Performance of SPX By VIX Index Regimes

STA2536H: Data Science for Risk Modelling

Authors:

Boris Migan 1009644725

John Ndolo 1009482560

Zhi Ye Luan 1004282867

Master of Financial Insurance

Department of Statistical Sciences,
FACULTY OF ARTS & SCIENCE

January 16, 2023

Abstract

VIX is the ticker symbol and the popular name for the Chicago Board Options Exchange's Volatility Index. It is the preferred measure of the stock market's expectation of volatility based on S&P 500 index options and is calculated on a real time basis according to the S&P 500. The VIX is commonly referred to as a "fear index" since its values are based solely on the performance of S&P 500 options which reflect major financial powerhouses of the U.S. economy . In that regard, this report will explore if the moniker is merited through two classification methods: Gaussian Naive Bayes and multi-class logistic regression in an attempt to predict future VIX levels. It was found the Gaussian model produced more accurate results compared to multi-class logistic regression.

Contents

1 Introduction **1**

2 Data Exploration **1**

2.1 Data Pre-processing and visualization 1

2.1.1 Returns 2

2.1.2 Histograms and heatmap of the data sets 3

3 Methodology and Results **5**

3.1 Data labeling 5

3.1.1 The Quantile method 6

3.1.2 Latent mixture models 6

3.2 Multi-class logistic regression 11

3.3 Gaussian Naive Bayes 12

4 Conclusions and Recommendations **16**

1 Introduction

This report is a hands-on experience in applying classification algorithms in real life problems. It entails the classification and predication of the VIX Volatility index using the recent past returns of the S&P 500 index. Our main objective is to explore how the SPX prices compare with different "levels" of the VIX index. To achieve, this we explored the suitable parametric models in observing the interdependence of the two data sets.

2 Data Exploration

The SPX(S&P) index and VIX volatility index over a period of ten years are the data sets used in this project. The data sets give information on the Open, High, Low, and Close prices for the S&P index and the Volatility Index(VIX).The VIX index is a measure of volatility in the markets and is derived from the Calls and Puts on the S&P 500 Options while the SPX index is a measure of daily S&P stock market prices.

2.1 Data Pre-processing and visualization

The data used was of over 8000 records. We first removed all null values in the data set then merged the two data sets (SPX & VIX) to match the day and to remove weekend VIX data - shrinking the entire data set to 2515 records. The figures below shows a visual summary of our data sets.

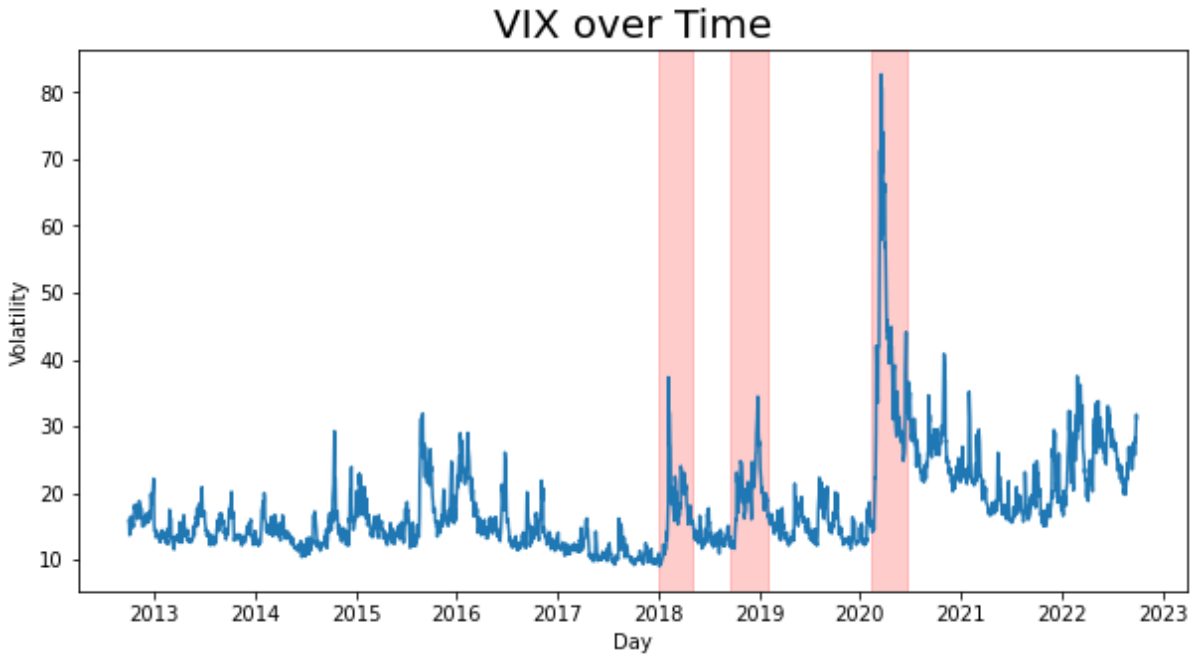

Figure 1: VIX Index Data

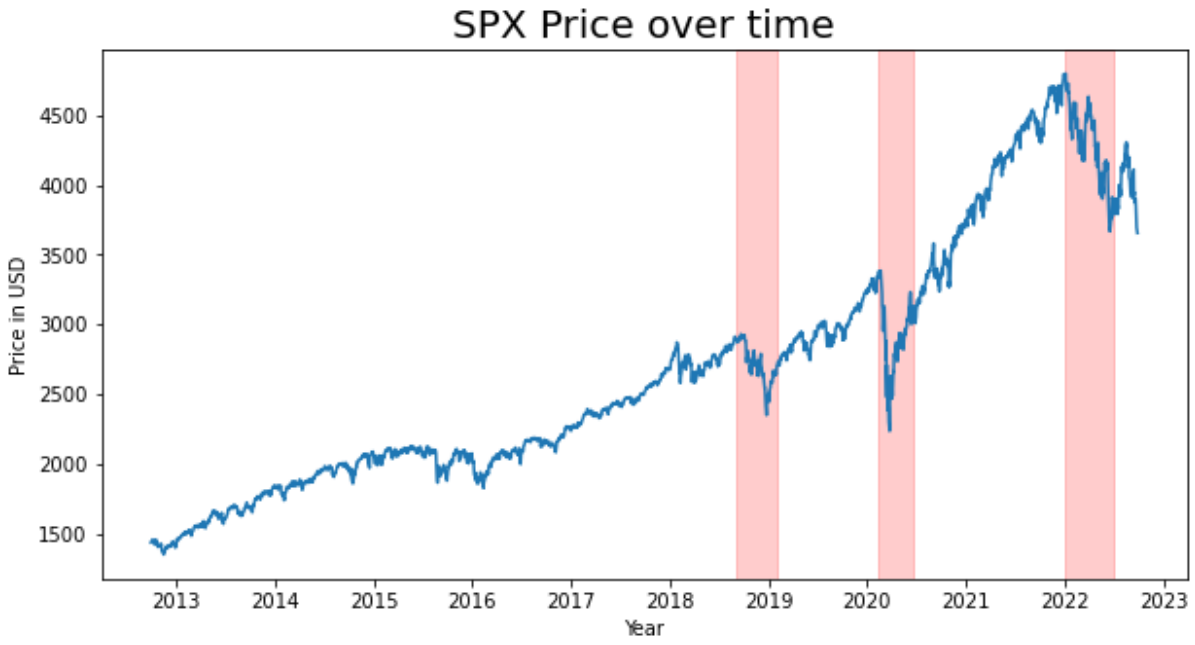

Figure 2: SPX Index Data

It can be observed from the above figures that there exists an upward trend in both SPX and VIX data. The trend in VIX data seems to increase steadily and less drastically while the trend in SPX is gradual and increases with time. However,we observed some stress areas that we highlighted in red bands on the plots. In the SPX data, we observe notable downward spikes in the end of the year 2018, early 2020 and early 2022. On the other hand, there are upward spikes in the VIX volatility index in almost corresponding years with SPX. We can argue that the spikes are mainly due to global crisis such as to the COVID-19 pandemic in 2020,the Ukraine-Russia war at the start of 2022 and general inflation rate changes across the years.

2.1.1 Returns

Returns are measures of variations in stock prices from one period to another [1]. Let R_t denote the returns at time t, and P_t denote prices at time t, then the formula for simple returns is given as;

$$R_t = \frac{P_t}{P_{t-1}} - 1. \tag{1}$$

According to Brockwell and Davis in the year 2002, log-returns portray better properties such as stationarity and normality, which prove better for analysis in most instances [2]. The formula of log-returns is given as;

$$\log(1 + R_t) = \log\left(\frac{P_t}{P_{t-1}}\right) = P_t - P_{t-1}. \tag{2}$$

In this project, we only considered the simple returns of the SPX data sets and the graphical representation of the returns is as follows;

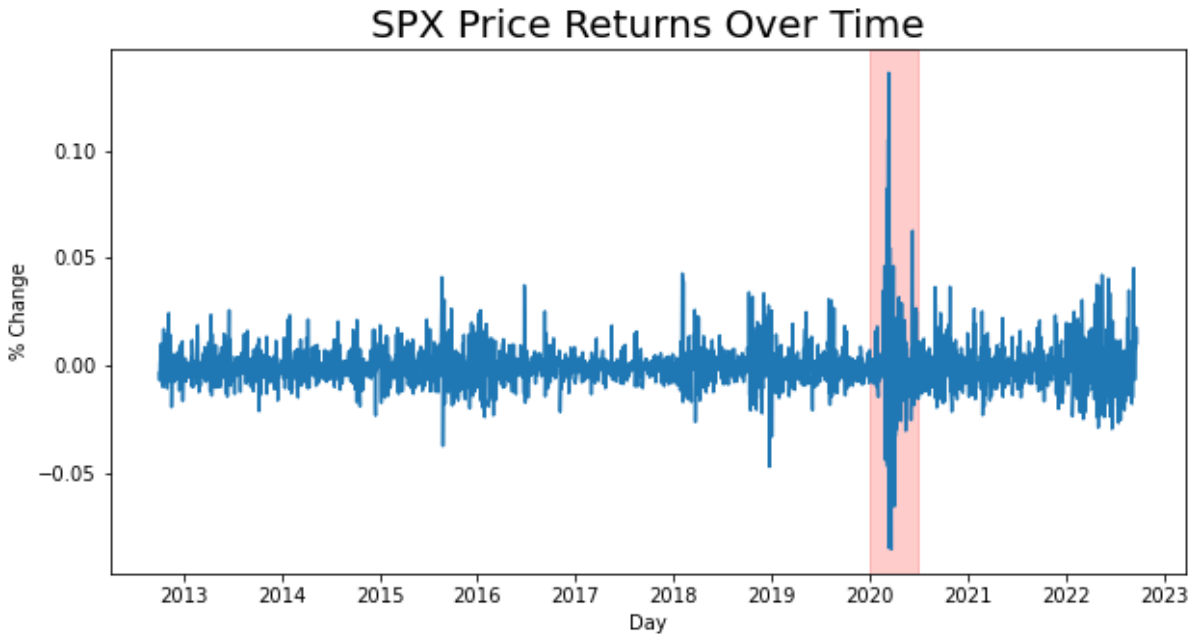

Figure 3: SPX Returns

It is evident from the plot, that the returns of the SPX were stationary for most parts of the 10 year period apart from the variation highlighted in red band between the year 2020 and 2021 which might be attributed to the rise of Covid 19 pandemic.

2.1.2 Histograms and heatmap of the data sets

To understand the distribution of our data, we generated unconditional histograms of the VIX (not returns, but levels) and the SPX returns. This is illustrated in the figures below.

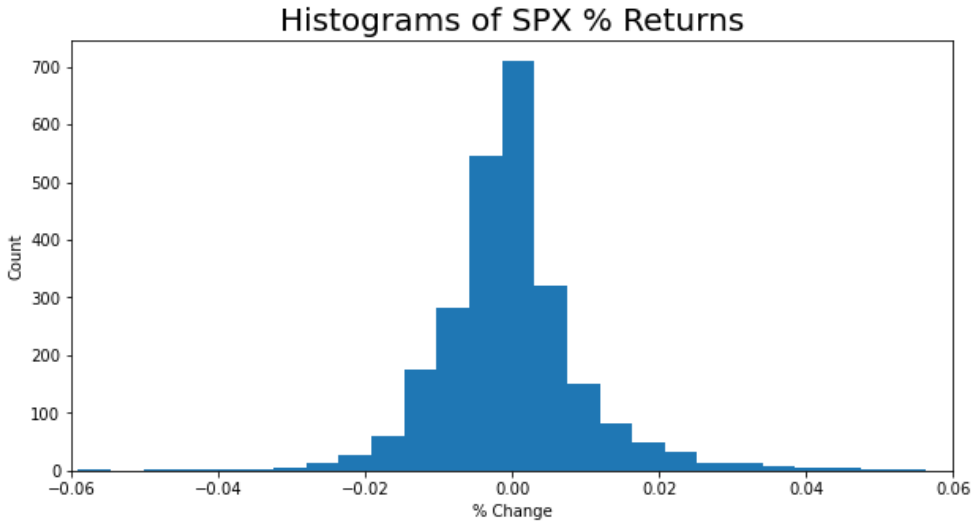

Figure 4: SPX % Returns

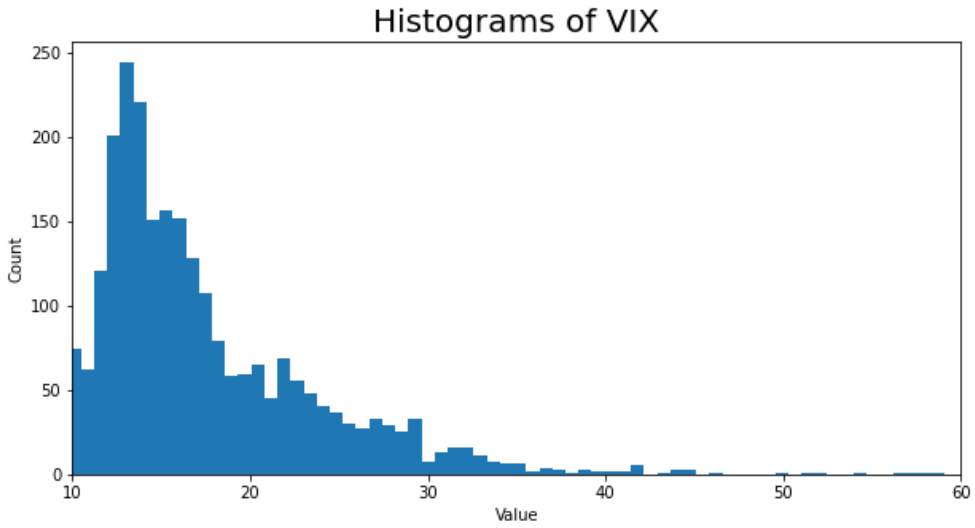

Figure 5: VIX Histogram

In figure (4), the distribution of the SPX data tends to assume the Gaussian-like distribution. On the other hand, figure (5) demonstrates the distribution of VIX volatility index which is positively skewed. Based on the analysis of the two histograms, we proposed some Parametric Models that would best fit our data. By definition, a parametric Model is a particular class of statistical models in which we find a family of probability distributions that has a finite number of parameters. In our case, the Gaussian distribution which is also known as normal distribution, would be most suited to fit the SPX data while a family of heavily tailed distributions such as exponential distributions would be viable to fit the VIX data.

To have a better visualization of the distribution of our data sets we generated a heat map of the joint Kernel density estimates (KDE) of the S&P returns and the VIX index as shown in figure (10) with the closer concentric rings to represent higher point clustering. To the eyes we could not easily tell the number of clusters distinguishable from the data. In addition, looking at the figure we notice that, as the VIX level increases the clusters seems to behave differently. Therefore, we opted to use a latent mixture model, whose main goal was to identify different clusters in the data set with some level of accuracy.

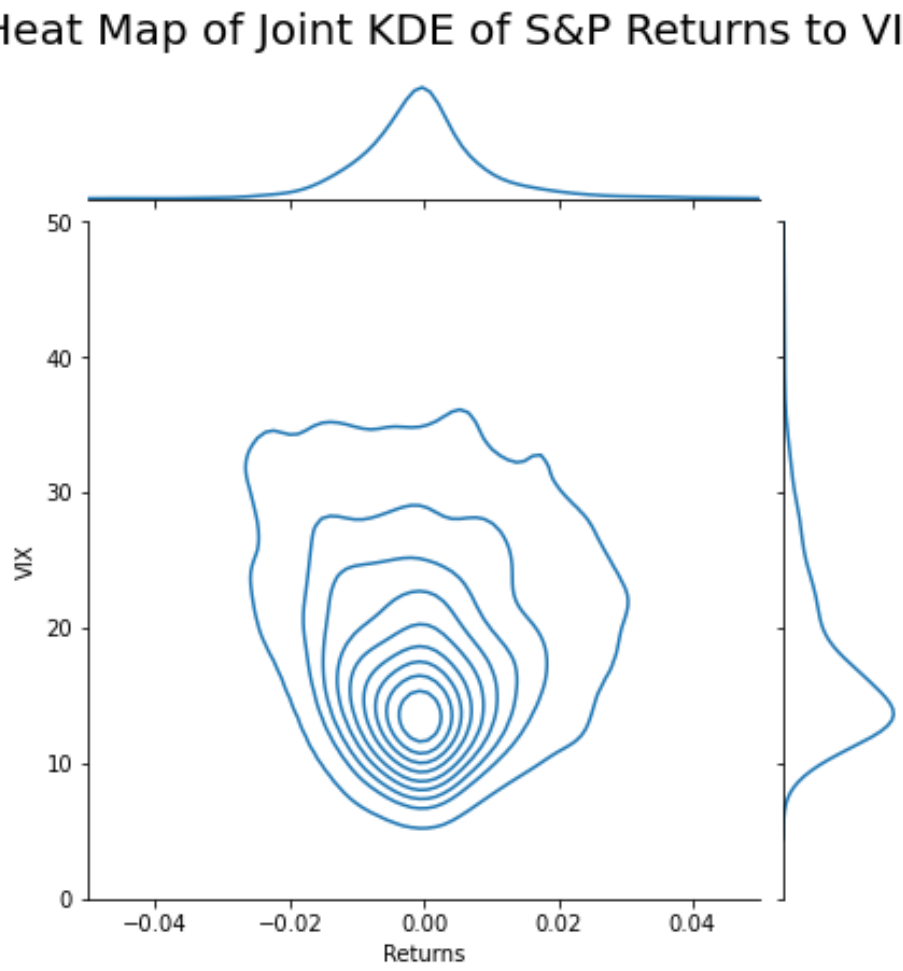

Figure 6: Joint KDE Heat Map of S&P Returns and VIX

3 Methodology and Results

3.1 Data labeling

There are different approaches of labeling data. In our case, the VIX data was unlabeled, hence we considered unsupervised methods of classifications to label the data into three categories. We considered VIX index data to be the features to classify. Let $\mathcal{X}$ denote the feature space, consisting of all the possible features we are considering(Volatility index values) and $\mathcal{Y}$ which takes value in the discrete set of class labels our features take. Throughout the report we used *Low*, *Medium* and *High* to denote the three possible class labels assigned to the data. We decided to adopt this classification method as it is the most optimal and basic method that fits the context of our data. To solve our classification problem we needed to specify a classification rule and by the rule we grouped the data into different classes. We have summarized the two methods we used to label our data below.

3.1.1 The Quantile method

For the quantile Method we divided the data into three almost equal parts using the quantile function in python. We obtained 13.79 VIX value being the threshold from *Low* and *Medium* and 18.08 VIX value being the threshold from *Medium* and *High*. We can see in figure (7) the plot of the VIX values along with the VIX regimes where the color bands represents the three different regimes. Our Labels consist of 840, 838 and 839 data points for the Low, Medium and High Levels respectively. we are fully aware that "eyeballing" and using quantiles is not going to produce the more accurate results as it might result on over-weighting of the outlier for each regimes thus we have decided to used the Gaussian Mixture Model (GMM).

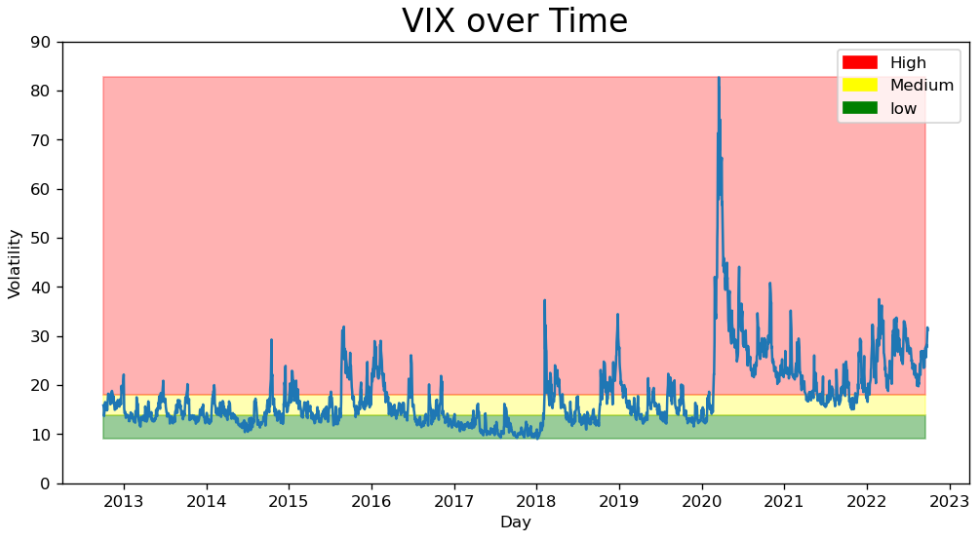

Figure 7: Plot of VIX with Labels over time

3.1.2 Latent mixture models

Unlike other classification methods, labels are not observed in mixture models and thus it is considered an unsupervised method of classification. In this method, the underlying assumption is that the data are generated by randomly selecting a (hidden) label and then, given that label, the features are randomly selected from a distribution that depends on that label. As a result, samples are generated from a weighted average of the conditional distributions, in others a mixture of the distributions hence the name mixture model [3].

More specifically, suppose our data is given by a feature space $\mathcal{X}$ and labels given as $\mathcal{Y}$ and we have N data points. Given mixture models are generative types of distributions, we model the conditional distribution of the feature conditional on the specific class (Latent class labels in this case). The prior distribution of the labels is given by;

$$\pi_y := P(Y = y), \quad y \in \mathcal{Y} \tag{3}$$

Once a label Y is drawn, we specify the distribution (e.g., Gaussian, exponential,Poisson, multinomial) of the points in feature space, and we write the generative model:

$$\mathbb{P}(X = x|Y = y) = g_y(x, \theta), \quad x \in \mathcal{X},\ y \in \mathcal{Y}, \tag{4}$$

where θ denotes the model parameters of the conditional distribution (e.g., mean and variance for conditional Gaussian).

In our case, based on the deductions from Figure(4), Figure(5) and Figure(10) our data sets are Gaussian-like hence we considered the conditional distribution to be Gaussian and as a result we employed the Gaussian mixture models (GMM).

3.1.2.1 Gaussian mixture models

Based on our deductions that our data is Gaussian-like, we settled for the Gaussian mixture models. Therefore we assumed that conditioned on label $c \in \mathcal{Y}$, the points in the VIX index are normally distributed. The Gaussian generative model we assumed is of the form;

$$g_c(x) := \frac{e^{-\frac{1}{2}(x-\mu_c)^T\Sigma_c^{-1}(x-\mu_c)}}{\sqrt{(2\pi)^d \det \Sigma_c}}, \qquad x \in \mathbb{R}^d, \quad c \in \mathcal{Y} \tag{5}$$

In our case, we implemented the algorithm using the python inbuilt SciKit Learn which simplified our work. However, the main notion behind the Gaussian mixture model is estimation of its parameters which are the mean (μ_c) and covariance matrix (Σ_c) for all the classes, where c denotes the number of classes of which we settled for 3 classes on our model. The key process of implementing the Gaussian model was;

(i) Estimation of the log-likelihood function which in general is given as;

$$\ell(\Theta) = \sum_{j\in N}\sum_{c\in\mathcal{Y}} \gamma_{jc}^{(k)} \log g_c(x_j \mid \Theta) \tag{6}$$

$$= -\frac{d}{2}\log(2\pi)\sum_{j\in N}\sum_{c\in\mathcal{Y}}\gamma_{jc}^{(k)} - \frac{1}{2}\sum_{j\in N}\sum_{c\in\mathcal{Y}}\gamma_{jc}^{(k)}\Big\{\log \det \Sigma_c + (x-\mu_c)^T\Sigma_c^{-1}(x-\mu_c)\Big\} \tag{7}$$

$$= -\frac{d}{2}\log(2\pi)N - \frac{1}{2}\sum_{j\in N}\sum_{c\in\mathcal{Y}}\gamma_{jc}^{(k)}\Big\{\log \det \sum_c + (x-\mu_c)^T\Sigma_c^{-1}(x-\mu_c)\Big\} \tag{8}$$

(ii) Maximization of the log-likelihood and estimation of the parameters of the mean and covariance which are give as;

$$\mu_c = \frac{\sum_{j\in N}\gamma_{jc}^{(k)}x_j}{\sum_{j\in N}\gamma_{jc}^{(k)}} \tag{9}$$

where this is the weighted sample means of the classes, and

$$\sum_{pq}^{c(k+1)} = \frac{\sum_{j\in N}\gamma_{jc}^{(k)}\big[x_j-\mu_c\big]_p\big[x_j-\mu_c\big]_q}{\sum_{j\in N}\gamma_{jc}^{(k)}} \tag{10}$$

is the weighted covariance matrix of dimension p and q, where the weights are equal to the responsibility of that data point being in class c.

3.1.2.2 Bayes Classification rule with mixture models

Mixture models are used to perform classification of data through the posterior estimates of the label identity [3]. Based on updates computed for model parameter estimates $\hat{\Theta}$, using EM-algorithm, we compute the posterior expectation of any given class as a function of location in feature space using Bayes rule as follows;

$$\mathbb{P}(Y = y \mid X = x, \hat{\Theta}) = \frac{\mathbb{P}(X = x \mid Y = y, \hat{\Theta})\mathbb{P}(Y = y \mid \hat{\Theta})}{\mathbb{P}(X = x \mid \hat{\Theta})} \tag{11}$$

$$= \frac{\mathbb{P}(X = x \mid Y = y, \hat{\Theta})\mathbb{P}(Y = y \mid \hat{\Theta})}{\sum_{c \in \mathcal{Y}} \mathbb{P}(X = x \mid Y = c, \hat{\Theta})\mathbb{P}(Y = c \mid \hat{\Theta})} \tag{12}$$

$$\tag{13}$$

The classification rule assigns the class label at a point in feature space $x \in \mathcal{X}$ as follows;

$$h(x) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\ \mathbb{P}(Y = y \mid X = x, \hat{\Theta}) \tag{14}$$

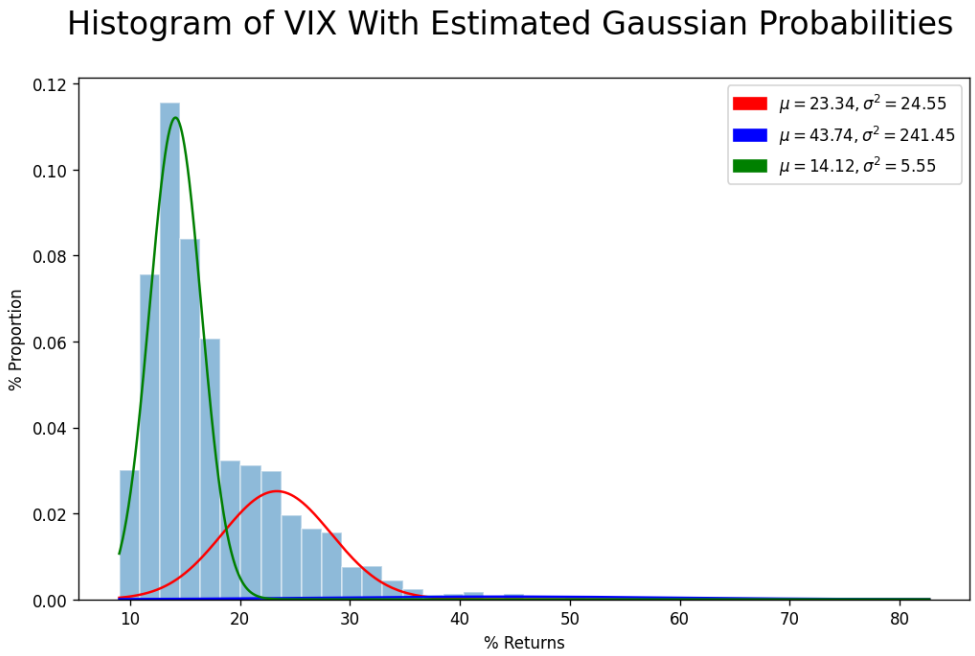

Figure 8: Class Results of Unsupervised Learning

Before we get into the implications of how our Gaussian mixture model on the returns on the SP 500, there is something to be said about the choice of using Gaussian Mixture models in the first place. In the figure above, we can see a histogram of VIX as well as the three classes of low, medium, and high level each with its own mean and variances. Despite the histogram not exhibiting the distinctive bell-pattern of the Gaussian distribution, we can see that the cumulative sum of the classes fit fairly well for the given data.

If we consider the semantics of having a "low", "medium", and "high" class for VIX in the first place, we would like for the model to be able to tell us the uncertainty of the market of the current day. This model fits that criterion quite well. We can see that most of the data falls under the likelihood of being in the green class with a lower mean and variance compared to the other classes. It is because of this reason, that the classes with lower mean will be designated as the "low" volatility class meaning most of the time, SP falls under expected circumstances. If we look at the tail end of the spectrum, we see that very few data points fall under the likelihood of belonging to the blue class. The benefit of using Gaussian classifiers can be alone explained by the existence of the blue class of which few data points belong. From a practical standpoint, it does not really make sense to have a heavily populated "high" volatility class as the existence of extraneous circumstances are only merited by their rarity.

One could make an argument *"why we are not using even more classes to classify VIX?"* To such questions, we make the counter argument of *"why do we need such classes in the first place?"* From a purely technical standpoint, there may be a higher class specification that produces better information criterion scores (e.g. AIC, BIC, etc.). However, our discussion is mainly focused on the necessity of classifying VIX into levels, we relegate the discussion of "optimal class numbers" and its interpretations for another time. When we look at the histogram of VIX, we only really see two possible peaks, one centered at 15, and the other centered at around 21. While it is nice to be sure of the current state of the world we live in, we risk pure statistical inference over practical implications.

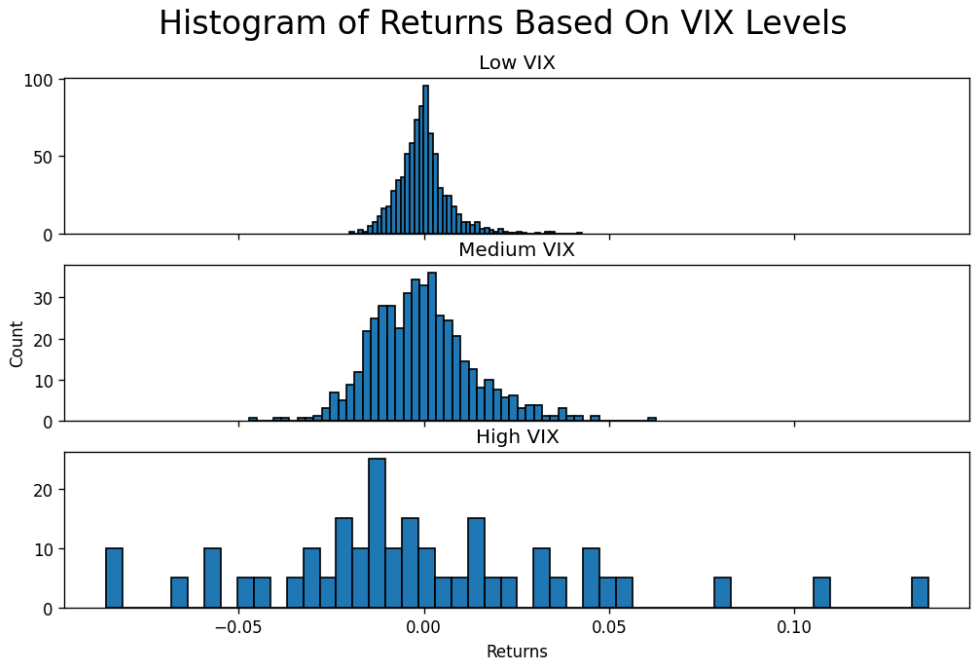

Figure 9: Conditional distribution of SPX given VIX

Above we can see the results of our unsupervised labeling. It can be observed that the returns conditional on the VIX classes slightly resemble the normal distribution. When

we performed a Shapiro-Wilkes test on all the three conditional samples of the volatility classes of low, medium, and high, we found that the only class where normality test failed was the high volatility class with a p-value of around 0.09. On the other hand, the low and medium classes had p-values significantly less than 0.01. This lead us to assume that the low and medium volatility classes have shifted beta distributions with a high beta parameter that leads to a higher Kurtosis. This further suggests that the high volatility class its true distribution is beta distributed but just so happens to have a milder Kurtosis. When observing the plotted distributions, there is evidence to suggest that the variance of the data is conditional on the volatility class with the lowest class having most of the data highly concentrated around zero. As we move up the volatility classes, we can see that the samples are taking on distributions with longer and longer tails with the high volatility class having the most dispersed tail and highly skewed. The figure below summarises our observations.

A natural extension of this idea of daily classification is the notion of forecasting. *"Can we apply this to future observations?"* comes to mind when exploring the volatility of the market on a given day. How much can we rely on the current status of the volatility index? Are there any correlations with the performance of S&P 500 today with its performance tomorrow? Let us observe the following plot to answer these questions.

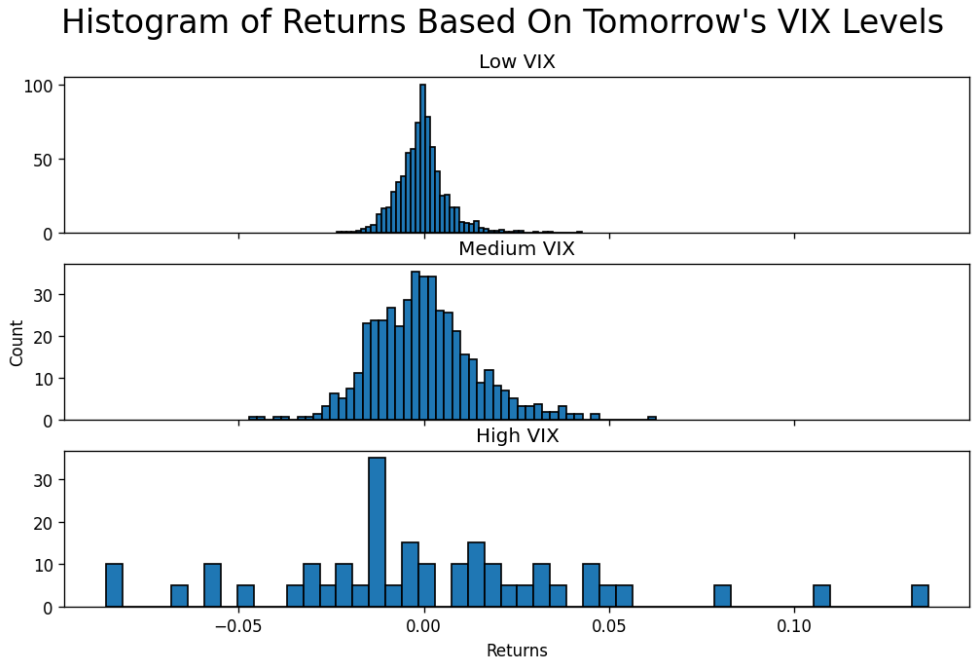

Figure 10: Conditional distribution of SPX given VIX

We can see from the figure of today's stock price based on tomorrow's VIX suggests that the return distributions do have some form of observable conditionality on the volatility classes. The condition seems to mirror the same day VIX conditional distribution quite closely with the low volatility class having the smallest deviation from the center mean with the deviation only increasing the volatility classes progresses to high. This result is quite exciting as it suggests that the VIX levels are robust enough to not be swayed by short interday differences.

In broader strokes, the results suggest there is merit to exploring deeper into the relationships of returns to VIX levels and returns. Is it possible to forecast future VIX levels based on today?

3.2 Multi-class logistic regression

In our work, the Multi-logistic regression was considered as a method of cl ossification with the results explained in section 4. We used the "SciKit Learn' library in python and adapted our data to it for ease of classification. However, the generalisation of the algorithm of multi-logistic adopted is as highlighted below. By definition, Multi-Class Logistic Regression also known as the Maximum entropy classifier is a generalisation of the Logistic regression and belongs to the family of discriminative models [3]. It simultaneously accounts for existence of multiple labels than classifying them one one by one [3]. As a generalization of logistic regression, it assumes that:

$$\mathbb{P}(Y = c \mid X = x) := \mu_c(x) = \frac{e^{w_c^T x}}{\sum_{y=1}^{C} e^{w_y^T x}} \tag{15}$$

with $w_1 = 0$ for identifiability. The critical part of the classifier was estimation of the model parameters with its likelihood function given as:

$$\begin{aligned} \mathcal{L}((w_c)_{c\in\mathcal{Y}}) &= \mathbb{P}((Y_i = y_i)_{i\in\mathcal{N}} \mid (w_c)_{c\in\mathcal{Y}}, (X_i = x_i)_{i\in\mathcal{N}}) && (16)\\ &= \prod_{i\in\mathcal{N}} \mathbb{P}(Y_i = y_i \mid (w_c)_{c\in\mathcal{Y}}, X_i = x_i) && (17)\\ &= \prod_{i\in\mathcal{N}} \frac{e^{y_i^T W x_i}}{\sum_{y=1}^{C} e^{w_y^T x}}, && (18) \end{aligned}$$

and the Log-likelihood function $\ell(W) = log(\mathcal{L}(W \mid X)$ is:

$$\ell(W) = \sum_{n=1}^{N} (y_n W x_n - log \sum_{c=1}^{C} e^{w_c^T x_n} \tag{19}$$

where $W = (w_1, w_2, \cdots, w_c)^T$, and $[y_n]_c = \mathbb{1}_{Y_n=c}$

The estimation of the model parameters is not tractable using analytical methods, thus we consider iterative procedure such as newton-Ralphson method and gradient descent. The gradient descent update of the parameters is given as;

$$\hat{W}^{k+1} = \hat{W}^k - \eta \nabla \ell(W) \tag{20}$$

The optimal goal is find the argument ţhat maximises the linear function $W_c^T X$ which changes for every single class which as leads to linear classifications.

3.3 Gaussian Naive Bayes

Gaussian Naive Bayes is a type of multinomial features classifiers. It is normally useful when the feature space assumed is continuous and take on values in reals, that is $[x_i]_q \in \mathbb{R}$. The classifier assumes that a given dimension in feature, conditional on the label $c \in \mathcal{Y}$, is normally distributed. The generative model associated with Gaussian naive Bayes is defined as,

$$g_q(x; y) = \frac{1}{\sqrt{2\pi(\sigma_q^y)^2}} e^{\frac{-1}{2}\left(\frac{x-\mu_q^y}{\sigma_q^y}\right)^2} \tag{21}$$

To implement the model, we used python libraries "SciKit Learn" with main steps of consideration been parameter estimations and the determination of the classification rule. The collection of model parameters to be estimated are the mean, covariance and class responsibility given as;

$$\Theta = \{(\mu_q^c, \sigma_q^c) q \in \mathcal{D}; \pi^c\}_{c \in \mathcal{Y}}, \tag{22}$$

where the class responsibility $\pi^c = \mathbb{P}(Y = c)$. The parameters may be estimated using the MLE method with the estimated parameters given as follows;

(i) Responsibility of every class given as

$$\hat{\pi}^c = \frac{n^c}{n}$$

where n^c is the number of times a class c is observed and n is the total number of classes.

(ii) Mean $\hat{\mu}_y^p = \frac{1}{n^y} \sum_{i \in \mathcal{N}} [x_i]_p \mathbb{1}_{y_i = c}$

(iii) The covariance estimated as; $(\hat{\sigma}_p^y)^2 = \frac{1}{n^y} \sum_{i \in \mathcal{N}} \left([x_i]_p - \hat{\mu}_p^y\right)^2 \mathbb{1}_{y_i = c}$, which is the variance of the $P - th$ component of the features with label y.

Upon implementation of the models we observed the results of the supervised learning from the two models: multi-class logistic regression and naive Gaussian Bayes classifiers. For all of the models, we split the 90% of the data as the training data and reserved the final 10% of the data for testing prediction accuracy. We had split the data in this manner where we specified that the testing data would come after the training data is so that future results do not influence future observations which would undermine the goal of predicting future data. Below are the python output of these models with respective interpretations.

```
Training Accuracy : 0.661
Accuracy Score : 0.151

Confusion Matrix of Past 3 Days as 3D Vector 90 - 10 Split Multiclass Logistic
-----------------------------------------------
[[ 38   0   0]
 [212   0   0]
 [  1   0   0]]
```

```
Intercepts:
[ 1.41100553  0.68136543 -2.09237096]
Class Coefficients:
[[ 0.33921484  0.27236369  0.12023547]
 [-0.3490579  -0.20924535 -0.05429672]
 [ 0.00984306 -0.06311833 -0.06593876]]
```

First, we observed the results of the multi-class logistic regression trained on the estimated class labels from the unsupervised GMM model earlier using the past 3 days of returns from the SPX as a $3D$ vector per day. The main metrics we paid attention to were the training accuracy, the accuracy score, and the confusion matrix. We saw that training accuracy for the multi-class logistic was very low at 66.5% with an even lower prediction accuracy score of 15.1%. In addition to having a a poor prediction accuracy, we can see that in the confusion matrix, the model predicted all of the prediction data points to be under the first class. This suggests that the 15% accuracy score was in fact inaccurate as the accuracy was only dependent on the proportion of the amount of data points belonging to class one to the total amount of training data. This was confirmed by re-training the model ten times which produced almost similar result.

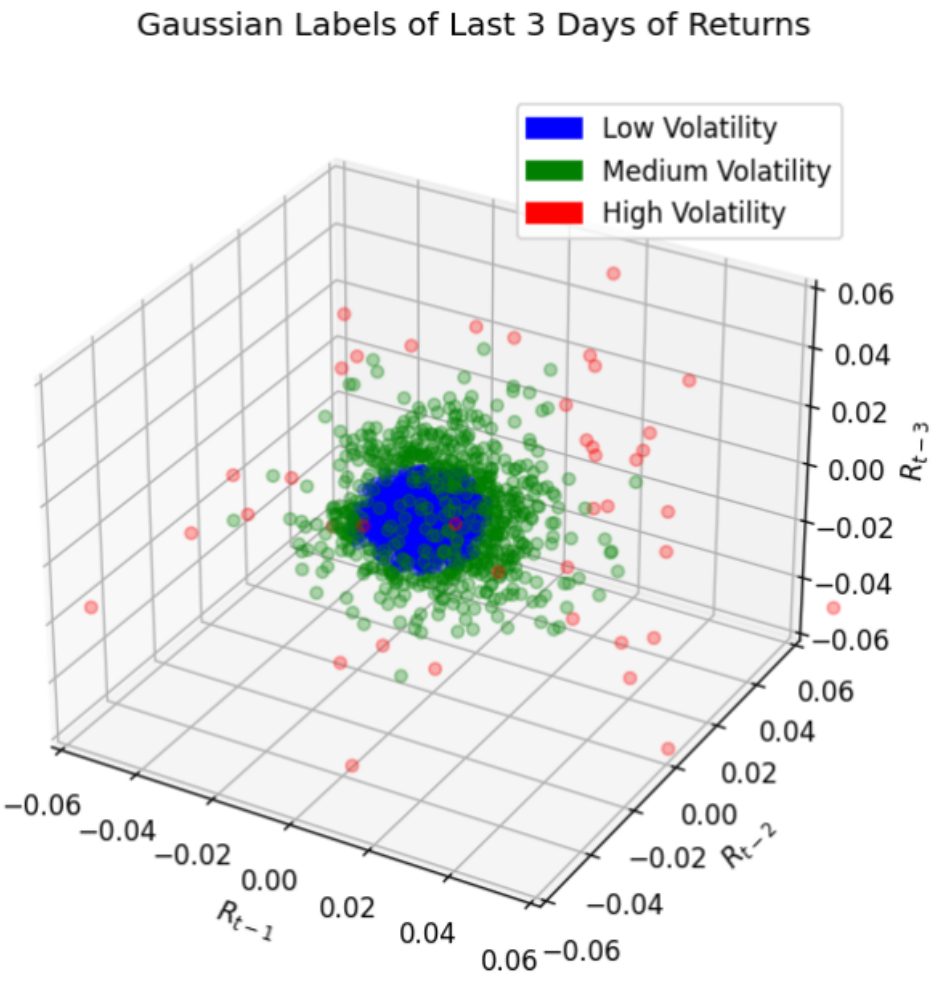

Figure 11: 3D representation of the labels using GMM

We observed that our data set was very unbalanced with most of the data located between Medium and low classes while very few data points were located in the high class. This is well illustrated in the figure(11) on top, which represents the results of data labeling using the Gaussian Mixture Model (GMM).

A solution to this issue and that might have improved the multi-class logistic regression fairly poor accuracy score of 15.1% may be attributed to under-sampling or Over-Sampling. Removing samples from the over-represented classes or adding more samples from the under-represented classes may have improved classifier performance. Another method another method we propose is to estimate the class weights and then provide bias toward the minority classes during training phase.

An observation that would have been good to consider a priori is that logistic regression works on linear boundaries. When observing the 3D plot of the returns, we can see that the classes in fact spread out radially from the center which suggests that ellipsoidal boundaries may have performed better.

add figure of 3D labelling here

```
Training Accuracy : 0.753
Accuracy Score : 0.618

Confusion Matrix of Absolute Sum of Past 3 Days 90 - 10 Split Multiclass Logistic
-------------------------------------------------
[[ 30   8   0]
 [ 84 125   3]
 [  0   1   0]]
Means of Each Feature Per Class:
[[-1.66113779e-05 -7.05024781e-05 -1.88073021e-04]
 [-8.02616366e-04 -5.91806189e-04 -3.60740244e-04]
 [-3.05498212e-05 -1.80846718e-03 -1.88216976e-03]]
Variances of Each Feature Per Class:
[[4.76414287e-05 5.09944486e-05 5.47162414e-05]
 [1.87422739e-04 1.83267902e-04 1.79596550e-04]
 [1.80057991e-03 1.75540347e-03 1.68936050e-03]]
```

‘

We now observe the result of the naive Gaussian Bayes classifier once more trained on the estimated class labels from the unsupervised GMM model using the past 3 days of returns from the SPX as a 3D vector per day. Here we notice that the training accuracy is very decent at around 75.3% with a lower but still decent prediction accuracy at 61.8%. These results can be further examined by looking at the confusion matrix and we can clearly see that this model in fact fails at accurately predicting the medium and high volatility classes and is also more wrong than right when it comes to the first class. On the other hand we can see below the result still using the Naive Gaussian Bayes still trained on the estimated class labels from the unsupervised GMM model. However, this

time using the sum absolute returns from the SPX form the past 3 days. Here we notice that the training accuracy is barely lower than the previous method but is still very decent at around 74.6% but the prediction accuracy is noticeably Lower at 56.6%. When analysing the confusion matrix for this model we observe the opposite compared to the first approach for this model in terms of class 2 and 3 as we have a better accuracy for class 1 than class 2 but still no accurate prediction for the high class.

```
Training Accuracy : 0.746
Accuracy Score : 0.566

Confusion Matrix of Absolute Sum of Past 3 Days 90 - 10 Split Multiclass Logistic
-------------------------------------------------
[[107 104   1]
 [  3  35   0]
 [  1   0   0]]
Means of Each Feature Per Class:
[[0.03037057]
 [0.01480421]
 [0.09184899]]
Variances of Each Feature Per Class:
[[0.00032042]
 [0.0001201 ]
 [0.00480762]]
```

4 Conclusions and Recommendations

After the exploration of the relationship between SPX and VIX, we came to the conclusion that there is indeed relationship between the volatility of the price of SP 500 with the level of the "fear index" VIX. It was found that while multi-class logistic regression models perform quite poorly on regimes classified by unsupervised Gaussian mixture models, the circumstances that caused the poor performance suggested that using logistic regression in the first place was not a prudent decision (data imbalance and linear decision boundaries on non-linear regimes). We observed that naive Gaussian Bayesian classifiers performed (comparatively) better with the model trained on vectors of last three days of returns performing slightly better than on the sum of absolute returns of the last three days. These classifiers were correct at predicting future volatility levels based on past return data more times than not.

A factor that we did not explore in this report was the usage of multi-class logistic regression. We did not take into account the conditions for logistic regression to perform well before we trained the model.

Another factor we did not explore was how the model performance may have changed given less training data. Would the Gaussian Bayes classifiers fared just as well with less training data and more testing data? If it did, not only would it suggest that past volatility stay consistent throughout a ten year period, but also that future volatility can be predicted with less information.

An addendum area we did not explore was the change in performance from labeling data into more regimes. We majorly focused on exploring on whether the relationship between the VIX and recent SPX was significant in the first place. In other words - was the "fear index" truly merited? Perhaps if we wanted more accurate prediction results, we might have observed more favourable information criteria during the unsupervised learning portion of this report. But perhaps this was a fruitless endeavor to begin with, after all the labels that we fitted out models to were themselves the product of unsupervised learning. Perhaps using Gaussian mixture models to classify labels had influenced the performance of Naive Gaussian Bayesian classifiers. In recap, in future we focus on trying more robust unsupervised methods of data labeling such as clustering, K-means and other unsupervised learning takes which may improve our results.

References

[1] R. Cont, "Empirical properties of asset returns: stylized facts and statistical issues," *Quantitative finance*, vol. 1, no. 2, p. 223, 2001.

[2] P. J. Brockwell and R. A. Davis, *Introduction to time series and forecasting*. Springer, 2002.

[3] C. Alvaro and S. Jaimungal, *Machine Learning and Algorithmic Trading*. Unpublished Manual, 2023.