Course Name: Machine Learning For Engineers

Course Number And Section: 14:332:445

Title: Term Project Phase III Report

Instructor:

Professor Waheed Bajwa

Date Submitted:

12/23/2019

Submitted by:

Team Name: “90 Degree Gang”

Alexander Ameri
Rizwan Chowdhury
Phurushotham Shekar

ABSTRACT

The Project report is on the exploration of three different types of datasets using machine learning techniques. The types of datasets consist of time-series, text, and images. The project data sets include “S&P500 Stock Data”(1), “Whiting Oil Price”(2), “Federal Interest Rates”(3), “USD Daily Gold Prices”(4), “Daily Wheat Price”(5), “Department of Justice 2009-2018 Press Releases”[6], and Fashion-MNIST (14). Each dataset has an objective to be met with machine learning. For each dataset the pre-processing, feature extraction, and machine learning algorithm are explored. Finally, whether or not the objective is met and what machine learning algorithm is the best for the objective is discussed.

INTRODUCTION

This Machine learning project goal is to explore different machine learning techniques and use them to analyze three particular sets of datasets that are of types time-series, text, and images. The time series dataset includes “S&P500 Stock Data”(1), “Whiting Oil Price”(2), “Federal Interest Rates”(3), “USD Daily Gold Prices”(4), “Daily Wheat Price”(5). The objective for the analysis of the time series data was to use the prices of all considered stocks and commodities over the past 5 days to determine if the prices of each stock or commodity would rise or fall on the next day, i.e. this was a binary classification problem. The text dataset is “Department of Justice 2009-2018 Press Releases”(6). The objective with the text dataset (10) is to perform cluster analysis to obtain prominent themes, that way one can see the crimes and affairs that have been prominent in America and the dealings of the Justice Department. The image dataset was the fashion-MNIST (14) data set. The goal for this data was to correctly classify articles of clothing. Throughout the report the pre-processing steps, feature extraction, and the three machine learning algorithm per dataset as well as results are discussed.

DATA SET & MACHINE LEARNING ALGORITHM EXPLANATION

Text Dataset

Data:

The text cluster analysis takes place with the "Department of Justice 2009-2018 Press Releases" dataset from kaggle.com. This dataset consists of a series of 13087 press releases in a JSON Lines or newline-delimited JSON format. Each line of the json file is a single press release sample and contains date of the release, the content of the release, and a topics element which contains information for a very few numbers of samples. For cluster analysis, only the contents element of each sample is used. Since cluster analysis is unsupervised and only a few samples have topics, the topics are ignored during our process. The release dates are also ignored because there is no interest in when the press releases took place.

Pre-Processing:

The pre-processing step for the text dataset transforms the raw samples into a usable format that can be clustered. Samples are loaded by reading the JSON file line by line and obtaining specifically the "contents" section which contain the text. For each sample, Python's string function are used to make all words lowercase and remove punctuation. Then each sample has its stopwords, common words with no meaningful information, removed. The Natural Learning Toolkit(nltk) library was used to obtain a list of stopwords. Afterwords, each word in each sample is stemmed using the nltk library's stemming function.

Feature Extraction:

Text features extraction takes place through bag of words model and Term Frequency-Inverse Document Frequency (TF-IDF) model. Bag of Words model obtains all the words in a series of documents and the frequency of each term in each document (6). TF-IDF model provides terms with a weight of importance based

on the number of times each term appears in a document relative to all that document's words multiplied by the log of how many times the term appears throughout the series of documents (7).

Machine Learning Algorithms:

The cluster analysis takes place through three different clustering algorithms: K-Means, Mean-Shift, and Gaussian Mixture Model with Expectation Maximization. The K-Means implementation is created from scratch while the other two are implemented through Sci-Kit Learn functions.

K-Means places samples in different clusters, initially randomly, then calculates a centroid for the cluster from its members. Then for each sample, a Euclidian distance is obtained from every centroid, then the sample is reassigned to whichever cluster had the least distance. This process is repeated until there are no reassignments. Assigning each sample to the closest centroid repeatedly eventually gathers similar samples together forming clusters. The above steps are executed through loops and numpy functions in python for the text dataset clustering. The complexity for K-Means is O(n) or linear time (8).

Mean-Shift obtains clusters by finding the gradient of the kernel density function/ probability density for the samples provided (9). The gradient is found for each sample and from that the mean shift vector is found, the sample is then moved in the direction of the vector (9). The repetition of this process leads to clusters forming. The text-dataset cluster analysis takes place using "sklearn.cluster.MeanShift". This function uses a flat kernel and it was supplied 0.95 for bandwidth. The complexity for Mean Shift is the greatest of the three methods as it has a complexity of O(n^2), which makes it very slow for larger datasets (8).

The Gaussian Mixture Model with Expectation Maximization (GMM) is implemented thorough "sklearn.mixture.GaussianMixture". The function works like

K-Means, however, uses a Gaussian probabilistic and maximizes probability instead of minimizing distance (10). Sample are put into clusters and then mean and covariance matrix calculated for that cluster (10). Then for each sample, the probability is calculated for each of the cluster and the sample is reassigned to highest probability (10). These steps are repeated until no more reassignments. For the text cluster analysis, the GaussianMixture object initialization takes place with 10 or 25 clusters and the means attribute is used once the data is fitted. Like K-Means the GMM method is linear time complexity O(n) (8).

Time Series Datasets

Data / Preprocessing:

Since the structure of each time series dataset is different, some boilerplate getter code needed to be written to retrieve the value of a commodity or stock on a given day. In addition, each dataset had to be pruned for entries where the value was either empty or zero, in which case the date was removed from the analysis. Also, the S&P500 dataset was pruned of any companies that were not members of the Fortune 500 for the entire time period under consideration, which resulted in a final company count of 470.

The value of the 5 commodities, together with these 470 stocks, on a single day is considered to be a rudimentary independent feature vector, a column vector of dimension 475. Some utility code was written to concatenate the appropriate values together for each day under consideration to create a numpy 2D array, which had 475 rows (for 475 features for that day) by 1259 columns (for every day in the time period under consideration).

Then, for each day, rudimentary feature vectors of the previous 5 days were concatenated together along a common column. This resulted in a final independent feature column vector matrix of 2375 rows (for 2375 features) by 1254 columns (for each day under consideration, minus 5 days for the initial lag).

A matrix of result (dependent) vectors was also created, which was of dimension 475 rows by 1254 columns. If the value of a stock or commodity rose between two days, the value of the corresponding dimension of that vector was “1”, else the value was “-1”. In other words, this is a binary classification scheme.

Feature Learning:

This is a very large feature space, and the curse of dimensionality becomes apparent. Therefore, Principal Component Analysis (PCA) feature learning is in order.

The goal of PCA is dimensionality reduction, so as to reduce computational cost. This is done by projecting data points which live in 2375 – dimensional space onto a hyperplane which "floats" in 2375 – dimensional space, but itself is of much lower dimensionality. The goal is to find out what the principal components (i.e. the components of this hyperplane's normal vector) of this hyperplane are, such that as much of the variance (or "energy") of the data is captured as possible, and project all 1254 of the 2375 dimensional independent features on to this plane.

In order to do this, we must first center the input vectors about their mean. Then, we take the singular value decomposition of the resulting matrix. The column vectors of the right singular matrix which result from this operation are the principle components of the hyperplane, and the square of the corresponding singular values in the singular value matrix represents the variance that that component captures.

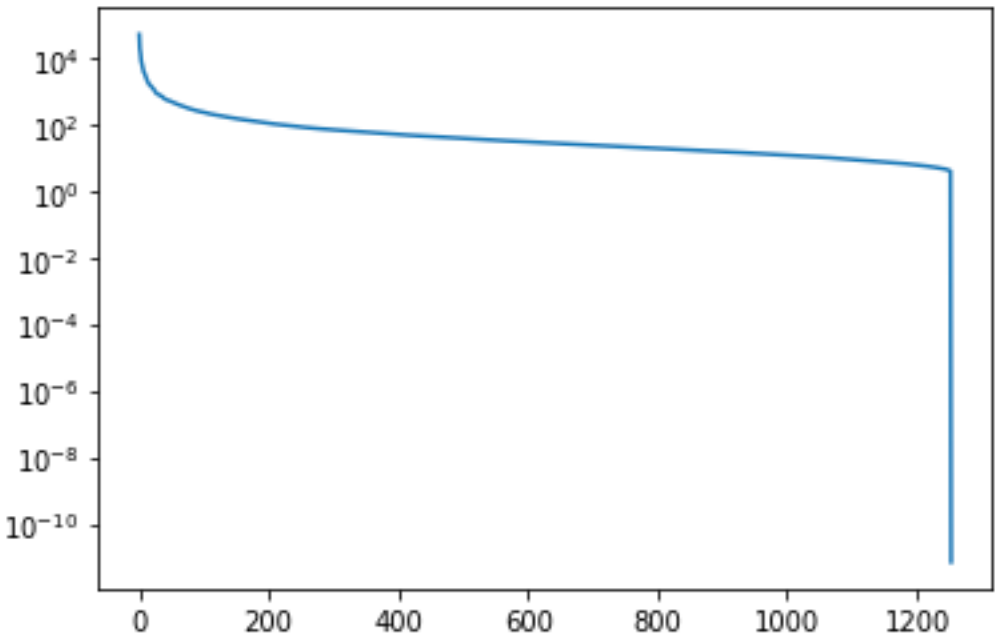
$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$$

M is the original data matrix, sigma are the singular values, and V* contains the principle components.

Scikit – learn provides a function in which we can specify the total energy we desire to capture, and will perform the PCA transform (projection) on the input data for us. This function was used with an energy capture of 99.9%, which resulted in a

subspace hyper plane of 118 dimensions down from 2375. A Scree plot of this can be seen in figure 2A below.

Figure 2A: Logarithmic Scree Plot for PCA of Stock Data



Linear Discriminant Analysis (LDA):

The first algorithm applied to the data here is Linear Discriminant Analysis. LDA is a classification algorithm which assumes that the PDF of the independent features for each class is multivariate Gaussian (118 – dimensional Gaussian, in this case), with each class data having a different mean, but the same covariance matrix.

For each pair of classes, the algorithm finds a hyperplane in 118 dimensional (in this case) upon which, given these assumptions, the probability of all points in the span of the plane belonging to either of the two classes is the same. This is the "decision boundary" of the two classes.

$$\log \hat{\pi}_k - \frac{1}{2}\hat{\mu}_k^T \hat{\mathbf{\Sigma}}^{-1}\hat{\mu}_k + x^T \hat{\mathbf{\Sigma}}^{-1}\hat{\mu}_k = \log \hat{\pi}_\ell - \frac{1}{2}\hat{\mu}_\ell^T \hat{\mathbf{\Sigma}}^{-1}\hat{\mu}_\ell + x^T \hat{\mathbf{\Sigma}}^{-1}\hat{\mu}_\ell$$

$\boldsymbol{\pi_k}$ represents the sample prior probability for each class,
$\boldsymbol{\mu_l}$ represents the sample mean of each class,
$\boldsymbol{\Sigma}$ represents the covariance matrix of the data.

x^T represents the normal vector of the decision hyperplane.

In this study, scikit-learn's DiscriminantAnalysis.LDA() class was used. This algorithm is $O(N^3)$ fast.

Quadratic Discriminant Analysis (QDA):

Quadratic Discriminant Analysis is very similar to LDA, except that it does not assume that the covariance matrices of the data for each class are the same.

$$\delta_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) + \log\pi_k$$

The above equation describes the probability of a data point belonging to a specific class k. For each pair of classes, the following equation is solved for x, as with LDA, to define a quadratic hyperplane decision boundary. The main difference here is that the convariance matrix, $\boldsymbol{\Sigma}$, can be different for each class.

In this study, scikit-learn's DiscriminantAnalysis.QDA() class was used. This algorithm is $O(N^3)$ fast.

Logistic Regression (Logit):

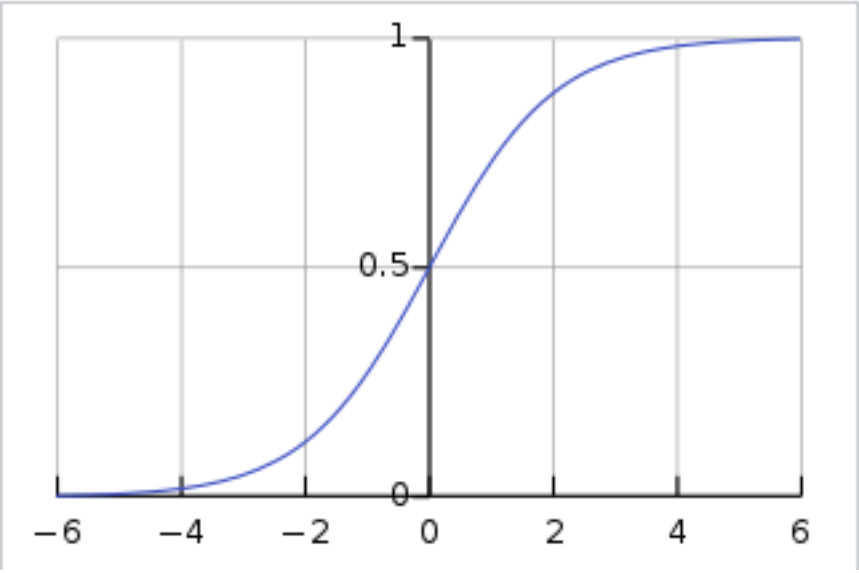
The goal of logistic regression is to find a hyperplane in 118 dimensional space (in this case) to serve as a decision boundary that minimizes the following loss function, called "logistic loss":

$$\frac{1}{1+e^{-f(\mathbf{x})}}$$

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b,$$

In which w represents the principle components of the hyperplane decision boundary, and b represents the biases (axis intercepts) of this hyperplane. Another way of looking at this is that one must find the normal distance between each datapoint and the hyperplane, and the penalty / loss incurred by this point (depending on whether it is misclassified) is given by plugging that normal distance into the x axis of the logistic function, show in figure 2B below.

Figure 2B: Logistic Loss Function (Inverted About Y Axis)



Since there is no closed – form solution to finding this hyperplane, iterative numerical methods must be used. One common way of doing this is by using gradient descent:

1. An initial guess for w and b is created.
2. The gradient of the loss function at this point is evaluated
3. The values of the guesses are moved in the direction of this gradient by some small "step" size
4. Steps 1-3 are repeated until a solution is converged to.

In this study, scikit – learn was used to do this. Scikit learn uses the LAPACK suite of numerical solvers to perform gradient descent. This procedure is computationally expensive, and takes much longer to solve than the other two methods considered in this study, even though the other methods are $O(N^3)$ time complex.

Image Dataset

The image dataset was formatted very similar to the MNIST dataset with 60,000 training samples and 10,000 test samples. Each image was 28x28, for a total of 784 pixels, and each pixel had a value between 0 and 255.

Feature Learning - PCA

Using PCA we were able to reduce the number of features from 784 to 84 while maintaining 90% of the energy.

Feature Learning - Image Resampling:

In our case, we "downsampled" the image to reduce the dimension from 28x28 to 7x7 which reduced the features from 784 to 49. Multiple methods of image downsampling were tested such as bilinear interpolation, bicubic interpolation, pixel area relations, and lanczos.

Bilinear interpolation applies a linear formula when calculating the gradient and differences between neighboring pixels and the derivatives found aren't continuous over pixel boundaries so the resulting image has "sharper" boundaries for objects and won't contain as much nuanced data that may be spread over many pixels. It's more likely to see sharper edges and changes in color between neighboring pixels.

Bicubic interpolation, on the other hand, uses a quadratic formula so the changes visible between neighboring pixels aren't as stark and you can see more details carried over from the original image.

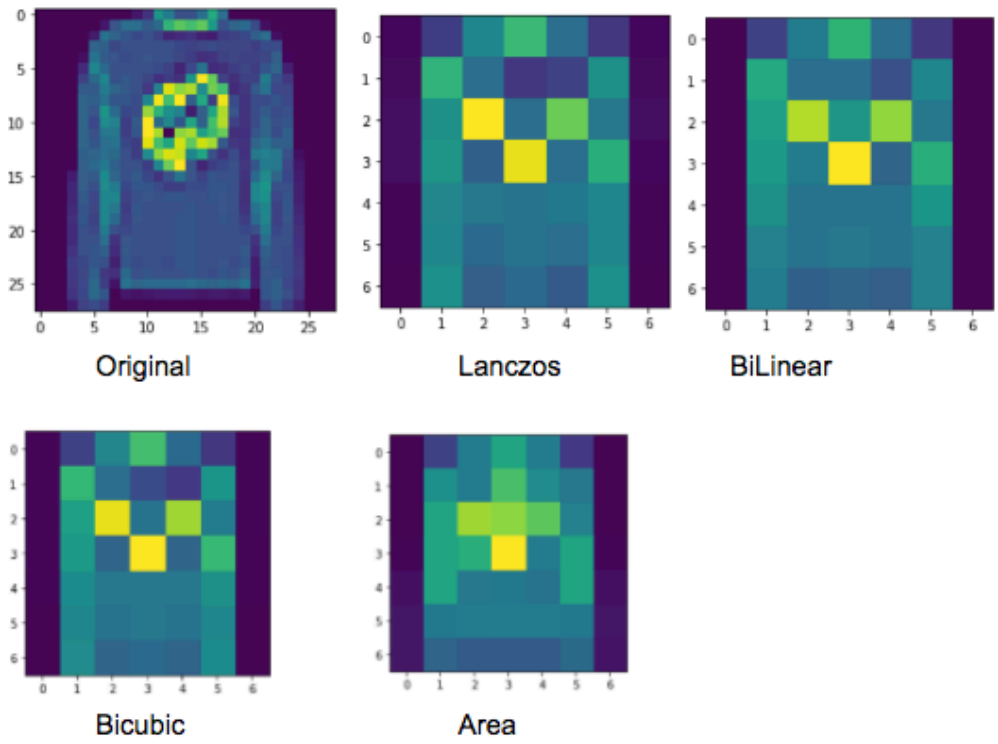
Area interpolation (INTER_AREA) is interpolation that works very similar to Bilinear Interpolation, however there is more exact with its pixel bit choices

(slightly more computationally complex). This method was selected since it provided the most correct classificatoins.

Lanczos interpolation tries to find the greatest differences in pixel changes and places more weight on that while shrinking the image.

Below (Figure 2D) are comparisons of each of the methods on the same image:

Figure 2D: Comparison Of Image Down sampling Methods



The three machine learning algorithms used on this dataset were Linear Discriminant Analysis, Quadratic Discriminant Analysis, and K Nearest Neighbor. The results can be seen in the Discussion section below.

RESULTS

Time Series Dataset Results:

Figure 3A: Mean LDA Model Predictive Accuracy For Fortune 100 Stocks:

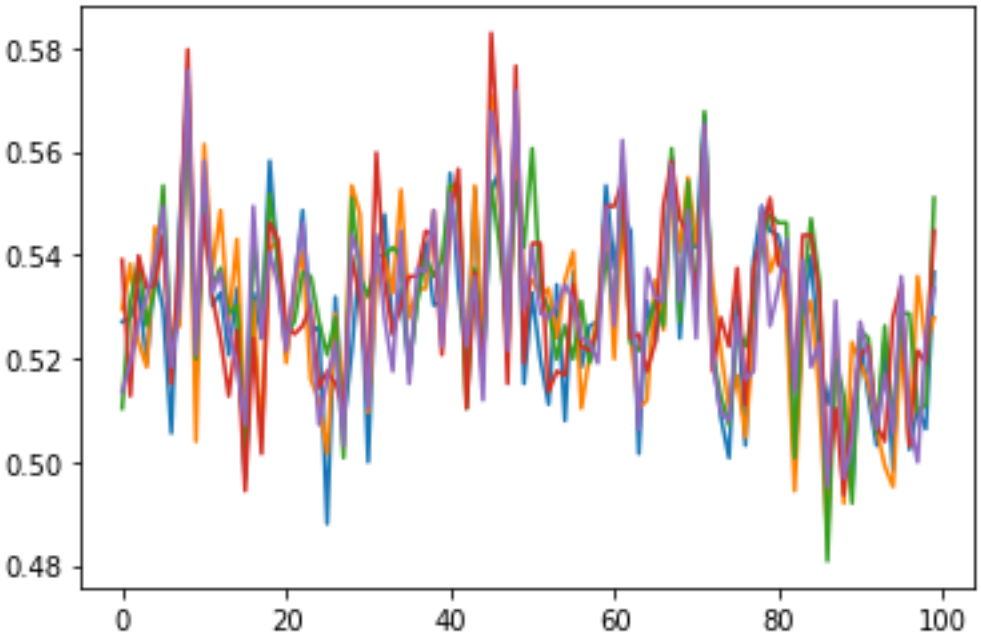


Table 3A: LDA Mean Overall Accuracy For Each Run:

K – Fold Cross Validation Run #	Mean Accuracy of All Folds Across All Companies
1	52.83%
2	52.93%
3	53.11%
4	53.06%
5	52.95%

Table 3B: Mean Confusion Matrix For LDA, All Runs, All Companies, Un-normalized:

	Actual Rise	Actual Fall
Predicted Rise	26.753	31.977
Predicted Fall	26.907	39.763

Figure 3B: Mean QDA Model Predictive Accuracy For Fortune 100 Stocks:

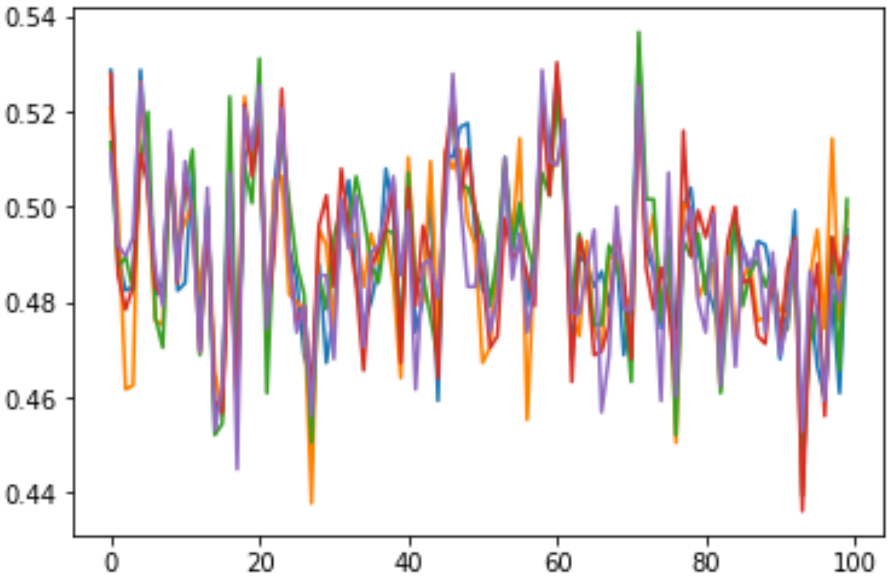


Table 3C: QDA Mean Overall Accuracy For Each Run:

K – Fold Cross Validation Run #	Mean Accuracy of All Folds Across All Companies
1	48.94%
2	48.87%
3	48.96%
4	48.90%
5	48.86%

Table 3D: Mean Confusion Matrix For QDA, All Runs, All Companies, Un-normalized:

	Actual Rise	Actual Fall
Predicted Rise	27.481	31.249
Predicted Fall	32.94	33.73

Figure 3C: Mean Logistic Regression Model Predictive Accuracy For Fortune 100 Stocks:

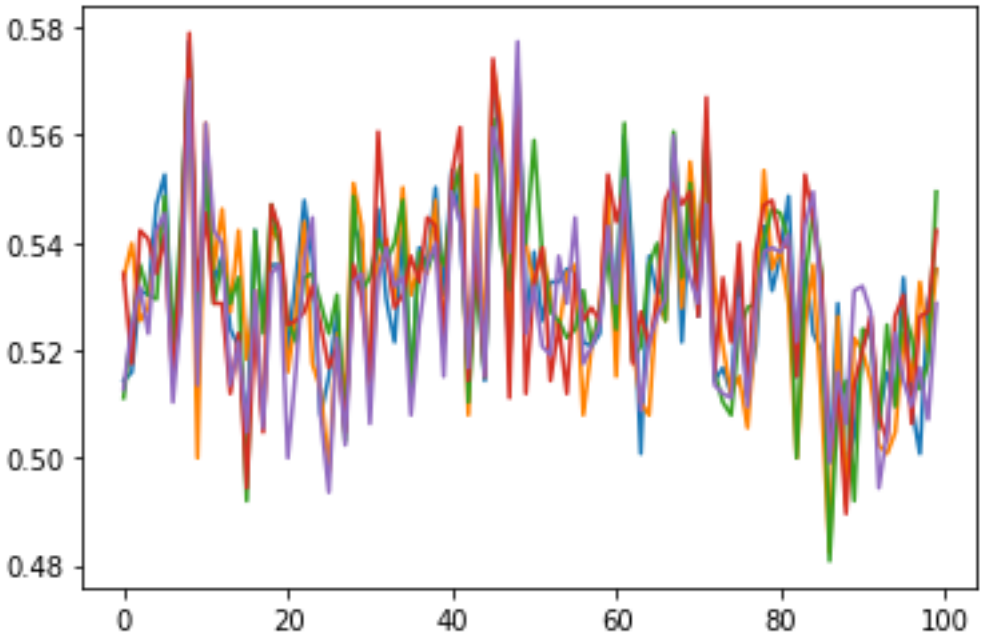


Table 3E: Logistic Regression Mean Overall Accuracy For Each Run:

K – Fold Cross Validation Run #	Mean Accuracy of All Folds Across All Companies
1	53.0 %
2	52.92%
3	53.10%
4	53.11%
5	52.78%

Table 3F: Mean Confusion Matrix For Logistic Regression, All Runs, All Companies, Un-normalized:

	Actual Rise	Actual Fall
Predicted Rise	26.887	31.843
Predicted Fall	27.2	39.47

Text Dataset Results:

Figure 3D: K Means Results

```
KMeans:
['attorney', 'us', 'district', 'crimin', 'sentenc', 'guilti', 'offic', 'admit', 'plead', 'fbi']
['attorney', 'law', 'depart', 'commun', 'justic', 'gener', 'school', 'nation', 'offic', 'program']
['settlement', 'epa', 'water', 'environment', 'air', 'natur', 'resourc', 'consent', 'requir', 'act']
['bribe', 'offici', 'corrupt', 'compani', 'briberi', 'payment', 'foreign', 'crimin', 'contract', 'launder']
['fda', 'drug', 'food', 'manufactur', 'product', 'complaint', 'distribut', 'consum', 'health', 'injunct']
['right', 'civil', 'offic', 'polic', 'law', 'attorney', 'victim', 'violat', 'us', 'divis']
['gang', 'member', 'murder', 'texa', 'aka', 'polic', 'firearm', 'indict', 'drug', 'counti']
['export', 'us', 'secur', 'militari', 'nation', 'defens', 'control', 'illeg', 'attorney', 'compani']
['tax', 'ir', 'incom', 'account', 'bank', 'return', 'attorney', 'file', 'us', 'fail']
['contract', 'armi', 'us', 'defens', 'crimin', 'investig', 'fraud', 'govern', 'briberi', 'compani']
['card', 'credit', 'comput', 'us', 'use', 'account', 'bank', 'attorney', 'crimin', 'district']
['tax', 'refund', 'ir', 'fals', 'return', 'file', 'claim', 'fraudul', 'incom', 'attorney']
['tax', 'ident', 'refund', 'theft', 'alabama', 'stolen', 'return', 'ir', 'check', 'attorney']
['fraud', 'financi', 'invest', 'us', 'scheme', 'money', 'attorney', 'trade', 'crimin', 'secur']
['tax', 'return', 'prepar', 'fals', 'ir', 'client', 'incom', 'file', 'busi', 'indict']
['antitrust', 'price', 'competit', 'bid', 'industri', 'fine', 'charg', 'conspiraci', 'depart', 'division']
['medicar', 'health', 'care', 'fraud', 'medic', 'patient', 'hhsoig', 'claim', 'bill', 'servic']
['drug', 'traffick', 'mexico', 'state', 'organ', 'enforc', 'unit', 'attorney', 'distribut', 'law']
['bank', 'loan', 'mortgag', 'financi', 'fraud', 'inspector', 'scheme', 'million', 'us', 'attorney']
['inmat', 'right', 'assault', 'correct', 'civil', 'offic', 'attorney', 'violat', 'former', 'fbi']
['tax', 'prepar', 'injunct', 'return', 'complaint', 'perman', 'custom', 'incom', 'credit', 'busi']
['terrorist', 'attack', 'york', 'support', 'new', 'materi', 'us', 'fbi', 'nation', 'attempt']
['child', 'exploit', 'sexual', 'project', 'safe', 'children', 'us', 'attorney', 'abus', 'district']
['indict', 'charg', 'alleg', 'count', 'defend', 'right', 'attorney', 'us', 'fbi', 'investig']
['us', 'attorney', 'state', 'unit', 'district', 'illeg', 'assist', 'law', 'investig', 'protect']
```

Figure 3E: Mean Shift Results

```
meanShiftCentroids = getMeanShift(kMeansData,wordList,.99)
```

```
['tax', 'attorney', 'us', 'offic', 'district', 'assist', 'charg', 'investig', 'depart', 'crimin']
['we', 'threat', 'meet', 'home', 'work', 'agre', 'nation', 'terrorist', 'attack', 'intern']
['implement', 'manag', 'system', 'financi', 'depart', 'design', 'account', 'firearm', 'central', 'improv']
['2011', 'washington', 'natur', 'jan', 'must', 'hold', 'resourc', 'report', 'assist', 'gener']
['holder', 'commiss', 'jan', 'eric', 'must', 'jr', 'human', '2012', 'right', 'interest']
['call', 'women', 'violenc', 'jan', '2012', 'director', 'report', 'hous', 'justic', 'inform']
['holder', 'everi', 'eric', 'crime', 'releas', 'work', 'violent', 'prosecutor', 'reduc', 'throughout']
['firearm', 'safe', 'public', 'award', 'prevent', 'violenc', 'owner', 'crime', 'respons', 'reduc']
['competit', 'manufactur', 'materi', 'billion', 'develop', '2014', 'base', 'would', 'revenu', 'approxim']
['secur', '2015', 'presid', 'american', 'nation', 'threat', 'reduc', 'holder', 'purpos', 'eric']
['organ', 'must', 'feder', 'financi', 'public', 'day', 'base', 'provid', 'may', 'grant']
['consum', 'legal', 'north', 'competit', 'benefit', 'websit', 'bill', 'allow', 'gener', 'would']
['inc', 'product', 'manag', 'price', 'base', 'provid', 'would', 'san', 'depart', 'central']
['hous', 'foreign', 'american', 'terrorist', 'continu', 'section', 'safe', 'potenti', 'like', 'thank']
['presid', 'travel', 'foreign', 'order', 'american', 'protect', 'today', 'review', 'threat', 'safeti']
['market', 'power', 'share', 'antitrust', 'agenc', 'price', 'sale', 'address', 'justice', 'evid']
['contract', 'potenti', 'california', 'competit', 'could', 'report', 'antitrust', 'consum', 'effect', 'servic']
['facil', 'report', 'sexual', 'correct', 'specif', 'address', 'bureau', 'hold', 'mani', 'bring']
['follow', 'review', 'legal', 'natur', 'process', 'determin', 'seek', 'issu', 'statement', 'penalti']
['make', 'competit', 'reduc', 'commiss', 'effect', 'without', 'peopl', 'court', 'statement', 'american']
['ident', 'holder', 'request', 'eric', 'avail', 'gener', 'attorney', 'inform', 'today', 'document']
['direct', 'check', 'report', 'step', 'issu', 'fbi', 'system', 'take', 'the', 'nation']
['implement', 'execut', 'grant', 'order', 'safeti', 'issu', 'make', 'public', 'unit', 'depart']
['sheriff', 'counti', 'deputi', 'public', 'two', 'famili', 'live', 'everi', 'throughout', 'mark']
['famili', 'terrorist', 'safe', 'materi', 'support', 'depart', 'could', 'group', 'citizen', 'prosecut']
```

Figure 3F: Gaussian Mixture Model Results

GMM

```
['tax', 'attorney', 'us', 'offic', 'depart', 'district', 'justic', 'court', 'injunct', 'assist']
['attorney', 'us', 'tax', 'offic', 'district', 'depart', 'assist', 'state', 'right', 'charg']
['tax', 'attorney', 'us', 'indict', 'district', 'fals', 'crimin', 'investig', 'offic', 'assist']
['tax', 'attorney', 'us', 'offic', 'investig', 'district', 'charg', 'depart', 'justic', 'crimin']
['tax', 'prepar', 'fraud', 'medicar', 'attorney', 'return', 'us', 'health', 'care', 'district']
['attorney', 'us', 'charg', 'offic', 'state', 'district', 'depart', 'assist', 'indict', 'crimin']
['attorney', 'us', 'offic', 'right', 'assist', 'district', 'justic', 'investig', 'crimin', 'tax']
['us', 'attorney', 'tax', 'charg', 'assist', 'district', 'investig', 'fraud', 'sentenc', 'offic']
['attorney', 'us', 'tax', 'investig', 'offic', 'depart', 'district', 'justic', 'assist', 'law']
['tax', 'us', 'attorney', 'prepar', 'offic', 'law', 'return', 'depart', 'state', 'district']
['tax', 'attorney', 'us', 'district', 'charg', 'indict', 'offic', 'investig', 'fraud', 'depart']
['tax', 'attorney', 'us', 'fraud', 'offic', 'crimin', 'district', 'charg', 'medicar', 'assist']
['tax', 'attorney', 'right', 'district', 'us', 'depart', 'offic', 'prepar', 'civil', 'return']
['tax', 'attorney', 'us', 'indict', 'district', 'offic', 'fraud', 'investig', 'charg', 'depart']
['attorney', 'us', 'texa', 'offic', 'district', 'charg', 'depart', 'tax', 'fraud', 'assist']
['tax', 'attorney', 'us', 'offic', 'district', 'depart', 'justic', 'charg', 'return', 'state']
['us', 'attorney', 'offic', 'district', 'sentenc', 'crimin', 'depart', 'charg', 'assist', 'investig']
['attorney', 'us', 'tax', 'law', 'offic', 'assist', 'depart', 'fraud', 'right', 'justic']
['us', 'tax', 'attorney', 'district', 'charg', 'state', 'assist', 'offic', 'crimin', 'indict']
['attorney', 'tax', 'us', 'child', 'district', 'sexual', 'offic', 'sentenc', 'assist', 'crimin']
['tax', 'us', 'attorney', 'fraud', 'sentenc', 'offic', 'servic', 'investig', 'district', 'medicar']
['tax', 'attorney', 'us', 'depart', 'fraud', 'state', 'investig', 'assist', 'district', 'financi']
['attorney', 'us', 'offic', 'district', 'right', 'crimin', 'state', 'justic', 'assist', 'depart']
['attorney', 'us', 'offic', 'tax', 'crimin', 'district', 'investig', 'depart', 'state', 'charg']
['tax', 'attorney', 'us', 'district', 'assist', 'justic', 'investig', 'gener', 'crimin', 'offic']
```

Table 3G: Actual Crime Statistics Table, 2009 - 2016

Offense	2009	2010	2011	2012	2013	2014	2015	2016
Murder	923	958	979	1,044	400	370	385	381
Negligent manslaughter	0	0	0	0	2	0	1	2
Assault	1,065	1,131	1,110	1,131	1,791	1,678	1,788	1,849
Robbery	1,888	1,725	1,561	1,409	1,318	1,305	1,171	1,112
Sexual abuse	869	985	1,225	1,189	1,257	1,009	1,013	999
Kidnapping	250	242	203	194	218	223	176	158
Threats against the President	275	382	442	430	358	159	148	74
Fraud	19,520	19,675	19,416	16,913	17,086	14,012	14,066	13,071
Forgery	973	1,011	877	749	778	617	489	352
Counterfeiting	379	400	341	324	296	182	123	148
Embezzlement	3,601	3,782	3,907	3,515	3,870	3,387	3,271	2,649
Burglary	38	28	31	23	203	208	219	283
Larceny-felony	815	746	779	651	781	686	673	773
Motor vehicle theft	393	481	438	558	489	356	404	422
Arson and explosives	531	513	429	468	418	575	550	461
Transportation of stolen property	53	58	44	45	55	39	47	70
Other property offenses	398	409	413	457	1,020	668	831	723
Drug possession	880	853	912	843	721	638	777	802
Drug trafficking	34,479	34,659	36,333	34,580	34,134	27,025	29,254	28,167
Other drug offenses	106	109	101	165	132	112	116	117
Agriculture	58	12	12	12	38	24	32	47
Antitrust	33	31	28	36	23	14	37	42
Food and drug	137	172	144	261	136	185	226	185
Transportation	221	229	202	164	208	225	229	177
Civil rights	763	770	759	616	617	570	501	595
Communications	22	30	26	30	20	11	15	18
Customs laws	243	288	385	417	265	227	223	284
Postal laws	127	142	117	76	84	88	151	122
Other regulatory offenses	4,014	4,275	4,244	4,846	3,995	3,269	3,233	3,291
Weapon offenses	11,096	10,616	10,777	10,307	10,430	8,901	9,572	10,544
Immigration felonies	85,950	83,690	81,486	90,258	92,189	79,340	71,093	68,640
Tax law violations	1,072	1,085	1,127	1,061	894	788	800	684
Bribery	258	330	326	327	342	264	244	223
Perjury, contempt and intimidation	382	415	369	351	379	316	317	294
National defense	799	1,164	1,150	1,062	981	1,048	1,375	1,377
Escape	1,724	1,412	1,363	1,219	1,324	1,157	1,261	1,217
Racketeering and extortion	3,509	3,863	4,123	3,665	3,881	3,495	3,972	4,079
Gambling	89	157	129	60	198	89	68	76
Liquor offenses	11	11	33	42	115	119	113	150
Other sex offenses	3,889	3,703	3,840	3,622	4,027	3,618	4,244	3,701
Traffic offense	142	143	99	541	1,196	971	1,065	848
Wildlife offense	446	508	463	523	507	305	301	305
Environmental offenses	7	19	15	9	5	7	6	11
Conspiracy/aiding and abetting	3,819	4,055	2,568	1,565	1,214	948	1,171	1,033
All other offenses	860	827	947	1,016	1,290	994	1,275	1,007
Missing/Unknown	1,234	1,822	3,462	3,822	582	283	287	431
Total	188,341	187,916	187,735	190,596	190,267	160,505	157,313	151,994

From Bureau of Justice Statistics(bjs.gov)

Image Dataset Results:

Figure 3G: Confusion Matrix Using LDA

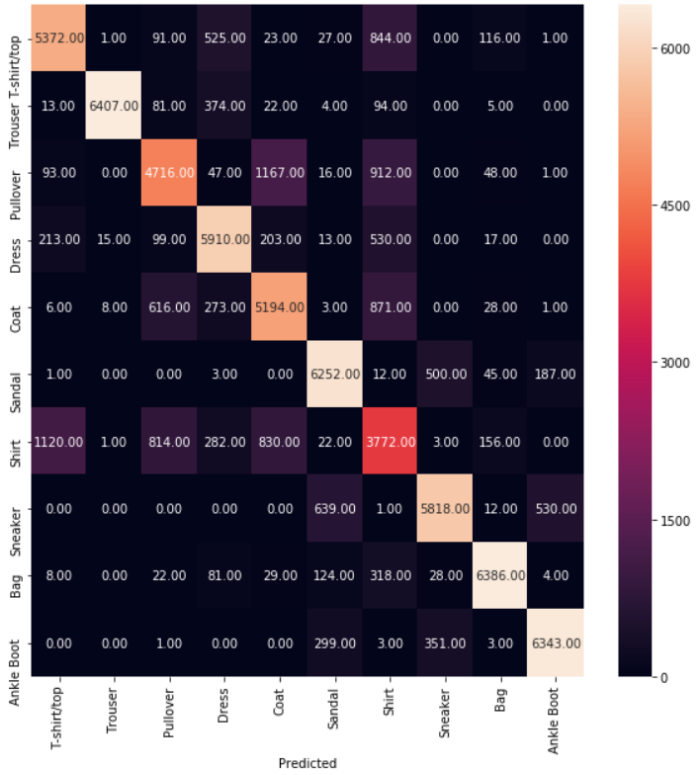


Figure 3H: Confusion Matrix Using QDA

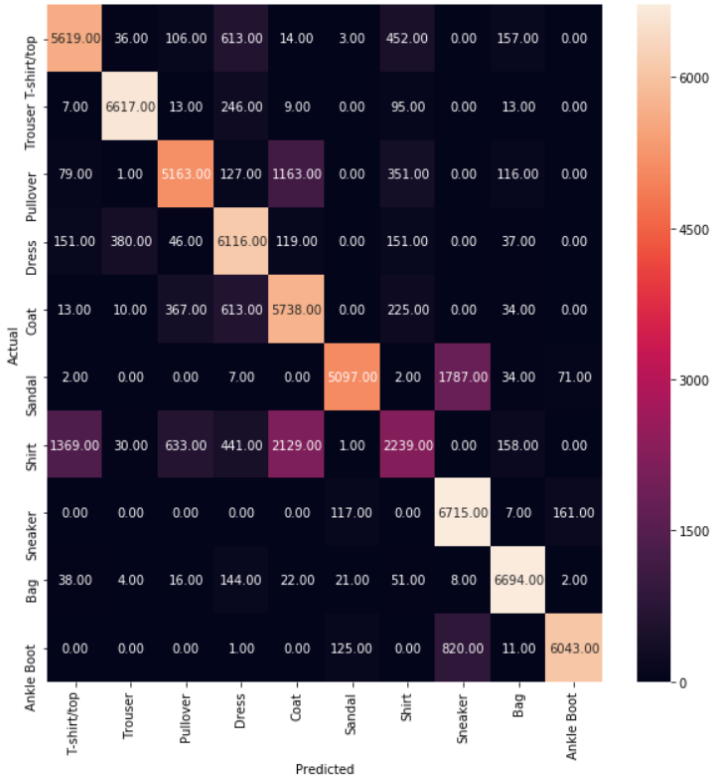


Figure 3I: Confusion Matrix Using K Nearest Neighbor, K = 5

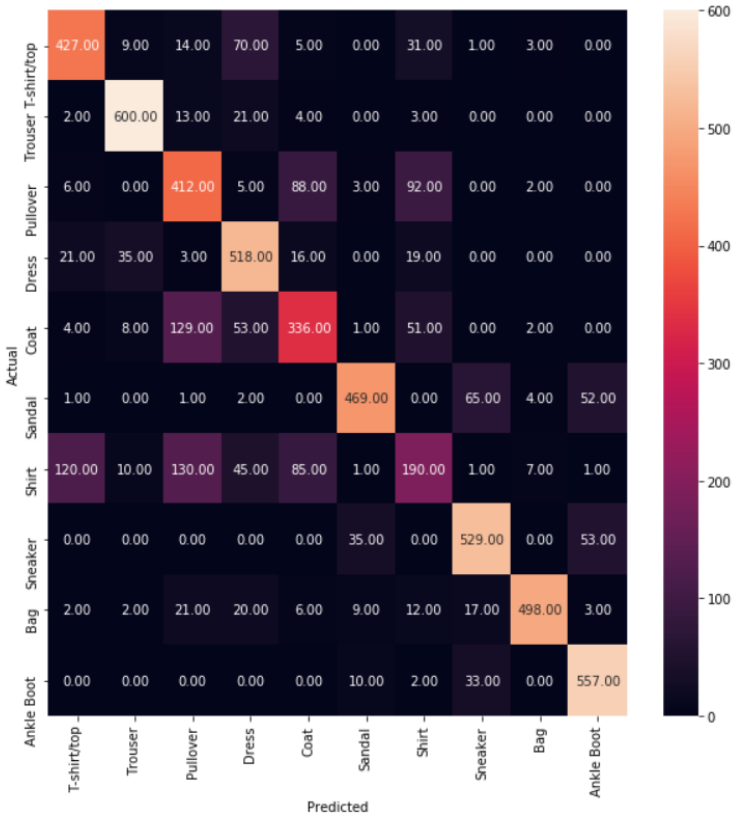


Table 3H: Overall Mean Accuracy Of All Methods

Method	Mean Accuracy
QDA	80.2%
LDA	81.9%
K Nearest Neighbor	76.0%

DISCUSSION

Time Series Dataset Results Discussion

For each algorithm, 10 – fold k fold cross validation was used, for five runs each, and the results were averaged. For each run, a different seed for Python's random number generator was used, and the data was scrambled randomly. The mean accuracy of prediction across all folds for all 5 runs for the Fortune 100 companies is shown above in Figures 3A, 3B, and 3C, and Tables 3A, 3C, and 3E. The Confusion matrices of prediction are shown in Tables 3B, 3D, and 3F.

We can see that LDA and Logistic Regression have approximately the same performance, both in terms of accuracy and which companies have the most accurate predications. QDA does not perform very well in any case.

Logistic Regression has a very slight performance edge over LDA. However, Logistic Regression is significantly more computationally expensive to calculate, since it involves using iterative numerical methods. Since the goal of the user of such a model would be to make money, it is recommended to use Logistic Regression.

These results suggest that the stock market is autocorellated and cross – correlated to some small degree, and that the underlying forces governing the market are correlated in some way. Since PCA revealed that relatively few principal components were needed to track the market, this would suggest that the markets as a whole are primarily governed by the actions of the "big fishes in the pond".

Text Dataset Results Discussion

Clustering Analysis is being used toward the objective of analyzing the types of crimes that took place from the press releases in order to obtain an insight into the affairs of the Justice Department and America. The clustering methods provide a view of the topics and ideas of the text dataset. The list of topics words per cluster , as shown in Figure 3D, Figure 3E, and Figure 3F, are the results of the three methods for the "Department of Justice 2009-2018 Press Releases".

K-Means and Mean-Shift provide various distinct clusters with lots of information while GMM provides clusters that seem to have almost the same topic. The topics/terms that seem prevalent across all three methods are "tax", "attorney", "fraud", "chrg", "investig". Other similarities that exist are along the themes around the term "anti-trust", sexual assault, civil violations, terrorism, and gang violence. The most common terms seem to point to tax fraud or financial/tax crimes and crimes related to businesses, according to these documents it would appear as through these are the prevailing types of crimes. However, that is not the case according to the statistics obtained from the Bureau of Justice Statistics, as shown in Table 3G. From the table its clear that financial, taxes, and business-related crimes exist but are not as prevalent as others. It is also clear that other themes that are not as prevalent in the analysis are more prominent in the statistics. This leads to the conclusion that the cluster analysis provides the prominent themes in the dataset, which itself reflects the press releases of the Justice Department and not the actual crime statistics. The objective was somewhat achieved but fell short of providing better insight into the crimes and affairs that take place in America due to the data rather than the methods.

As per the methods themselves, K-mean performs the best and would be the recommendation for analysis of this sort. K-means and Mean-Shift provided the most distinct and various results. GMM's results were not as distinct as the other two. Furthermore, Mean-Shift has O(n^2) complexity which causes it to be too slow

for large datasets. K-Means on the other hand, has linear time complexity making it faster Due to its nice results and speed K-Means is the recommendation for clustering.

Image Dataset Results Discussion:

The confusion matrices (Figures 3G, 3H, and 3I) show that shirts were the most misclassified across the board. This could be due to their shape being very similar to coats and shirts. Coats were misclassified to boots or pullovers, due to their similar shapes. Also sneakers and sandals (types of footwear) were also misclassified for each other. More unique clothing such as ankle boots, bags and trousers were almost never misclassified.

K-fold cross-validation was tested with different values of k: 5, 6, 10, 50, 100. Each of these numbers were chosen because they are sufficiently small enough to not increase the computational complexity from many trials while also ensuring that the data was evenly split. 6 folds were chosen as it maximized correct results with minimal computation time.

Amongst the three algorithms used, LDA and QDA are very similar algorithms, however QDA requires different covariance matrices for each class while LDA uses only one covariance matrix calculated from the training data. Due to this, LDA is less computationally complex than QDA. Also, when comparing each test sample, we perform only 10 comparisons (one for each class) and then select the best one to classify our sample. Both were all able to predict the correct class approx. ~80% of the time, with LDA being slightly better in predicting the correct class, especially when it came to shirts which was the most incorrectly classified article of clothing.

KNN was the most computationally intensive as for each test sample, there was 6000 (#training samples) computations and then the most frequent class was found. Just due to the raw number of operations, this was more inefficient than LDA and QDA. KNN was tested with K values 5, 6, 10, 20, 50, 100 and there was no gain in correct classifications found in increasing K so K = 5 was chosen. Also, this method was the least accurate of the three above (as seen in the confusion matrices), correctly classifying 76% of the time.

It is recommended to use LDA, as it was the least computationally complex of the three chosen algorithms and was also the best at correctly classifying articles of clothing. Doing much better than both QDA and KNN with shirts.

CONCLUSION

In this project, several important things were learned regarding machine learning. For all of the datasets various techniques were explored for pre-processing and feature extraction. Also different machine learning algorithms were researched and used for the different datasets. All of the machine learning techniques provided the tools to reach the objectives defined before. Linear Discriminant Analysis and Logistic Regression were able to predict stock trends more often than not, although not by a very large margin. This would suggest that the stock market is autocorellated and cross – correlated to some small degree, and that the underlying forces governing the market are correlated in some way. Since PCA revealed that relatively few principal components were needed to track the market, this would suggest that the markets as a whole are primarily governed by the actions of the "big fishes in the pond". Further analysis, in the form of autoregression testing, as well as analysis with a larger portfolio of stock data, would be a good follow-up study.

Furthermore, cluster analysis techniques such as K-means provided key insight into the Justice Department press release documents and the themes prevalent within them. Although the objective was not fully met, some interesting things were discovered. It would be beneficial to do further cluster analysis on actual crime records from the justice department for a better idea of crimes in America.

The image classification algorithms were all able to correctly identify articles of clothing by class. However, articles of clothing with similar shapes such as coats, shirts and pullovers were misclassified much more often than other classes which had more unique shapes such as ankle boots, bags and sneakers. This could have been due to the preprocessing removing many of the finer details in the middle of the clothing article (such as zippers, buttons or collars) that would differentiate some of these features, compared to just their shape. In the end, LDA is the best method among the three as it correctly classified objects the most while also being

the least computationally intensive. Ultimately the project provided an enlightening experience in terms of learning about machine learning and the different datasets.

BIBLIOGRAPHY

Time Series Datasets:

1. Nugent, Cam. "S&P500 Stock Data." *S&P500 Stock Data*, Kaggle, 2016, www.kaggle.com/camnugent/sandp500.

2. St. Louis Federal Reserve. "Whiting Oil Price." *DCOILWTICO*, St. Louis Federal Reserve, 2019, https://fred.stlouisfed.org/series/DCOILWTICO.

3. St. Louis Federal Reserve. "Federal Interest Rates." *INTDSRUSM193N*, St.Louis Federal Reserve, 2019, https://fred.stlouisfed.org/series/INTDSRUSM193N.

4. Quandl. "USD Daily Gold Prices." *USD Daily Gold Prices*, Quandl, 2019, https://www.quandl.com/data/WGC/GOLD_DAILY_USD-Gold-PricesDaily-Currency-USD.

5. Wong, Nick. "Daily Wheat Price" *Daily Wheat Price*, Kaggle, 2019, https://www.kaggle.com/nickwong64/daily-wheat-price.

Clustering Dataset

6. Wikipedia contributors. "Bag-of-words model." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 18 Dec. 2019. Web. 23 Dec. 2019. https://en.wikipedia.org/wiki/Bag-of-words_model

7. GeekforGeeks contributors. "". *geeksforgeeks.com*. GeekforGeeks. https://www.geeksforgeeks.org/tf-idf-model-for-page-ranking/

8. Seif, George. "The 5 Clustering Algorithms Data Scientists Need to Know". *kdnuggets.com*. KDnuggets, Jun. 2018. https://www.kdnuggets.com/2018/06/5-clustering-algorithms-data-scientists-need-know.html

9. Damien, RJ. "The mean shift clustering algorithm". *efavdb.com*. EFAV, 21. Apr. 2015. http://efavdb.com/mean-shift/#Tech

10. McGonagle, John. "Gaussian Mixture Model". *brilliant.com*. Brilliant. https://brilliant.org/wiki/gaussian-mixture-model/

11. B., John. “Department of Justice 2009-2018 Press Releases”. United States Department of Justice, Kaggle, 2019.
https://www.kaggle.com/jbencina/department-of-justice-20092018-press-releases

Imaging Datasets:

12. “Geometric Image Transformations.” OpenCV, docs.opencv.org/4.2.0/da/d54/group__imgproc__transform.html#ga5bb5a1fea74ea38e1a5445ca803ff121.

13. “THE MNIST DATABASE.” MNIST Handwritten Digit Database, Yann LeCun, Corinna Cortes and Chris Burges, yann.lecun.com/exdb/mnist/.

14. Wikipedia, Contributors. “Bicubic Interpolation.” Wikipedia, 17 July 2019, en.wikipedia.org/wiki/Bicubic_interpolation.

15. Wikipedia, Contributors. “Lanczos Resampling.” Wikipedia, 12 Mar. 2019, en.wikipedia.org/wiki/Lanczos_resampling.

16. “Zalandoresearch/Fashion-Mnist.” GitHub, 10 Aug. 2019, github.com/zalandoresearch/fashion-mnist.