mass data and information each day. It makes the vital knowledge difficult to extract and needs advanced data mining to find the path towards well-organized data and information structure. The handwritten digits in zip codes and characters with various spacing and size styles would make it hard to recognize and attain mail information accurately. Failing to attain those useful disciplines about organized the behavior of some customers makes it harder to build a targeted forecast for ideally assigning routines, resources and expenditures. As it is highlighted in the next sections, a potential prediction has a strong connection with the stock market which, in return, can impact the value and economic condition of the postal office. Undeniably, all of the variables interact

with each other hence the application of data mining for one such variable can cause a chain reaction for the whole system.

The benefits of using data mining on the postal office system are really extensive ranging from business to environment and society. For instance, the improved efficiency of recognizing handwritten codes and characters with different styles will shorten time for mailing delivery. It can help customers receive mails without long delays, make their lives easier to enjoy the mailing contents and build up better social praise from them. Accurately locating targeted groups of customers by extracting certain underlying rules about their mailing behaviors can create postal policies that specific target them thus facilitating their postal habits. These beneficiaries will spread the policies of this postal office to more people resulting in absorbing more customers and an increase of business. Moreover, data mining techniques in this setting will have a strong connection with the stock market thus allowing numerous people who hold the stocks of the postal office to benefit. What's more, the improved efficiency will lead to simplified routines network which will result in a decrease of utilization of social resources, for instance, tracks, to diminish the damage to the environment.

3 Datasets and Pre-Processing

In this project we consider many different datasets that correspond to various functions of the postal services. First, we gathered data from the USPS website that span 90 years, and include the: number of postcards, number of employees, number of mailing pieces that were handled and how many of them were stamped. We also merged this dataset with the USA GDP and population for each year. After performing EDA we filtered outliers and duplicate values. We also performed z-score normalization to our data, so that the larger instances would not dominate the sum (exact methodology is included in section 4.1).

The dataset that we considered for classification was provided to us from MNIST, which is a database that includes 60,000 handwritten digits. Since this dataset is pre-processed, EDA was unnecessary in this case.

As far as clustering is concerned, we use a dataset, which we retrieved from the USPS online data resources, and contains 50,000 users' package sending and receiving habits. Again, we perform z-score normalization to our data and discard outliers. We found that no duplicate values existed in this dataset (exact methodology is included in section 5.1).

Finally, for association analysis we experiment on the TSA claims dataset that includes the number of various stolen items per airport. The dataset contains approximately 20,000 instances and similarly to before, we clean the outliers of the dataset and normalize the attributes. We also use `OpenRefine` to locate naming inconsistencies but we found none in the dataset (exact methodology is included in section 5.2).

4 Supervised Learning

4.1 Prediction Models

In order to predict the profit of the US postal services like we suggested in Phase 1 and Phase 2, we collected relevant USPS data from 1926 to 2016 in the following format:

	Year	Postcards	Employees	profit	Stamped	GDP	Pop	Income	Pieces_of_Mail_Handled
0	1926	206000000	245769	-19884252	206000000	1	117.40	659819801	2.548353e+10
1	1927	184000000	248587	-31455503	1422000000	1	119.04	683121989	2.668656e+10
2	1928	172000000	250759	-32065845	1507000000	1	120.51	693633921	2.683700e+10
3	1929	283000000	253147	-85396070	1417000000	1	121.77	696947578	2.795155e+10
4	1930	298000000	254563	-98183121	1325000000	1	123.08	705484098	2.788782e+10

Figure 4.1: Attributes outline

First, we take a look at the distribution of the data.

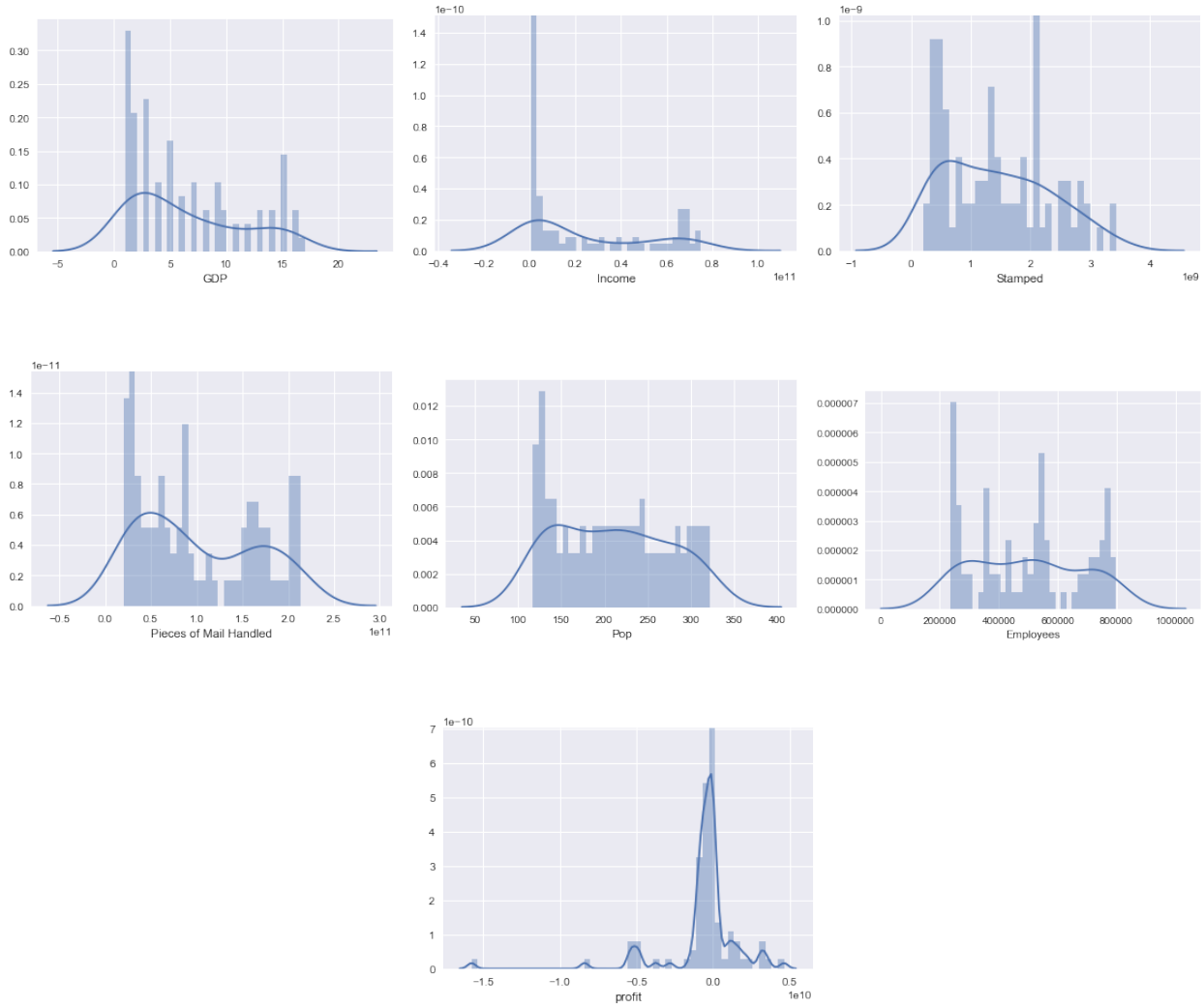


Figure 4.2: Distributions of attributes and target

And the correlations of these attributes as well:

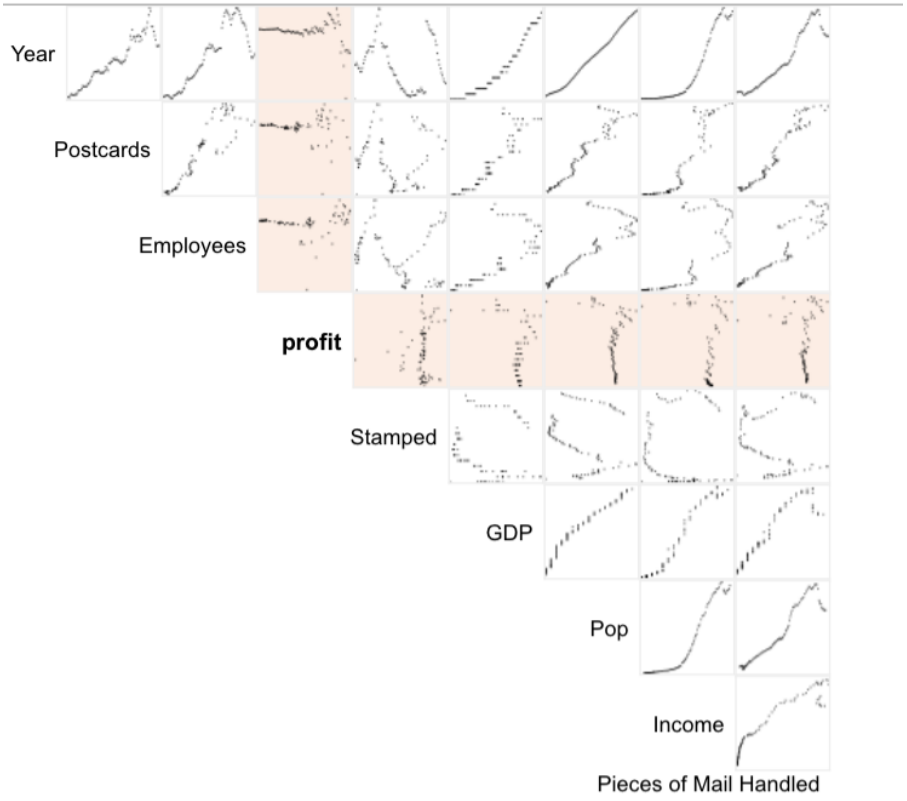


Figure 4.3: Correlations

Then we examine the mathematical features of our attributes.

	Year	Postcards	Employees	profit	Stamped	GDP	Pop	Income	Pieces of Mail Handled
count	91.00000	9.100000e+01	91.000000	9.100000e+01	9.100000e+01	91.000000	91.000000	9.100000e+01	9.100000e+01
mean	1971.00000	1.584591e+09	496281.901099	-5.899576e+08	1.439707e+09	6.417582	209.130000	2.386587e+10	1.008339e+11
std	26.41338	9.142781e+08	183858.262645	2.525584e+09	8.761949e+08	4.977875	63.619143	2.707579e+10	6.405685e+10
min	1926.00000	1.720000e+08	236472.000000	-1.574100e+10	2.060000e+08	1.000000	117.400000	5.867332e+08	1.986846e+10
25%	1948.50000	8.195000e+08	341288.000000	-7.835135e+08	5.690930e+08	2.000000	147.910000	1.491411e+09	4.191774e+10
50%	1971.00000	1.562961e+09	509145.000000	-2.056580e+08	1.325000e+09	5.000000	207.660000	8.751484e+09	8.698300e+10
75%	1993.50000	2.426912e+09	678805.500000	2.879425e+07	2.088000e+09	10.000000	261.525000	4.848270e+10	1.612785e+11
max	2016.00000	3.656291e+09	797795.000000	4.627000e+09	3.438000e+09	17.000000	321.930000	7.493200e+10	2.131380e+11

Based on our analysis, we know that there exist outliers in our attributes. Therefore for all of our attributes we define Q_0 = 5% quantile and Q_1 = 95% quantile and then take the attributes which fall inside these quantiles. Following this process, we only keep the following values:

$$1943 \leq \texttt{Year} \leq 1999$$

$$136 * 10^6 \leq \texttt{Population} \leq 279 * 10^6$$

$$32818262000 \leq \texttt{Pieces of Mail Handled} \leq 170859000000$$

$$495346000 \leq \texttt{Stamped} \leq 2206000000$$

$$273418 \leq \texttt{Employees} \leq 707485$$

$$612000000 \leq \texttt{Postcards} \leq 2561614000$$

$$2 * 10^{12} \leq \texttt{GDP} \leq 12 * 10^{12}$$

$$966227288 \leq \texttt{Income} \leq 62726000000$$

$$-942336448 \leq \texttt{Profit} \leq 287594000$$

```
data[np.abs(data.profit-data.profit.mean())>(2*data.profit.std())]
```

	Year	Postcards	Employees	profit	Stamped	GDP	Pop	Income	Pieces of Mail Handled
77	2003	2661507000	729035	4627000000	2551592000	14	290.11	68529000000	2.021850e+11
84	2010	2931565000	583908	-8374000000	1447435000	15	308.11	67052000000	1.708590e+11
86	2012	2588140000	528458	-15741000000	1157309000	15	312.76	65223000000	1.598350e+11

4.1.1 Result comparison of of Linear Regression, Ridge Regression, Lasso and ElasticNet

We first used the 8 attributes to build the prediction model, both basic linear regression model and regularized models(lasso, ridge and elnet). Their coefficients are as following:

Table 4.1: Coefficients of Linear Regression
intercept: -513061189.354

	Coefficient
Year	-1.338969e+08
Postcards	-2.936049e+00
Employees	-4.639075e+04
Stamped	-4.317481e-01
GDP	4.729469e+09
profit	4.253619e-01
Pop	-1.385664e+08
Pieces of Mail Handled	4.060289e-01

Table 4.3: Coefficients of Lasso Regression
intercept: -532875379.13

	Coefficient
Year	-2.162789e+08
Postcards	-2.905516e+00
Employees	-4.610187e+04
Stamped	-3.503691e-01
GDP	4.604996e+09
profit	4.314664e-01
Pop	-9.317486e+07
Pieces of Mail Handled	4.034347e-01

Table 4.2: Coefficients of Ridge Regression
intercept: -786067238.27

	Coefficient
Year	-2.674554e+08
Postcards	-7.149560e-01
Employees	-3.969022e+04
Stamped	5.896003e-02
GDP	4.530350e+09
profit	7.706666e-01
Pop	-1.921725e+07
Pieces of Mail Handled	3.099767e-01

Table 4.4: Coefficients of Elastic Net
intercept: -691254838.49

	Coefficient
Year	8.374675e+07
Postcards	2.978253e+00
Employees	-1.118950e+04
Stamped	2.179263e+00
GDP	2.171250e+09
profit	2.414834e-01
Pop	6.930514e+07
Pieces of Mail Handled	1.177818e-01

Next we had a comparison of their accuracy performance:

	Linear Regression	Ridge Regression	Lasso Regression	Elastic Net Regression
MAE	2018650759.3271382	2122606138.3892395	2009745791.6046386	2009739894.6248312
MSE	6.401276611096e+18	7.444501719328e+18	6.397528903768e+18	6.397475428829e+18
RMSE	2530074427.9756365	2728461419.7983232	2529333687.7067351	2529323116.7309823
RAE	0.11407334702606951	0.1199478342181633	0.1135701299794501	0.1135697967428955

Table 4.5: Accuracy comparison

To make a more intuitive comparison, we plot the error histogram of the four models below

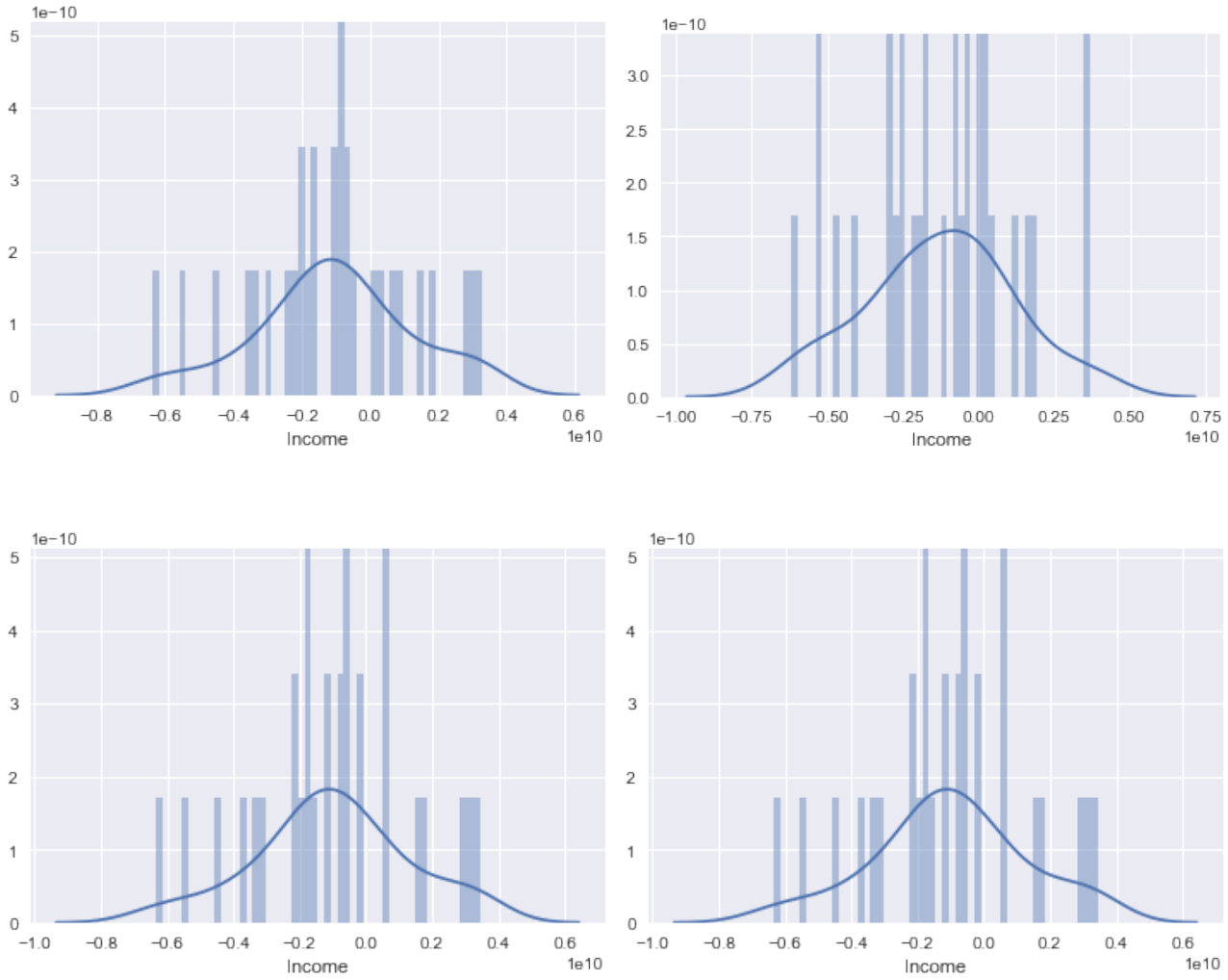


Figure 4.4: Error distributions

4.1.2 Analysis of the chosen model

As we can now compare the results of for the models using two different sets of combinations of attributes,we can safely infer that all the attributes are necessary in our case to predict the profit due to the following reasons:

- relative Error Metrics (RAE,MSE) are lower in case for all attributes compared to the second combination.
- Almost all the attributes have a positive co-relation to the Income.

By looking at tables 4.5, we see that one o the best performing linear Regression models is Ridge Regression which uses all the attributes to predict the profit. The model that we have created can be used to predict the profit of postal services based on 7 predefined attributes. The coefficients of our model tell us the mean change in the response variable (*profit*), for one unit of change in the predictor variable while holding other predictors in the model constant. Moreover, the $y-$ intercept in our model is approximately 235400000 which signifies the baseline happiness i.e. when all other factors are 0.

4.1.3 Application of Prediction to Postal Services

Our prediction model predicts the income of the US postal office for some future time using the attributes Postcards,Employees,Stamped amongst others. Knowing this information would prove extremely useful for the management because it would enable them to come up, in advance, with strategies that will be profitable

to the organization. For example, the management would be able to calculate the approximate number of workforce that would be necessary in order for the postal office to be profitable.

Predicting the income for the US postal office would be incredibly useful from many perspectives, nevertheless, choosing an appropriate model is one of the biggest difficulties for this type of problem. In this project we shown that using the Ridge Regression Model using all the attributes ,we can predict the profitability of the Postal services in the future time using the data we have collected range from years 1926 – 2016

4.1.4 Future Work

Future work will explore more complex prediction models in order to increase the accuracy of the existing models. For example, we could explore Bayesian linear regression and experiment with a variety of prior distributions. Moreover, we currently use data from the recent past in order to minimize the effect of autocorrelation in our data. In future, we should explore alternative models based on *time series*, in order to use our full dataset starting from 1789.

4.2 Classification Models

In the context of the postal services, the classification models that we explore try to predict the class of a handwritten digit represented as an image. It is important however to note, that we will not deal with breaking an image containing many digits into a sequence of separate images, each containing a single digit. We consider however, solving this segmentation problem in a future advancement of this project.

More precisely, our input data is represented as 28×28 pixel images of scanned handwritten digits and our task it to predict whether a digit is either a $0, \ldots, 9$. For this task we use the *MNIST* dataset which contains thousands of scanned images of handwritten digits, together with their correct classifications.

As the dataset is complete and preprocessed, we do not need to worry about missing attributes and extensive data cleaning. In our next step we will attempt to recognize digit patterns from this CSV model using SVM classifier. This is an example of Supervised Learning Classification.

4.2.1 Neural Networks

Before we start coding our neural network we first need to define its layers. Firstly, the input layer of the network contains the neurons that encode the values of the input pixels. Since in this problem we are dealing with 28×28 images the size of the input layer (and number of pixels) has to be $28^2 = 784$. Then we will arbitrarily choose the size of our hidden layer to be $n = 15$. Finally, since this problem contains 10 distinct classes, we include 10 neurons in our output layer. For example, we choose to classify a given handwritten digit as 4 only if the 4^{th} neuron has the highest activation value. Pictorially, the neural network is shown below in Figure 4.5.

As far as the implementation of the Neural Network is concerned, we consider using *Theano*. This would allow us fast implementation and flexibility to experiment with various loss functions and algorithms to perform gradient descent. Also in the following paragraphs **we split the dataset into** $60,000$ **testing images and** $10,000$ **test images**.

In our first attempt we build a neural network of a single hidden layer and then experiment with the hidden layer size. We use the sigmoid activation function between the input and hidden layer and then the softmax function between the hidden layer and the output layer. Since the softmax gives a distribution over our classes, we finally predict the class with the maximum activation value. Also, we set the learning rate be $\eta = 3$ and the batch size to be 10. Finally, we train our model by using *SGD* in order to minimize the *Mean Squared Error (MSE)*.

$$L = -\frac{1}{2n} \sum_x \| y(x) - t \|^2$$

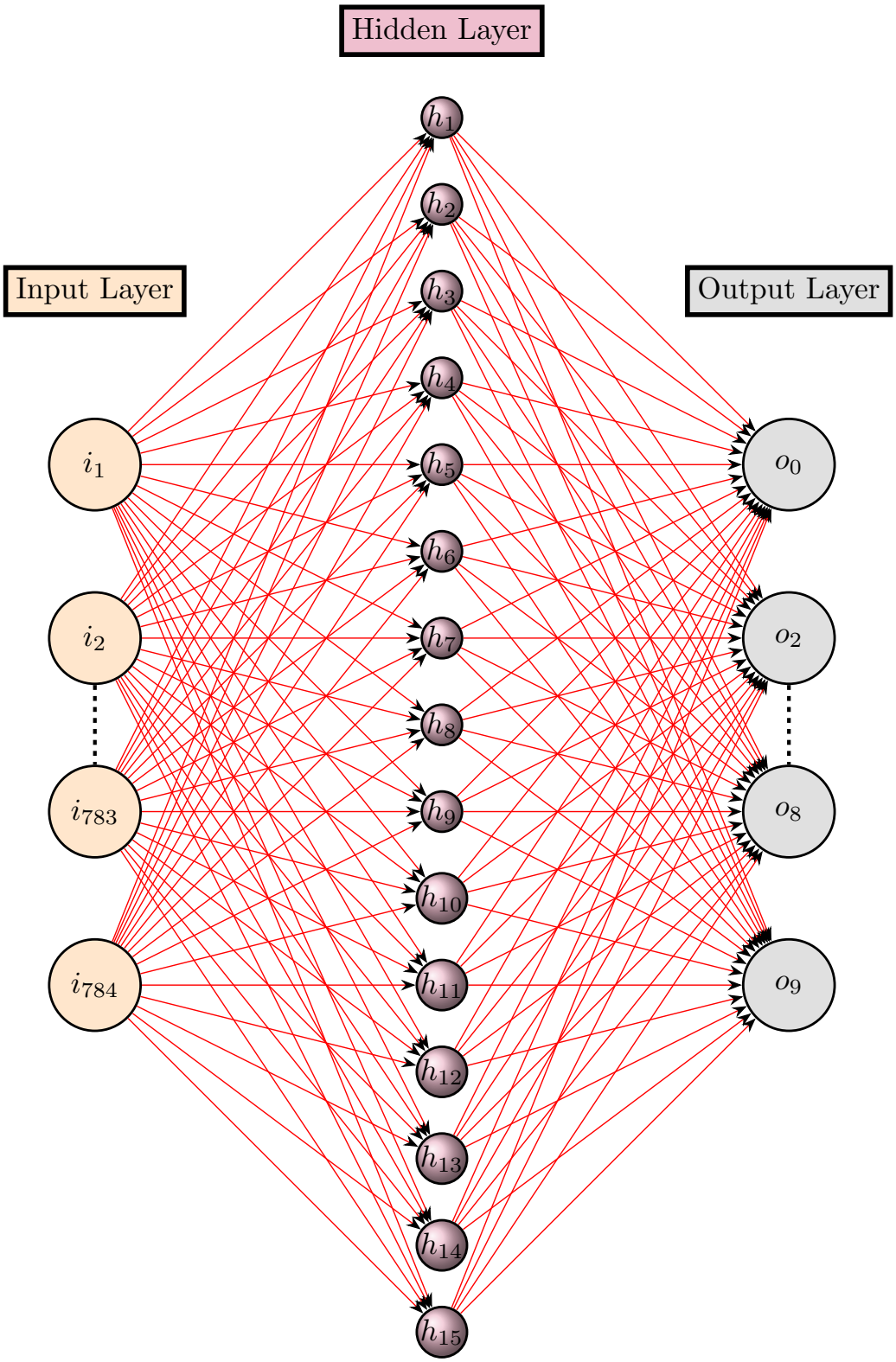


Figure 4.5: Neural Network Architecture

,where n is the number of all datapoints, y is the predicted value and t is the actual class of the datapoint x.

Our results are summarized in the following table after training for 35 epochs:

Table 4.6: Neural Network with MSE cost function

Hidden Layer	Accuracy(%)
10 neurons	91.27
20 neurons	93.50
30 neurons	95.22
40 neurons	95.35

In this case we see that as we increase the number of hidden layer neurons the accuracy also increases even though this is not generally the case due to overfitting. It is also worth mentioning, that the above results highly depend on the choise of hyperparameters η, the batch size and our cost function.

For example, in an attempt to increase the accuracy of our results , instead of using the MSE as our loss function we use the cross entropy function with regularization.

$$L = -\frac{1}{n}\sum_{x} y ln(t) + (1-y) ln(1-t) + \frac{\lambda}{2n}\sum_{w} w^2$$

,where w are the weights and the rest of the parameters are the same as before.

Moreover, in order to prevent our network from overfitting we will use **early stopping**. That is, we will split our data into train (50,000 images), test (10,000 images) and validation (10,000 images) and then we'll compute the classification accuracy on the validation data at the end of each epoch. Once the classification accuracy on the validation data has saturated, we stop training.

Our results are summarized in the following table after training for 35 epochs:

Table 4.7: Neural Network with Regularized Cross Entropy

Hidden Layer ($\lambda = 5$)	Accuracy(%)
10 neurons	92.54
20 neurons	94.89
30 neurons	96.40
40 neurons	96.71

As expected, we see that the accuracy of our model increased further.

4.2.2 SVM

To perform SVM classification we can either train the classifier using raw pixel data or apply Histogram Oriented Gradients(HOG) based feature extraction to obtain a vector abstracting the contents of the digit. The benefit of gradient extraction using HOG is better prediction accuracy for better recognition ability.

The things to select with SVM are:

- Training Image Resolution - Train Dataset
- HOG Parameters, Namely the cell size and block sizes
- The Kernel to be used by SVM

By correctly choosing all the above options we can obtain an SVM classifier whose accuracy is similar to human accuracy in classifying the digits.

To test our model and uncover patterns we use the MNIST dataset provided by Yann LeCun at http://yann.lecun.com/exdb/ The dataset provides 60000 images training images and labels and 10000 test images and labels.

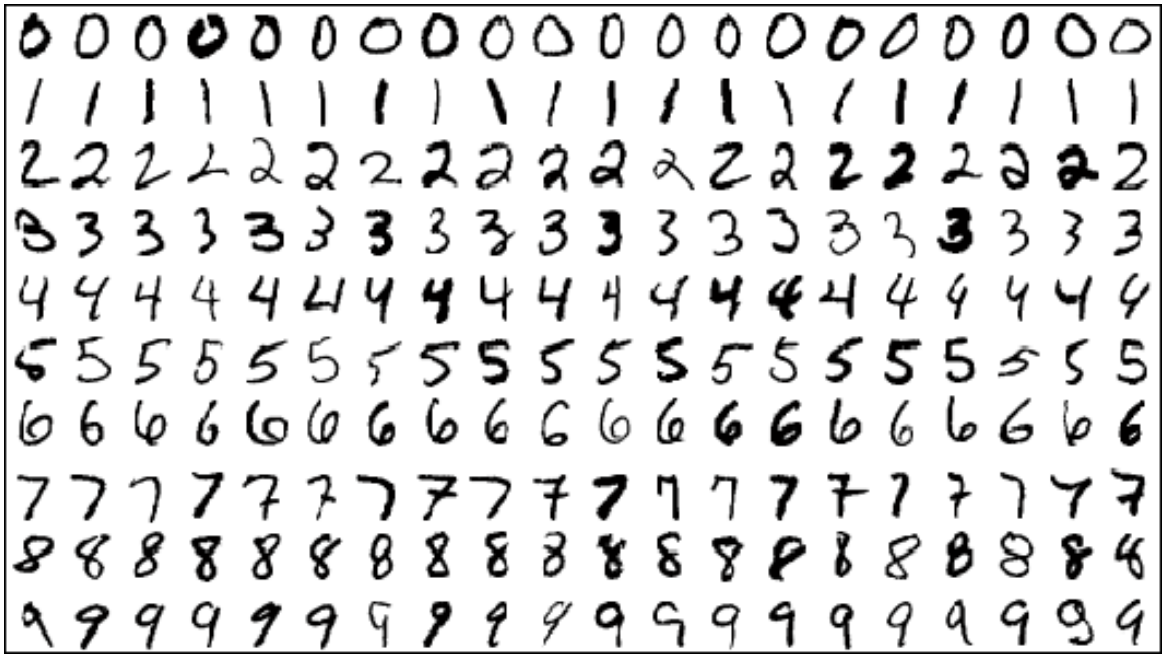
These images and labels are provided in binary format so as to minimize the size, simplify and make the task of machine learning quicker to test. We use the mnistHelper Matlab code provided by Stanford at http://ufldl.stanford.edu/wiki/index.php/Using_the_MNIST_Dataset to extract the binary image/label data into a csv file that can be easily loaded in a python program.

The format of the CSV File is:

$$[Label(0,1,2..9),\ p1\ p2\ p3\ p4\ \ldots\ldots\ P784]$$

The images provided in the training images is of size 28*28 pixels.

Let us visualise these digits to better understand their complexity:



The images are binary images with intensity values ranging from 0 - 255 with an ideal threshold value of 122. For now we let the intensity values non-thresholded. As we progress in the project we can test the results of Histogram Oriented Gradients on the images.

As the dataset is complete and preprocessed, we do not need to worry about missing attributes and extensive data cleaning. In our next step we will attempt to recognize digit patterns from this CSV model using SVM classifier. This is an example of Supervised Learning Classification.

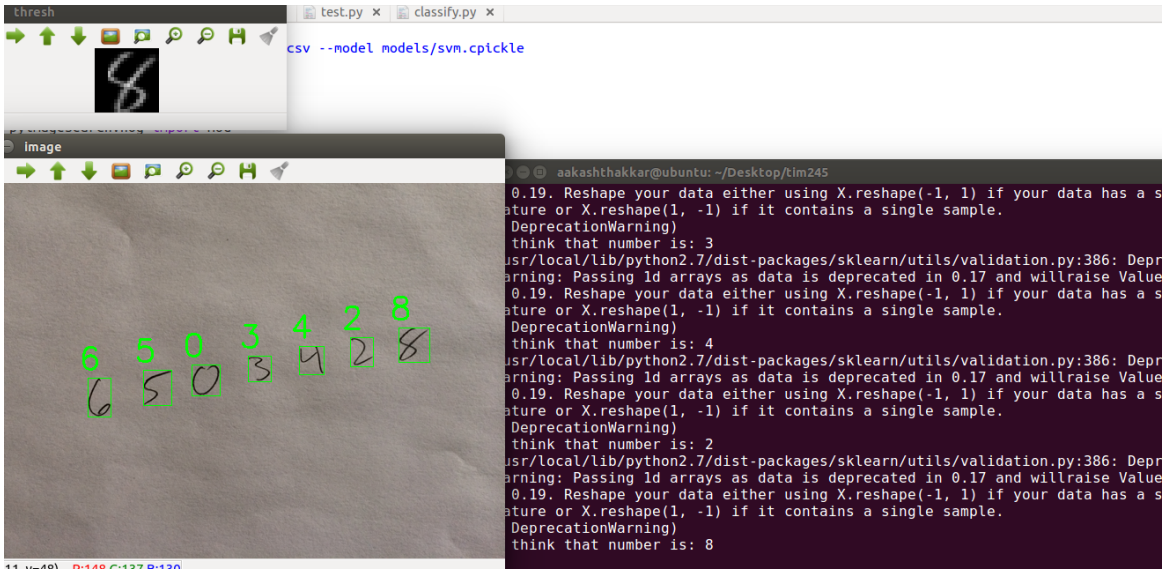
We use the following classifier with the linear kernel to measure accuracy of our experiment: http //scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC.fit.

Table 4.8: Accuracy of Classifiactions

Test Method	Accuracy(%)
Raw Pixel Data	89.5
HOG Character Vectors	92.80

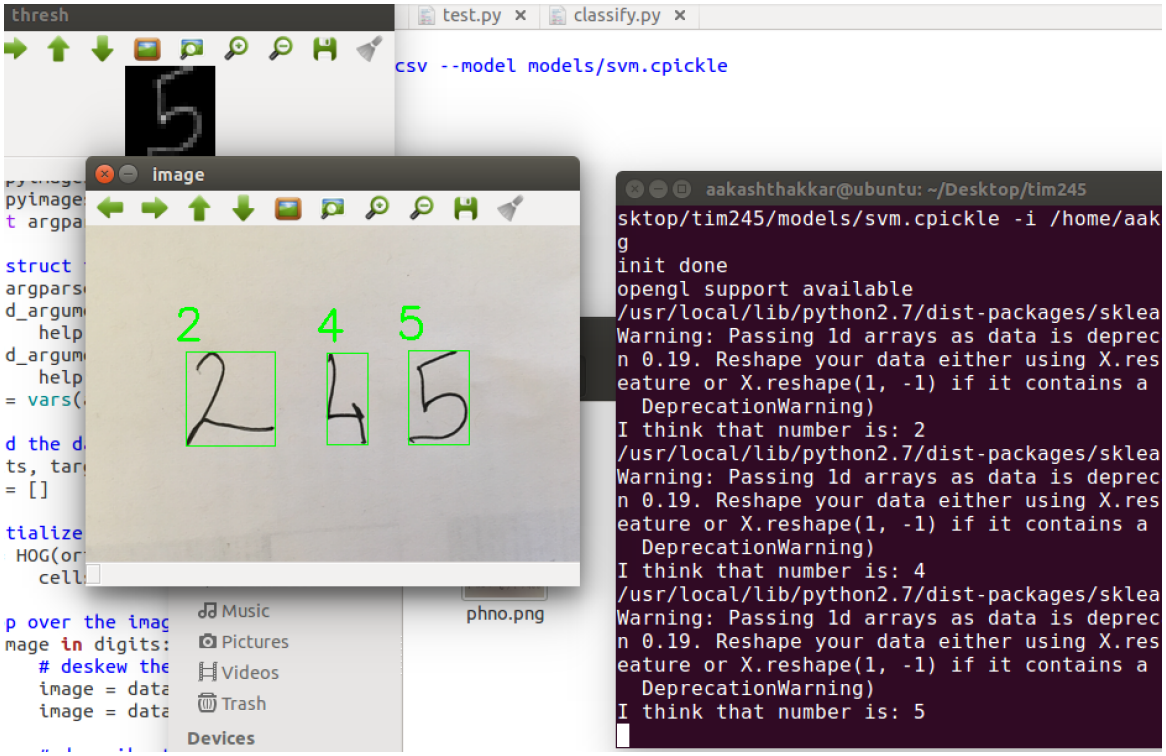
The above data is obtained on a dataset of 5000 digit data, picked randomly and cross folded with magnitude of 10 to avoid overfitting.

Figure 4.6: Test of our Python Program based on OpenCV on the standard digit recognition example image



In both the figures attached below showing handwritten digit recognition based on HOG SVM Model, We use contour detection and local thresholding to identify the location of the digits and then pass them to the classifier.

Figure 4.7: SVM Classifier on 245 digits written anew



4.2.3 Application of Handwritten Digit Recognition to Postal Services

The benefits of using data mining on the postal office system are really extensive ranging from business to environment and society. For instance, the improved efficiency of recognizing handwritten codes and characters with different styles will shorten time for mailing delivery. It can help customers receive mails

without long delays while at the same time make their lives easier to enjoy the mailing contents. Accurately classifying digits can also help the postal office save money. More precisely, the Postal Office can greatly reduce the number of employees that are performing data entry. As a result, the company can reduce its expenses by using the classifier we proposed in the previous sections.

4.2.4 Future Work

Future work will explore alternative methods of performing classification like Logistic Regression, in order to obtain a more spherical view of which model performs the best. Moreover, future work will attempt to create models that can classify arbitrary handwritten characters, including symbols and letters. By doing so, the application of such a system to the postal services will be more wide and impactful.

5 Unsupervised Learning

5.1 Clustering Models

The purpose of clustering analysis, in the context of our project is to cluster Postal Service customers into different groups based on some distinctive and important features. The data used in this project, consist of two broad parts, namely “sending packages” and “receiving packages”. More precisely, we exploit information such as how many packages a given user sent or received withing three months, how many of them were in the same state or country as well as whether a given package had insurance. Typical instances of the data are shown below:

Table 5.1: Clustering Inputs

ID	Sending	Within 3 months	Same state	Different state	Different country	Insurance	Receiving	Receiving within 3 months
1	10	0	8	2	0	1	4	1
2	4	1	3	1	0	0	7	5
3	29	11	2	27	0	0	4	1

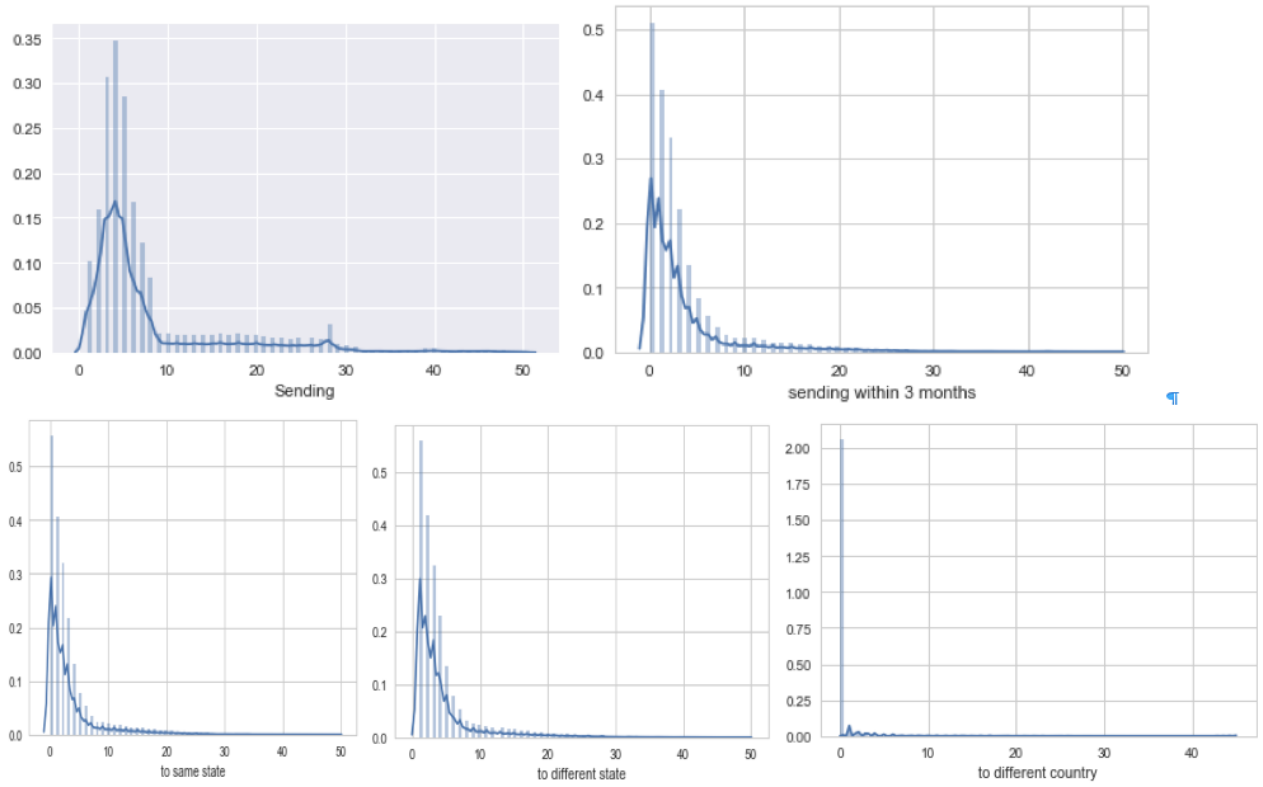
In the table above columns “Same state", “Different state", “ Different country" show how many times a given person sends a package to a particular location. Columns “sending within 3 months" and “receiving within 3 months", represent the total number of sending/receiving packages for a particular customer. Finally, the “insurance" is a Boolean parameter, with “1" representing the customer has bought an insurance for the item he sent, and “0" representing that he never bought insurance. We expect that these data can to some extent reflect the potential type of a customer.

As with every other dataset, we first filter outliers before proceeding with our analysis. For each attribute, discard outliers which are more that two standard deviations from the mean.

The following chart is an overview of the dataset:

	ID	Sending	sending within 3 months	to same state	to different state	to different country	insurance	receiving	receiving within 3 months
count	49317.00000	49317.000000	49317.000000	49317.000000	49317.000000	49317.000000	49317.000000	49317.000000	49317.000000
mean	24659.00000	8.193382	3.592919	3.413529	4.428088	0.351765	0.097532	8.153983	3.590344
std	14236.73595	8.630442	5.505970	5.390209	5.339523	2.035244	0.296684	8.574360	5.468590
min	1.00000	1.000000	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000	0.000000
25%	12330.00000	3.000000	0.000000	0.000000	1.000000	0.000000	0.000000	3.000000	1.000000
50%	24659.00000	5.000000	2.000000	2.000000	3.000000	0.000000	0.000000	5.000000	2.000000
75%	36988.00000	8.000000	4.000000	4.000000	5.000000	0.000000	0.000000	8.000000	4.000000
max	49317.00000	50.000000	49.000000	49.000000	49.000000	45.000000	1.000000	50.000000	48.000000

The following distributing plots show us information of some important parameters, and indicate that the attributes in the dataset are approximately normally distributed:



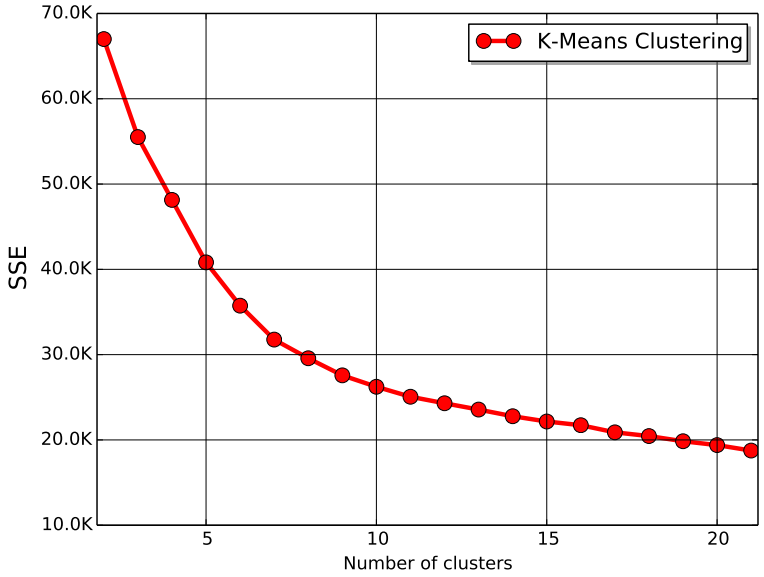


Figure 5.1: Sum of squared errors as we vary the number of clusters in the K-Means algorithm.

In the following paragraphs we explore clustering methods such as *K-Means, Agglomerative Hierarchical clustering, DB-Scan and Expected Maximization* in order to find useful patterns in our dataset. Since Agglomerative Hierarchical clustering and DB-Scan perform many complex operations, working on the full $50K$ dataset was impossible using a commercial Intel-Core i7-5600 running at 2.6 GHz. Therefore, for all subsequent models of this section, we sampled 50% of the original dataset.

5.1.1 KMeans

The first clustering approach that we explore is *K-Means*, which, is a fast and simple approach to clustering which works well for globular clusters.

Using the "elbow method" we speculate that a useful clustering occurs for a number of clusters between 2 and 8 (Fig. 5.1). In order to assess our model we compute the *silhouette* score. But in order to compute this

score, we first need to define quantities a_i, b_i corresponding at point i of our dataset. More precisely, let the set $\mathscr{D}$ represent the set of all data points and let $C_1 \dots C_k$ be the sequence of clusters, with $|C_j|$ being the total number of points in $C_j, 1 \leq j \leq k$. Then, for a point $i \in \mathscr{D}$, let $i \in C_j$. We define:

$$a_i = \frac{1}{|C_j|} \sum_{e \in C_j} D(i,e)$$

, where $D(x,y)$ is an arbitrary distance function with $D(x,x) = 0$. Intuitively, a_i is the average distance of i to all other points contained in the same cluster C.

Now let:

$$b_i = \min_w \left\{ \frac{1}{|C_w|} \sum_{e \in C_w} D(i,e) \right\}_{w=1, w \neq j}^{k}$$

Finally we can compute the silhouette score as:

$$silhouette = \frac{1}{n} \sum_{i=1}^{n} \frac{b_i - a_i}{\max(a_i, b_i)}$$

Given the above definitions, we see that computing the silhouette score, even for the fastest clustering algorithm would be very computationally expensive. Nevertheless, by restricting ourselves to using only a fraction of the total data we avoid this problem.

After experimenting with **20** different cluster sizes for the $K-Means$ algorithm we see that the number of clusters that achieves the highest silhouette score is 2 (Table 5.2).

Table 5.2: Reporting the most high scoring number of clusters for $K-Means$ after trying all clusters sizes between 1 and 20

Number of Clusters	Silhouette
2	**0.47**
5	0.41
3	0.39

5.1.2 Hierarchical Agglomerative Clustering

Next we experiment with *Agglomerative Clustering*. More precisely, for all possible linkage methods *Single, Complete, Average*, we compute the silhouette score for all cluster numbers between 1 and 20. The most high scoring number of clusters are reported in Tables 5.3, 5.4, 5.6.

Table 5.3: Reporting the most high scoring number of clusters for **Single** Linkage Hierarchical clustering after trying all clusters sizes between 1 and 20

Number of Clusters	Single Linkage Silhouette
5	**0.41**
2	0.40
4	0.40

Using the tables we see that the best performing linkage methods are *Average* and *Complete*. For both methods, the number of clusters with highest silhouette score is 2. This is also made evident by Figure 5.6 where we see that if we let distance be larger than 228 the resulting clusters are 2.

Table 5.4: Reporting the most high scoring number of clusters for **Complete** Linkage Hierarchical clustering after trying all clusters sizes between 1 and 20

Number of Clusters	Complete Linkage Silhouette
2	**0.70**
3	0.67
4	0.50

Table 5.5: Reporting the most high scoring number of clusters for **Average** Linkage Hierarchical clustering after trying all clusters sizes between 1 and 20

Number of Clusters	Average Linkage Silhouette
2	**0.75**
3	0.64
4	0.59

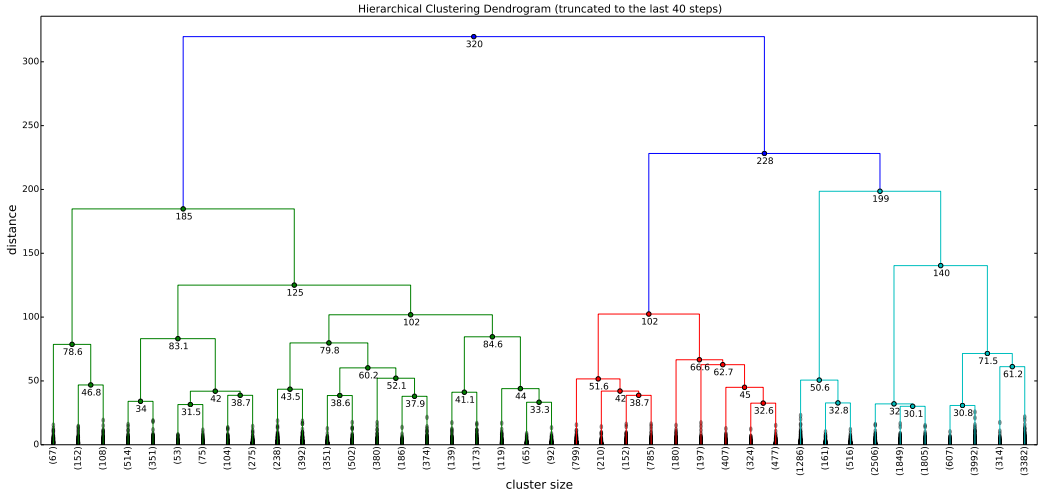


Figure 5.2: Dendrogram for Average Linkage

5.1.3 DB-Scan

The next clustering method that we explore, is *DB-Scan*. In contrast to the previously seen clustering algorithms, in this case we do not need to supply the number of clusters, as the algorithm finds the number of clusters based on ϵ and $MinPoints$. Heuristically, we found that letting $MinPoints > 10$ always resulted in the same clustering where only one cluster was present. Therefore, we fix $MinPoints = 10$ and let the values of ϵ vary. More precisely, we let $\epsilon = 0.2 + 0.1t$ for $t = 0, 1, \ldots, 34$.

Table 5.6: Reporting the most high scoring ϵ and number of clusters after varying ϵ.

ϵ	Average Linkage Silhouette
3.5 (clusters = 1)	**0.72**
3.4 (clusters = 1)	0.71
0.8 (clusters = 2)	0.35

5.1.4 Expectation Maximization

Finally, we try the *Expectation-Maximization* (EM) clustering algorithm.

The EM technique is similar to the K-Means technique. The basic operation of K-Means clustering algorithms is relatively simple: Given a fixed number of k clusters, assign observations to those clusters so that the mean across clusters (for all variables) are as different from each other as possible. The EM algorithm extends this basic approach to clustering in two important ways:

1. Instead of assigning examples to clusters to maximize the differences in the mean of continuous variables, the EM clustering algorithm computes probabilities of cluster membership based on one or more probability distributions. The goal of the clustering algorithm then is to maximize the overall probability or likelihood of the data, given the (final) clusters.

2. Unlike the classic implementation of k-Means clustering, the general EM algorithm can be applied to both continuous and categorical variables (note that the classic k-means algorithm can also be modified to accommodate categorical variables).

We used the simple EM class in Weka to perform our EM clustering. Keeping all parameters set to default, we get the following results:

Table 5.7: Reporting the most high scoring number of clusters for Expected Maximization clustering after trying all clusters sizes between 1 and 20

Number of Clusters	Average Silhouette
10	**0.08**
3	0.07
11	0.05

	Cluster						
Attribute	0	1	2	3	4	5	6
	(0.07)	(0.02)	(0.14)	(0.04)	(0.53)	(0.14)	(0.05)
Sending							
mean	4.3508	21.2882	17.5661	10.2403	4.2616	4.1505	32.761
std. dev.	1.9929	9.2197	6.7126	6.2033	1.8792	1.797	8.7357
sending within 3 months							
mean	1.6327	10.3216	7.8983	4.3575	1.5875	1.5422	17.6648
std. dev.	1.6028	8.3286	5.6759	3.9775	1.5464	1.494	10.2384
to same state							
mean	1.6173	9.8158	7.7204	4.2321	1.5928	1.5411	14.8676
std. dev.	1.6417	8.3677	6.2612	4.4006	1.5879	1.5498	11.5394
to different state							
mean	2.6728	10.3006	9.0188	5.4754	2.5947	2.5553	15.1843
std. dev.	1.6675	8.0829	6.4816	4.7711	1.5859	1.5353	11.1339
to different country							
mean	0.0607	1.1718	0.8268	0.5328	0.0741	0.0541	2.7091
std. dev.	0.3126	3.6578	2.3572	1.3869	0.3644	0.2854	7.0863
insurance							
mean	1	1	0	0.0413	0	0	0.0004
std. dev.	0.2967	0.2967	0	0.199	0	0.2967	0.0201
receiving							
mean	7.7067	7.4956	4.6987	30.384	4.2481	19.2998	10.7317
std. dev.	7.888	7.3202	2.584	11.0864	1.8676	7.6039	9.4877

The number of iterations performed is 2,and the Log-likelihood is -9.19203.

Finally, by looking at table 5.7 we see that this clustering method performs very poorly, since the silhouette score is an order of magnitude less than the other clustering methods discussed in this section.

5.1.5 Application of Clustering in Postal Services

We observe that in the majority of the models we used in the previous section, the cluster size with the highest silhouette score is 2. We believe that this cluster size has a very natural explanation given our dataset. More precisely, in the dataset that we used for clustering, we see that there are two broad categories of people, namely: those who mostly *receive* packages and those who, mostly *send* packages. Knowing to which cluster a person belongs, can prove crucial for the Postal Service company. For example, the company can directly target the costumer with more appealing specialized offers thus increasing its profit. More precisely, if a person belongs in the "sender" cluster, then the company can give this costumer offers that allow his packages to be delivered in less time, whereas, if a person belongs in the "receiver" category, then the company can offer this costumer unique deals for storing the packages for an extended period of time.

5.1.6 Future Work

In our current clustering models, we only manage to produce 2 clusters with a very natural meaning, namely: the "senders" and the "receivers" of packages. Future work, should explore how to create even more useful clusters based on other costumer characteristics. Moreover, in future work we consider exploring more advanced clustering methods that would allow us to incorporate our full dataset. Finally, after interesting clustering methods are produced we plan to consult domain experts in order to obtain a human evaluation of our results.

5.2 Association Analysis Models

Association Analysis, while being used in the Postal Services could also be extended to the case for HomeLand Security handling TSA claims of the lost goods/baggages at each Airport.

Each airport receives many such claimed/unclaimed items. If for each airport, it is possible to decide which type of items generally have a higher chance of being lost together, it would facilitate airport officials to handle similar categories of items.

For example, assume that *currency* and *expensive* items like *jewelry* and *watches* have a higher tendency to be lost together at any respective airport compared to other items. Then, the airport administration could designate more storage room of such similar items, in this case, by having vaults. Nevertheless, if Computer and Accessories and Personal Electronics are likely to occur more frequently together, then airport officials could instead have specialized larger storage areas for those items. We aim to streamline such a process by looking at the occurrence of similar products which will help the airport officials to account for more and better related storage facilities.

This process of discovering relationships in data is called Association Analysis and for finding out frequent patterns, we can use Association Rules. In order to find useful association rules we consider using the efficient Apriori Algorithm which allows us to generate strong association rules from the frequent item-set.

For this task we use the TSA Claims Dataset which was initially quite crude to our requirements. We first had to clean the data in terms removing many unnecessary attributes as well as outliers. For each attribute, discard outliers which are more than two standard deviations from the mean.

Here,we have Airport Codes and corresponding to them are the Categories Items that are reported to have been found at the Airport premises.Eg:Currency,Clothing,Jewelry and Watches.

Following are the input variables to our model:

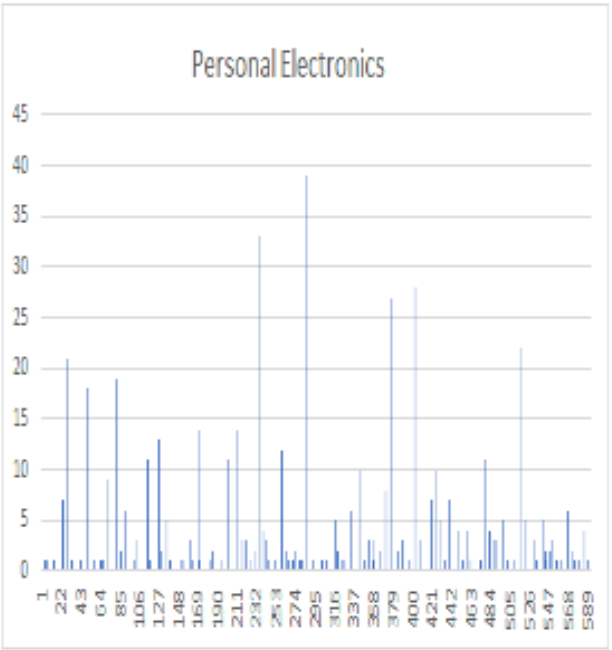
1. Currency

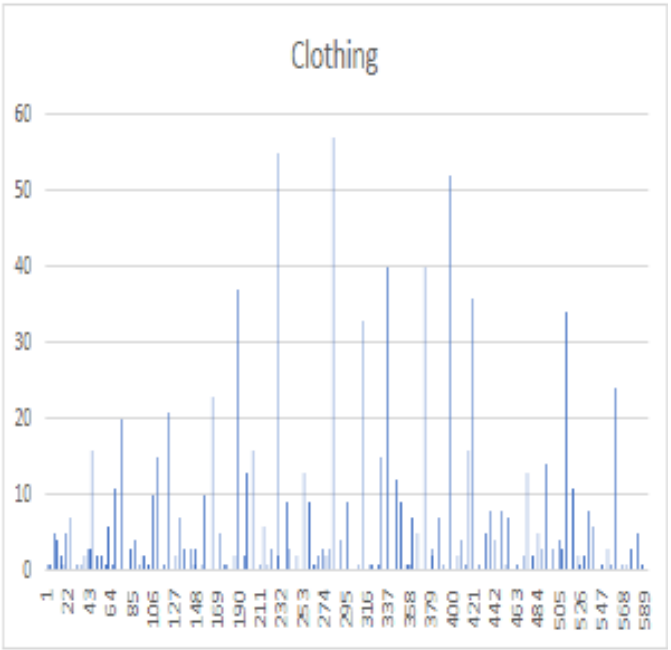
2. Clothing
3. Jewelry & Watches
4. Computer & Accessories
5. Personal Electronics

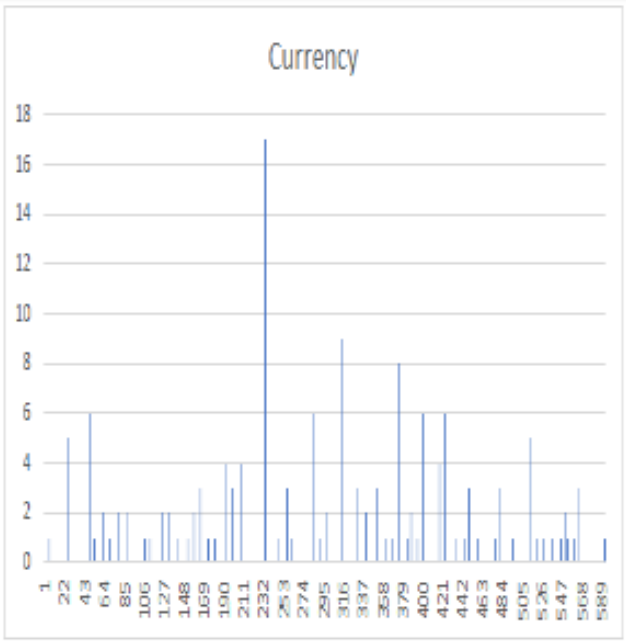
According to preliminary observation, it can be deduced that Clothing and Computer & Accessories have the highest chance of occurring:

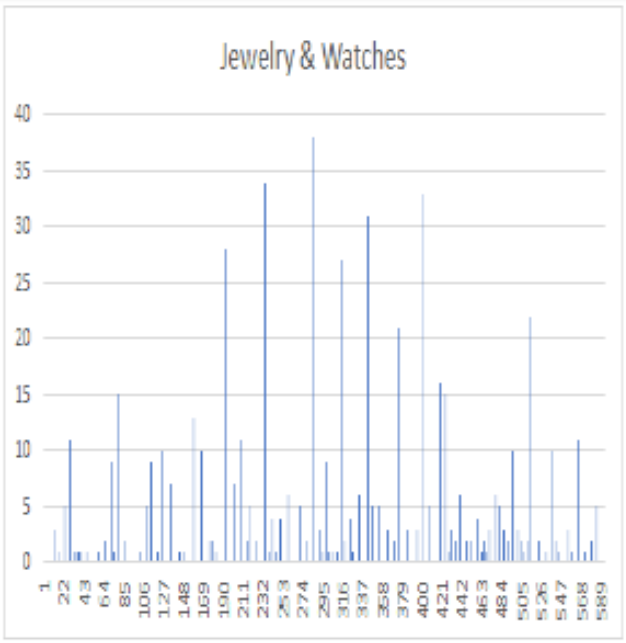
Measure	Clothing	Jewelry & Watches	Currency	Computer & Accessories	Personal Electronics
Mean	7.2977099	5.851485149	2.62068966	6.421487603	4.827272727
Standard Err	0.9682387	0.786065039	0.3554328	0.923652474	0.66370948
Median	3	3	2	2	2
Mode	1	1	1	1	1
Standard De	11.081998	7.89985587	2.70689559	10.16017722	6.961043751
Sample Vari	122.81069	62.40772277	7.32728373	103.2292011	48.45613011
Kurtosis	7.7606783	5.778678602	13.5490136	11.54498631	8.427737783
Skewness	2.7492511	2.447429272	3.16649	3.138049707	2.784614984
Range	56	38	16	60	38
Minimum	1	0	1	1	1
Maximum	57	38	17	61	39

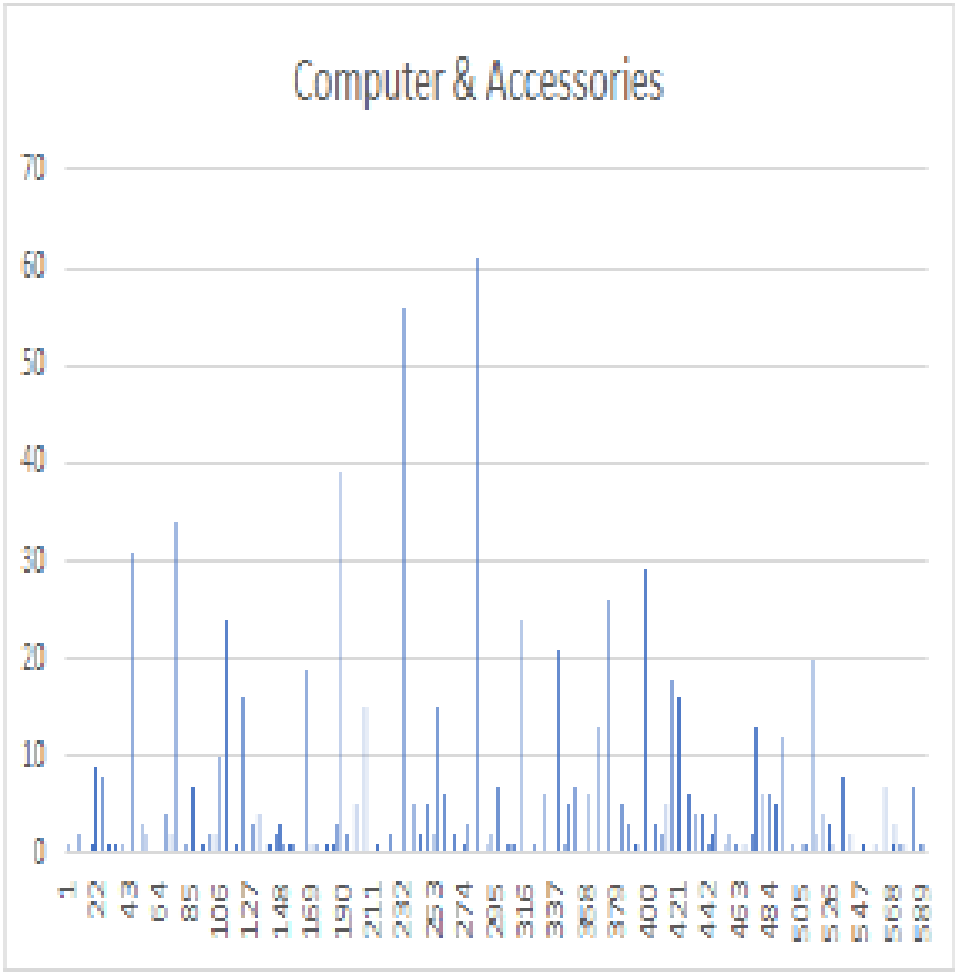
The following distributions give an indication as to how each of the attributes are distributed in terms of each Airport(x axis being given by the airport serial number and y axis being given by The quantity)






Currency
18
16
14
12
10
8
6
4
2
0
1
22
43
64
85
106
127
148
169
190
211
232
253
274
295
316
337
358
379
400
421
442
463
484
505
526
547
568
589


Jewelry & Watches
40
35
30
25
20
15
10
5
0
1
22
43
64
85
106
127
148
169
190
211
232
253
274
295
316
337
358
379
400
421
442
463
484
505
526
547
568
589



There are 2 metrics for calculating association analysis:

1. Support
2. Confidence

Support (sometimes called frequency) is simply a probability that a randomly chosen transaction t contains both itemsets A and B:

$$support(A \Rightarrow B)_t = P(A \subset t \wedge B \subset t) = \frac{\#\text{ of transactions containing both } A \text{ and } B}{\text{total }\#\text{ of transactions}}$$

Confidence (sometimes called accuracy) is simply a probability that an itemset B is purchased in a randomly chosen transaction t given that the itemset A is purchased. Mathematically,

$$confidence(A \Rightarrow B)_t = P(B \subset t \mid A \subset t) \frac{\#\text{ of transactions containing both } A \text{ and } B}{\text{total \# of transactions containing A}}$$

We carried out association analysis on our dataset using 2 algorithms:

5.2.1 Apriori Algorithm:

Apriori uses a breadth-first search strategy to count the support of itemsets and uses a candidate generation function which exploits the downward closure property of support. Using apriori algorithm and by changing the support and the confidence values, we got the following results.

confidence	support	best rules	lift
0.9	0.1	Cloths=More than 100 17 ==> Currency=less than 100 17 Jewelry & Watches=zero Computer & Accessories=less than 100 Personal=less than 100 19 ==> Currency=zero 18	2.78
0.8	0.2	Jewelry & Watches=less than 100 Currency=zero Computer & Accessories=less than 100 41 ==> Cloths=less than 100 35 Jewelry & Watches=zero 47 ==> Currency=zero 40	1.45
1	0.1	Cloths=More than 100 17 ==> Currency=less than 100 17	2.78
0.6	0.1	Cloths=less than 100 98 ==> Personal=less than 100 74	1.19

As we can see the lift for the rule in case of confidence value of 0.6 indicates that the occurrence of clothes less than 100 and personal items less than 100 is not fluke and have a good chance of occurring together.

The best rule amongst all the set of rules is for confidence 1 :

Cloths=More than 100 17 ==> Currency=less than 100 17

This rule has a lift value of 2.78 which indicates that the association did not occur by accident and it has a good chance of occurring.

Thus, in this way the Airport officials can infer which category and what quantity of items are likely to be found in their airport premises and accordingly they can handle their storage.Passengers who lose their baggage in the form of clothes are also likely to lose currency as well.

5.2.2 FP-Growth Algorithm:

The FP-Growth Algorithm is an efficient and scalable method for mining the complete set of frequent patterns by pattern fragment growth, using an extended prefix-tree structure for storing compressed and crucial information about frequent patterns named frequent-pattern tree (FP-tree).
This algorithm works as follows: first it compresses the input database creating an FP-tree instance to represent frequent items. After this first step it divides the compressed database into a set of conditional databases, each one associated with one frequent pattern. Finally, each such database is mined separately. Using this strategy, the FP-Growth reduces the search costs looking for short patterns recursively and then concatenating them in the long frequent patterns, offering good selectivity.

The constraint of the FP growth algorithm is that it works only on binary data so our first goal will be to binarize the entire dataset. As the dataset used has Numeric data, we use Weka to convert from Numberic to Nominal and then Nominal to Binary and end up with the following dataset.

Based on the above dataset, we obtain following rules using FP-Growth Algorithm:
Analyzing the top 3 rules show:

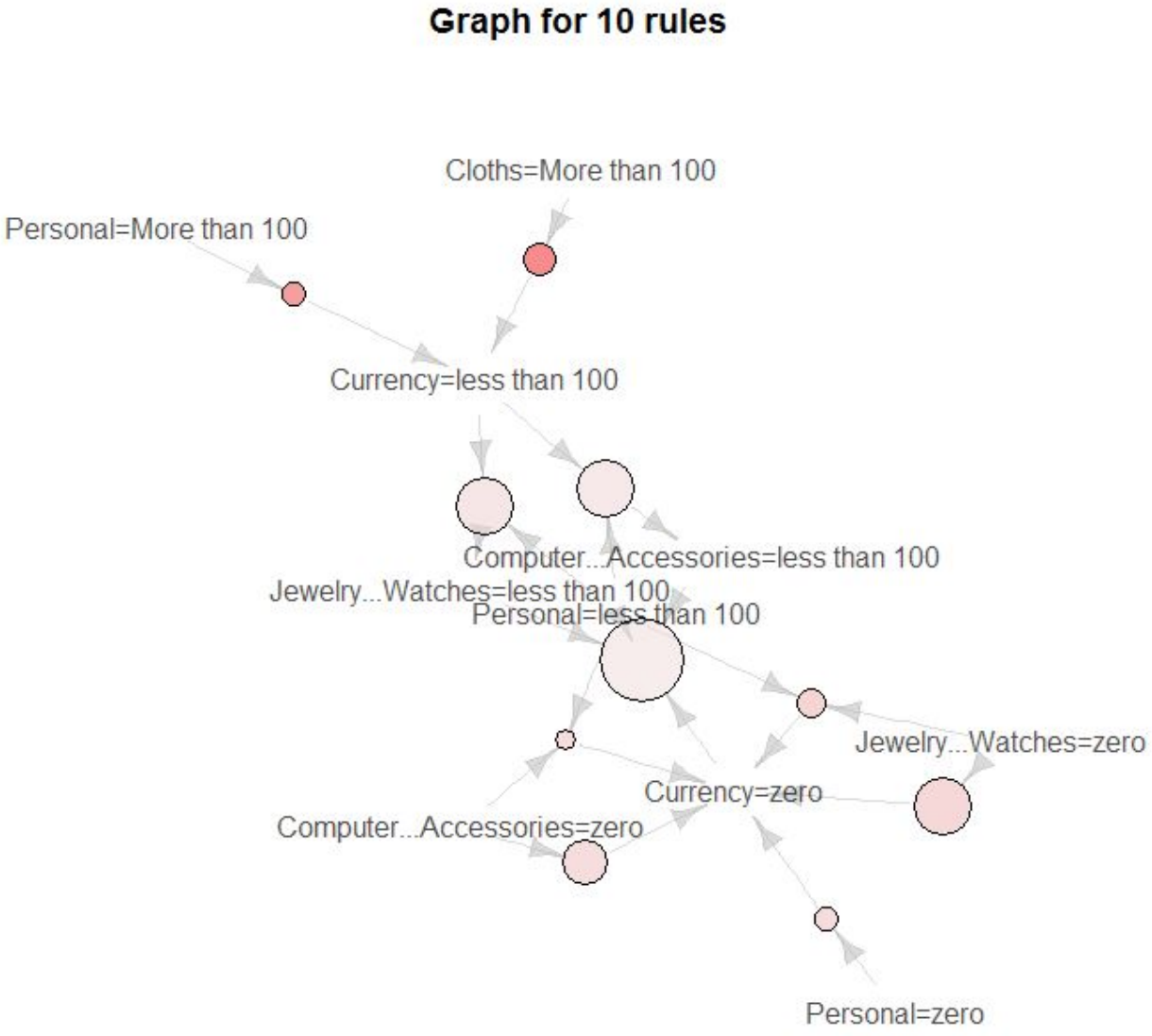


Figure 5.3: Association between different set of attributes and rules.

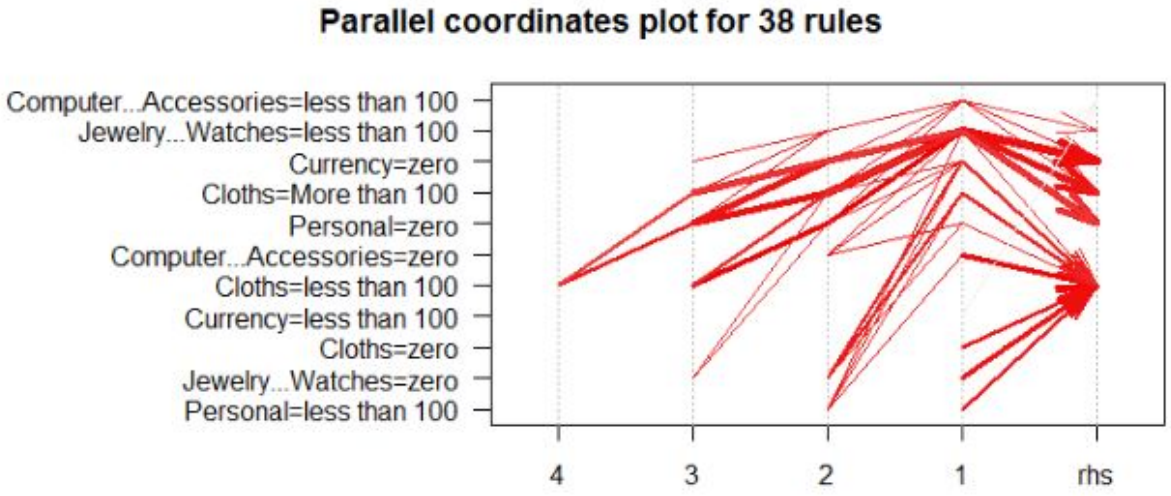


Figure 5.4: The above graph gives the overall high level view of all the rules and attributes in parallel.

1. Jewelry & Watches=zero, Computer & Accessories=zero, Cloths=less than 100 => Currencyžero, conf:(0.97), lift:(1.24), lev:(0.03), conv:(4.33)

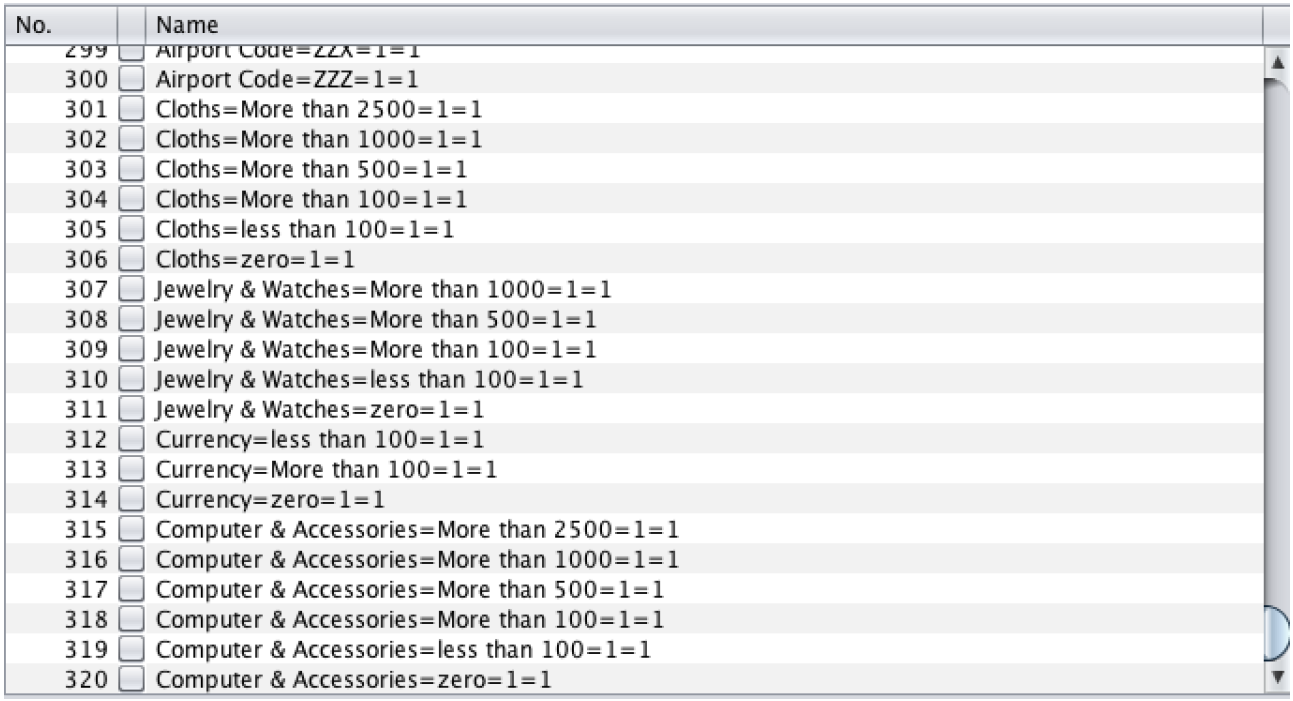


Figure 5.5: Modified Dataset for FP-Growth Assoc

```
Scheme:       weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1
Relation:     association_analysis_weka1-weka.filters.unsupervised.attribute.Remove-R2-weka.filters.unsupervised.attribute.NominalToBinary-Rfirst-last-weka.filters.unsup
Instances:    300
Attributes:   320
              [list of attributes omitted]
=== Associator model (full training set) ===

FPGrowth found 11 rules (displaying top 10)

 1. [Jewelry & Watches=zero=1=1=1, Computer & Accessories=zero=1=1=1, Cloths=less than 100=1=1=1]: 40 ==> [Currency=zero=1=1=1]: 39   <conf:(0.97)> lift:(1.24) lev:(0.03
 2. [Jewelry & Watches=zero=1=1=1, Computer & Accessories=zero=1=1=1]: 127 ==> [Currency=zero=1=1=1]: 121   <conf:(0.95)> lift:(1.22) lev:(0.07) conv:(3.93)
 3. [Jewelry & Watches=zero=1=1=1, Cloths=less than 100=1=1=1]: 58 ==> [Currency=zero=1=1=1]: 55   <conf:(0.95)> lift:(1.21) lev:(0.03) conv:(3.14)
 4. [Jewelry & Watches=zero=1=1=1, Computer & Accessories=zero=1=1=1, Cloths=zero=1=1=1]: 87 ==> [Currency=zero=1=1=1]: 82   <conf:(0.94)> lift:(1.2) lev:(0.05) conv:(3.
 5. [Jewelry & Watches=zero=1=1=1]: 169 ==> [Currency=zero=1=1=1]: 159   <conf:(0.94)> lift:(1.2) lev:(0.09) conv:(3.33)
 6. [Jewelry & Watches=zero=1=1=1, Cloths=zero=1=1=1]: 111 ==> [Currency=zero=1=1=1]: 104   <conf:(0.94)> lift:(1.2) lev:(0.06) conv:(3.01)
 7. [Computer & Accessories=zero=1=1=1, Cloths=zero=1=1=1]: 111 ==> [Currency=zero=1=1=1]: 104   <conf:(0.94)> lift:(1.2) lev:(0.06) conv:(3.01)
 8. [Cloths=zero=1=1=1]: 142 ==> [Currency=zero=1=1=1]: 133   <conf:(0.94)> lift:(1.2) lev:(0.07) conv:(3.08)
 9. [Computer & Accessories=zero=1=1=1]: 166 ==> [Currency=zero=1=1=1]: 155   <conf:(0.93)> lift:(1.19) lev:(0.08) conv:(3)
10. [Computer & Accessories=zero=1=1=1, Cloths=less than 100=1=1=1]: 55 ==> [Currency=zero=1=1=1]: 51   <conf:(0.93)> lift:(1.18) lev:(0.03) conv:(2.38)
```

Figure 5.6: Top 10 Rules Obtained by FP-Growth Algorithm

2. Jewelry & Watches=zero, Computer & Accessories=zero => Currency=zero, conf:(0.95), lift:(1.22), lev:(0.07), conv:(3.93)

3. Jewelry & Watches=zero, Cloths=less than 100: => Currency=zero, conf:(0.95), lift:(1.21), lev:(0.03), conv:(3.14)

All rules are dominated by data entries with attribute values of zero or low values, by having richer domain data that we can clean and process better. Also multiple levels of cleaning and application of FP-Growth will show better results.

5.2.3 Application of Association Analysis in Postal Service

Thus,using association analysis using Apriori and FP-Growth algorithms,we could infer that each airport can decide which type of items generally have a higher chance of being lost at the airport. Knowing this information would be crucial to the airport authorities, since it would be easier to to handle similar categories of items together in terms of storage and logistics.

5.2.4 Future Work

Future work will explore alternative methods to evaluate the interestingness of the association rules. Currently we are using the $Lift$ metric:

$$Lift = \frac{c(\{X\} \rightarrow \{Y\})}{s(\{Y\})} = \frac{\sigma(\{X\}, \{Y\})}{\sigma(\{X\})\sigma(\{Y\})}$$

, which addresses the problems of considering only the confidence and support, by taking into consideration the support of {Y}. From this formulation it is clear that a $Lift$ value equal to 1 implies the statistical independence of {X}, {Y}. Therefore, in our models we attempt to choose rules with $Lift \gg 1$.

In future, we will use other metrics such as *Correlation Analysis, IS Measure* etc. This will allow us to obtain a larger variety in the results we produce and also discover more interesting patterns in our dataset.

6 Conclusion

This project has explored how machine learning and data mining techniques can be applied in the context of postal services. We created both supervised and unsupervised models that can help key functional components of the postal services. Future work will focus on elaborating further these models in order to make them even more accurate and useful. As far as supervised learning is concerned we, consider implementing a time series model and a deep neural network in order to predict the future profit and classify digits more accurately, respectively. Finally, we consider assessing the clusters and patterns that we obtained with the aid of domain experts.

7 Appendix

The appendix follows in the next few pages and we have included all the work that was done during the previous Phases 1-4.