AN EARLY WARNING SYSTEM FOR ANOMALY DETECTION

FINTECH COURSE FINAL PROJECT

MSC MATHEMATICAL ENGINEERING 2021-2022

POLITECNICO
MILANO 1863

Alessandro Del Vitto
Raffaella D’Anna
Michele Di Sabato
Andrea Puricelli
Rita Numeroli

OUR SOLUTION

The key steps we followed to tackle the problem of anomaly detection were:

- Develop a robust **feature selection algorithm** : given the large number of features we had at our disposal, we implemented some feature selection algorithms to be able to work with simpler and more interpretable models. Two statistical approaches were used to perform feature selection after making them stationary. We removed the features that did not add much information to our model either because they were too correlated (hence they held the same information) or because they didn't seem to have a strong relevance from the theoretical-financial point of view.
- Implement a **distribution-based method** using **copulas** and infer the probability for a sample to be an anomaly with a Borderline-SMOTE oversampling modified k-NN algorithm.
- **Optimize the recall metric**: we set this as our main goal since from a business point of view we deemed it more important to catch a crisis and avoid false negatives (even if this may lead to some false alarms) rather than to miss an important event that may have catastrophic implications.

INTRODUCTION

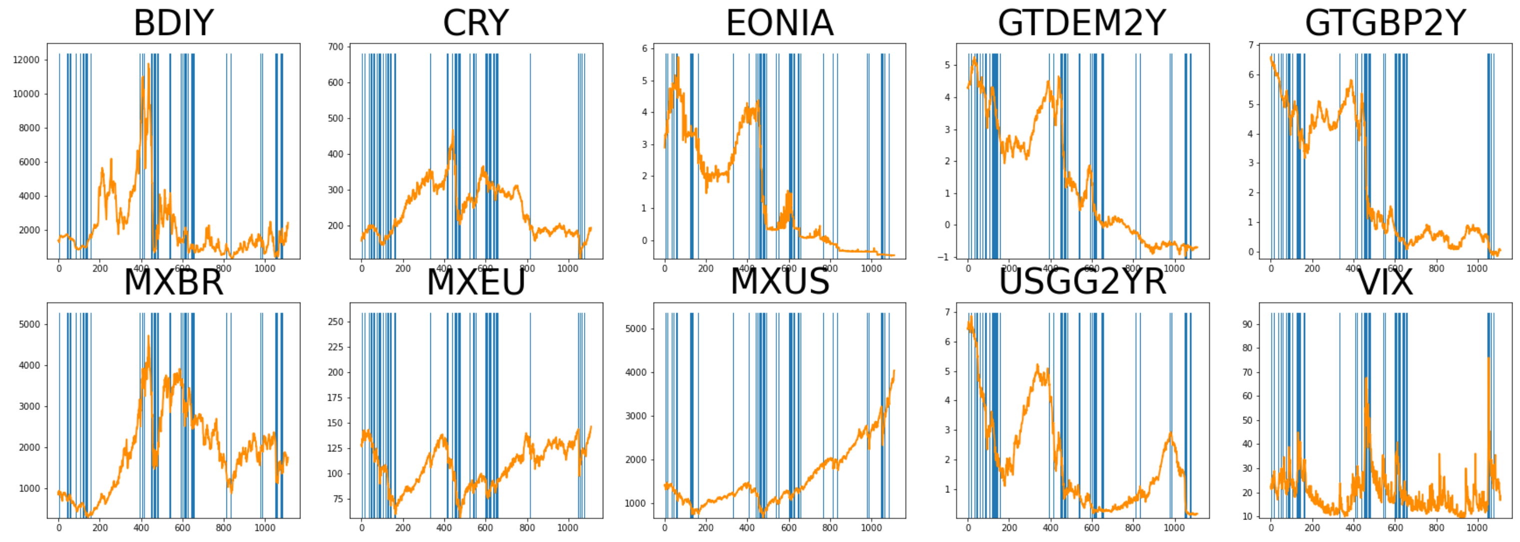
To improve financial performance and prevent risks, it's important to detect anomalous behavior in the Financial Markets, which have a periodical tendency to crash.

- During normal periods or **risk-on** periods, investors have a high-risk appetite and bid up the prices of risky assets in the market
- During a crisis, or **risk-off** periods, risk premia and financial assets exhibit anomalous behavior; investors become more risk-averse and sell risky assets, sending their prices lower, and tend to gravitate toward lower-risk investments

Two approaches to anomaly detection:

- Classification: treat the task with supervised classification exploiting the labels "Normal market" and "Abnormal market"
- Distribution based: estimate the distribution of the normal markets and classify the outliers as anomalies.

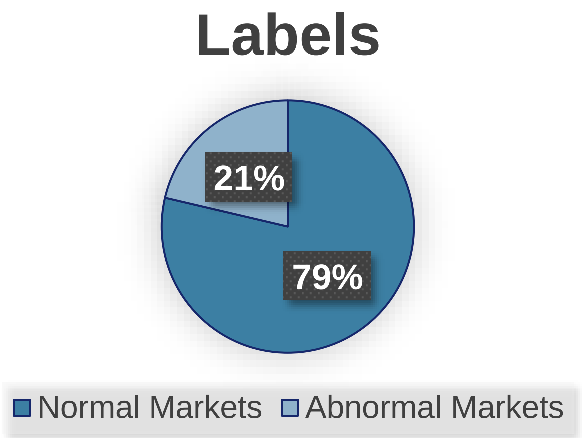
KEY CHALLENGES OF THE PROJECT



Plot of some features in the dataset. The vertical blue lines represent a **risk-on** period (anomaly).

Issues:

1) Unbalanced dataset with nearly an 80-20 ratio between classes
2) Too many features to be able to construct an effective and interpretable model
3) Non-stationary behaviors of the most part of the time series in the dataset



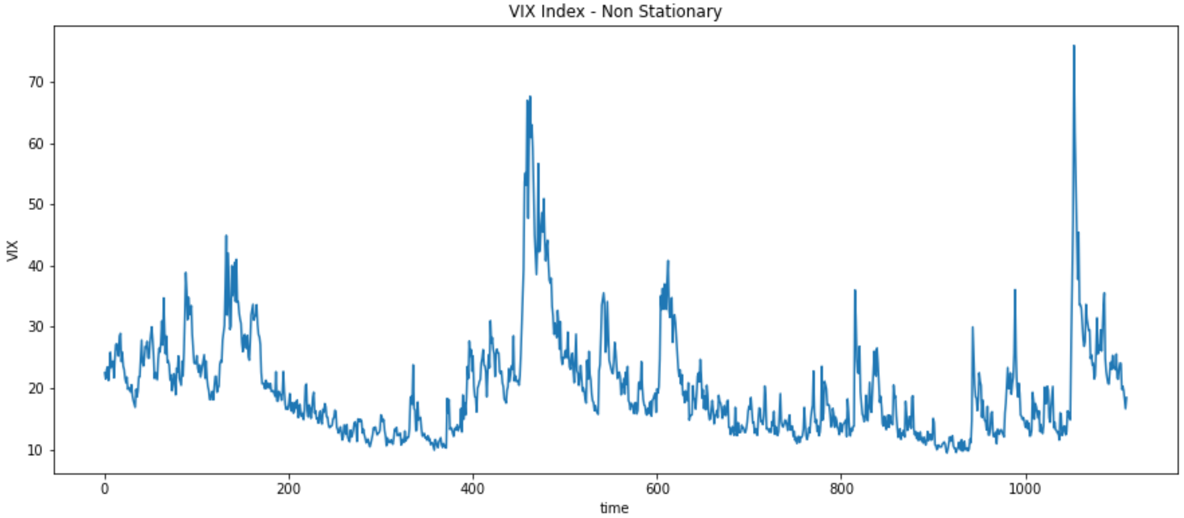
DATA PREPROCESSING

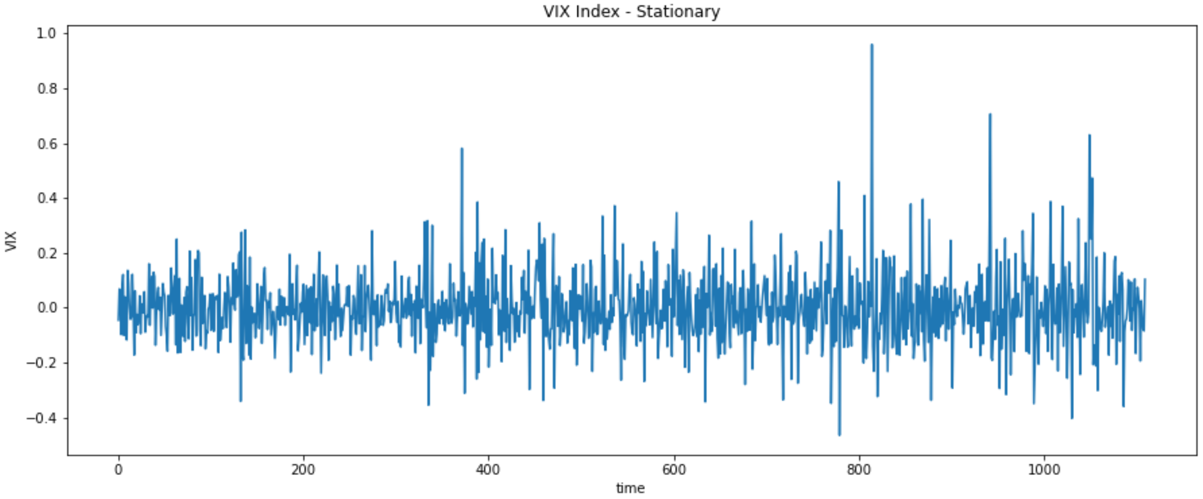
The original dataset features present a strong time dependence.
Since it is important to ensure and enforce **stationarity** to proceed with the Machine Learning model we applied some transformations to make the variables stationary.

The criterion we followed in particular is:

- If a feature is always **positive**, stock market indexes and currencies, for example, we transform it through the **logarithm** and consider the log-returns of that specific asset
- If it can become **negative**, for instance, the situation of interest rates, we work with the **first variation** of that quantity

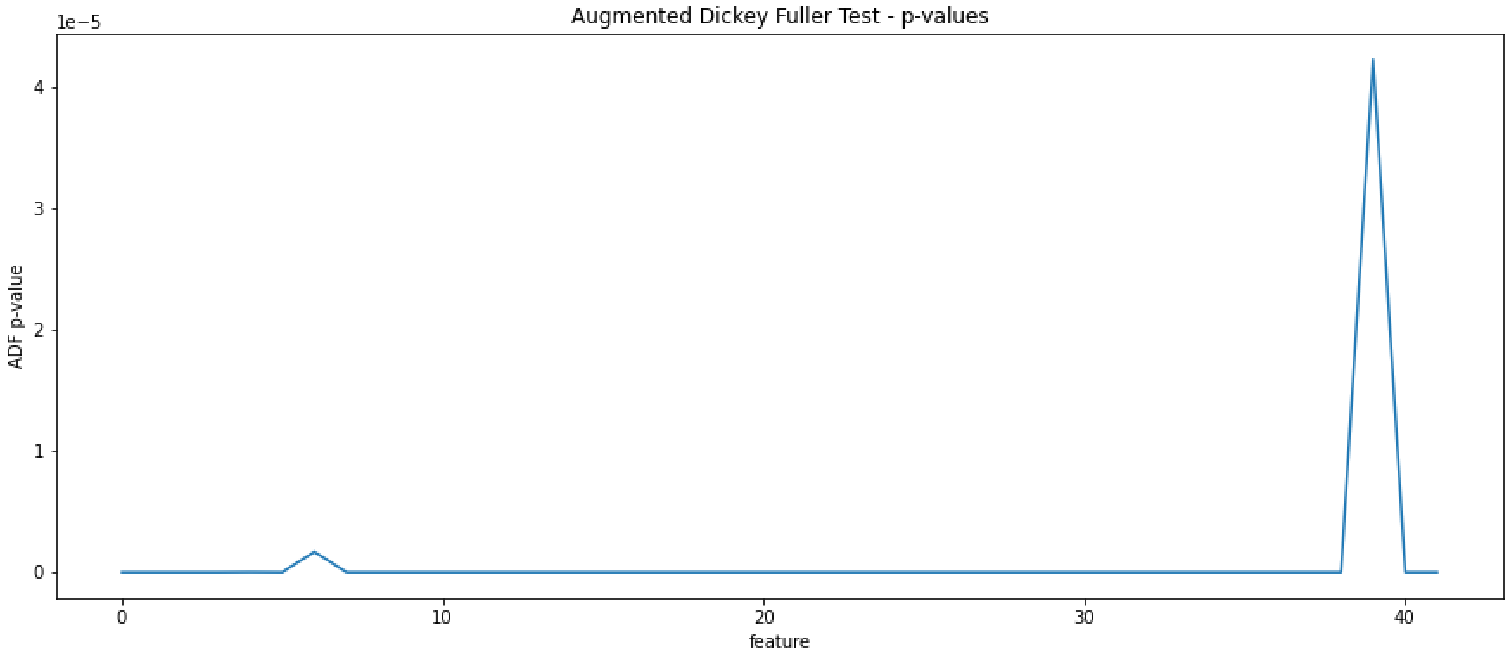
For example, this is the ''Implied Market Volatility Index'' variable before and after the log transformation:





DATA PREPROCESSING

We used the Augmented Dickey-Fuller test to check if stationarity was reached after this procedure. The p-values are small (less than 0.05) for all new features, so they are indeed stationary.

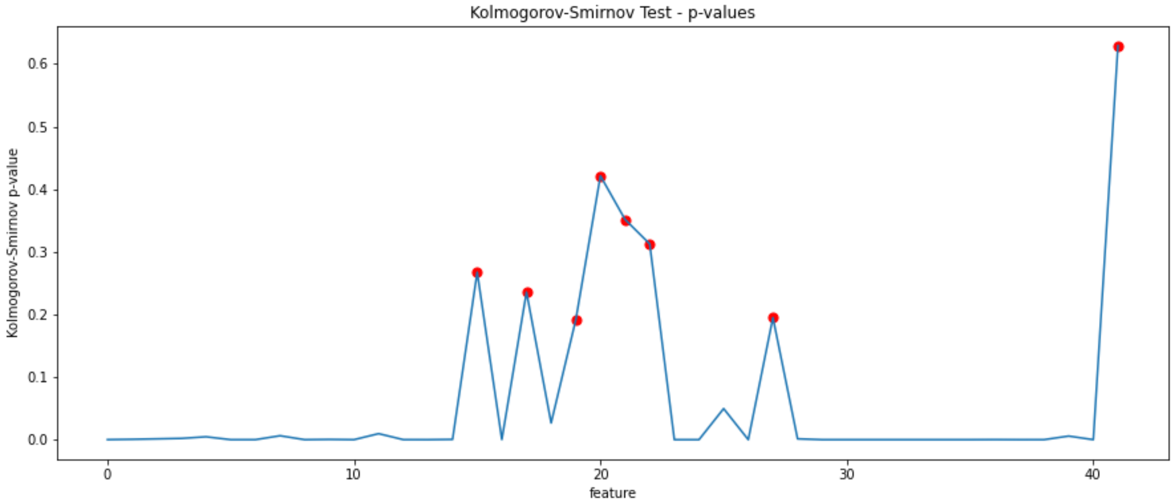


FEATURE SELECTION: STATEGIES IMPLEMENTED

1) Statistical test on the changes in distribution between **risk-on** and **risk-off** periods

We initially implemented a statistical test to assess whether features had different distributions during **risk-off** and **risk-on** periods, with the underlying idea that **variables that change distribution should be more relevant** to identify abnormal market conditions.

We chose to discard the features that had a p-value greater than a given threshold. In particular, we found that the spot gold price index, despite being considered a safe asset to revert to during a crisis, is not correlated with risk-on and risk-off periods.



The red dots correspond to features with a p-values > 0.05. They are:

- **GTITL10YR** : Italian bond market index 10yrs maturity
- **GTITL30YR** : Italian bond market index 30yrs maturity
- **GTJPY2YR** : Italian bond market index 2yrs maturity
- **GTJPY30YR** : Japanese bond market index 30yrs maturity
- **JPY** : Japanese stock market index
- **LF94TRUU** : Global inflation-linked total return index
- **LUACTRUU** : US corporate total return bond market
- **XAUBGNL** : Gold spot price for ounce

FEATURE SELECTION

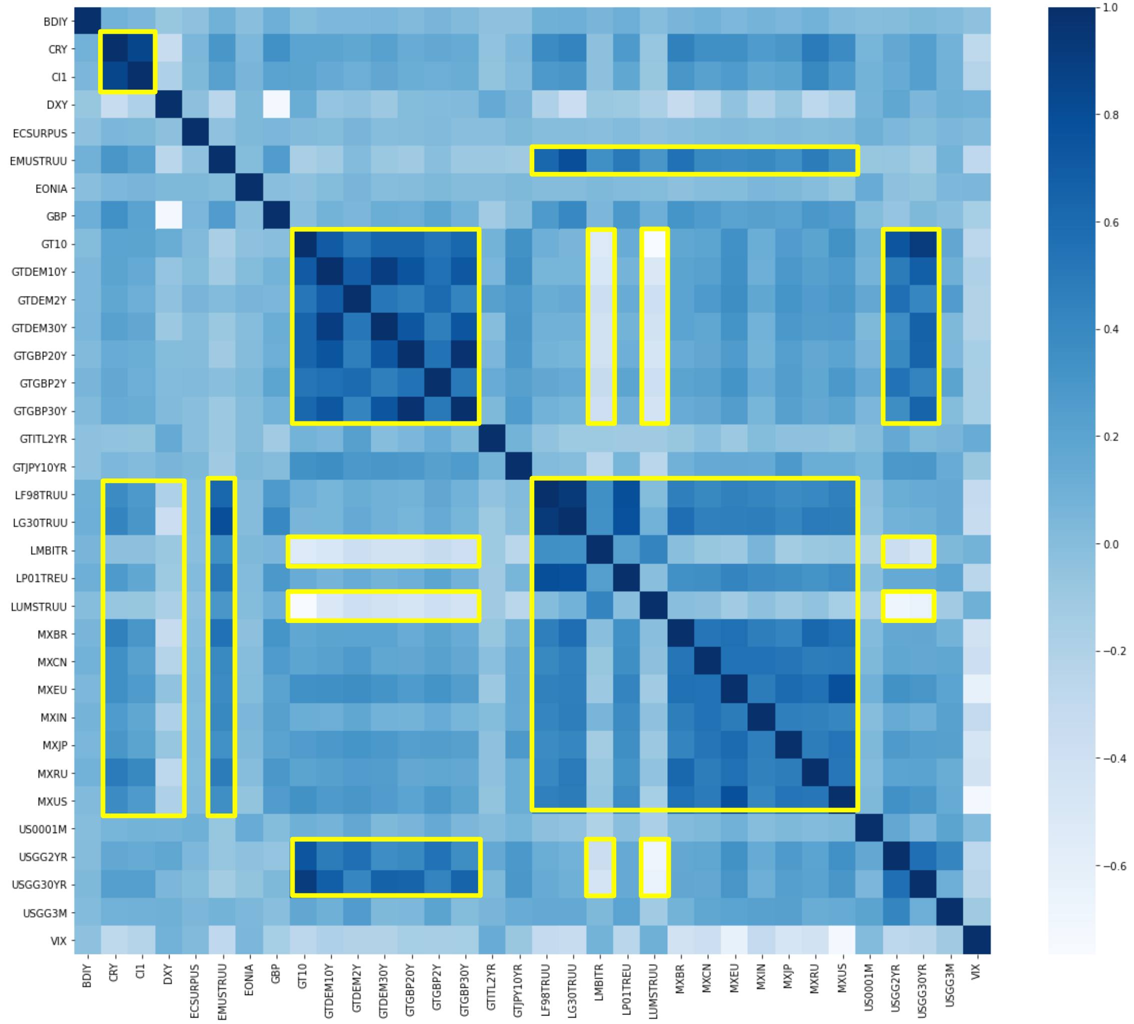
2) Correlation Study of the different features

We then decided to further decrease the number of features required by our model by getting rid of some too **highly correlated groups** of variables picking only just a reference one, chosen based on its financial meaning.

This was done with the idea to avoid introducing into the model redundant variables and keeping the model the simplest possible. In particular, we highlight in the heatmap on the left the highly correlated groups that we simplified, for instance:

- German and British bond market indexes
- Stock markets indexes
- Commodities markets indexes

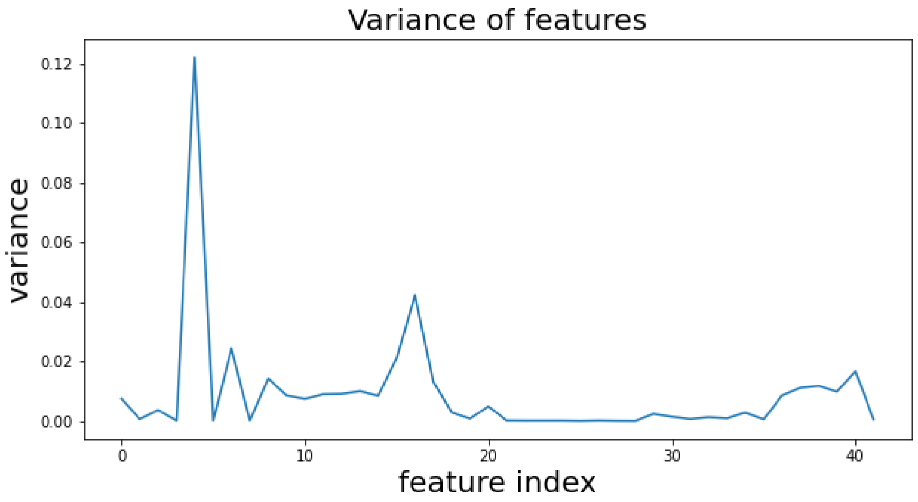
Notice that we tried to remove from the dataset groups that were both strongly positively correlated and strongly negatively correlated, those highlighted in the figure.

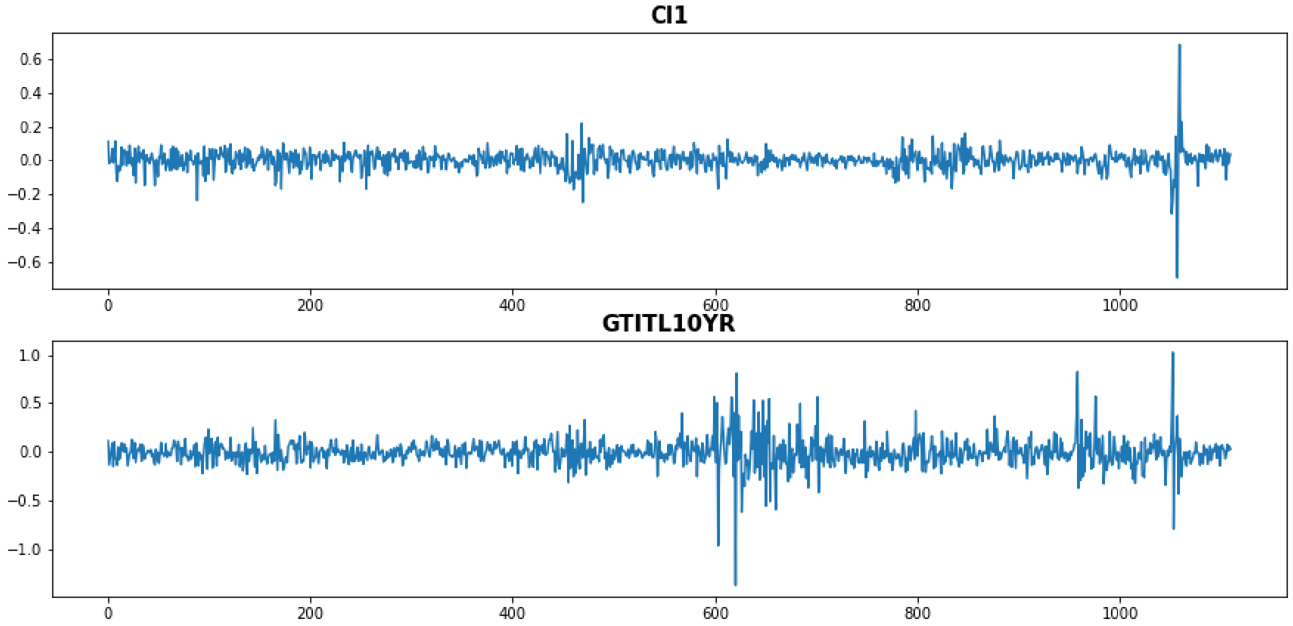


FEATURE SELECTION

3) Variance study

We discarded features that presented **low variance** and small correlation with the changes between **risk-on** and **risk-off** periods.





Example of two features with small (less than 0.01) variance.

FEATURE SELECTION

4) Financial Meaning

We also chose to focus our attention mainly on **European** and **American** indices, discarding the ones concerning other countries. Furthermore, we decided to keep only a subset of ten of the features selected up until now, considering only those we deemed to be financially meaningful and that we were able to justify from a theoretical point of view.

- **BDIY:** Baltic Exchange Dry Index
- **CRY:** Index performance for CoreCommodity
- **EONIA:** Euro OverNight Index Average
- **GTDEM2Y:** Germany Bund 2 Year Yield
- **GTGBP2Y:** United Kingdom Gilt
- **MXBR:** MSCI Brazil Index
- **MXEU:** MSCI European Index
- **MXUS:** MSCI U.S. Index
- **USGG2YR:** U.S. 2 Year Treasury
- **VIX:** Volatility Index

DISCARDED IDEAS FOR FEATURE SELECTION

- To classify the label of one week, we tried to include information coming from some of the previous ones by training a time series classification **LSTM** network and by including its result as a new neural-network-informed feature. Even though this approach is theoretically sound, the practical results of this model were quite poor. We think that this may be due to the **intrinsic complexity** of the task since it is challenging to understand the market evolution given the records of only a few weeks.

- We tried also to use the **targets of the previous two weeks** as additional features, however, the results were **too strictly linked** to this value: the model was not actually using any feature, other than the previous weeks' labels. Due to the small number of changes in momentum (from abnormal to normal markets and vice versa), the model was classifying every week's label as one of the previous week, obtaining good accuracy results.

- Motivated by the fact that **interest rates spreads** can be **good recession period indicators**, we have tried including these instead of the direct interest rates values. In particular, we know that when the interest rate curve has an inversion, ie. the 10yrs rate is lower than the 2yrs, historical data show that a recession is highly likely to happen in the following period*. This approach **did not seem to have** an **impact** on the results of the model, possibly because an inversion in the interest rates' curve might indicate that there will be a recession in the future, but it won't happen immediately. This dependence has a time lag that is not easy to capture.

* WRIGHT, JONATHAN H. "THE YIELD CURVE AND PREDICTING RECESSIONS." (2006)

COPULA

We tried many models starting with supervised classification techniques but we then preferred to pass to a density-based novelty detection algorithm exploiting copulas. In particular, we choose to split the dataset into three parts:

1. **Training set:** made of 80% of the total number of normal samples
2. **Validation set:** made of 10% of the total number of normal samples and the same number of abnormal ones
3. **Test set:** made of the remaining 10% of normal samples and the remaining portion of abnormal ones

These choices were motivated by the fact that in a novelty detection algorithm you train exclusively on normal data hoping to learn their true distribution and to be able later on to recognize abnormal samples from their low implied probability. Moreover, we decided to construct a balanced validation set where to tune the hyperparameter **ε** representing the probability threshold between being considered a **risk-on** and a **risk-off** period.

COPULA - RESULTS

We chose to focus on optimizing the **recall** performance metric while keeping the other indicators at a pretty good level with particular attention to precision.

This choice is motivated by the fact that we recognize as our main priority **detecting a crisis** while making sure to not have too many false alarms is just a secondary goal.

Notice that **avoiding false negatives** is crucial from a business point of view since not catching a crisis in advance may have serious implications.

We tried with two different copulas but we chose to keep the Gaussian one since it was computationally more efficient than the T-copula.

Performance Metric	Value
Recall	0.9933
Precision	0.6368
Accuracy	0.6356
F1 metric	0.7760

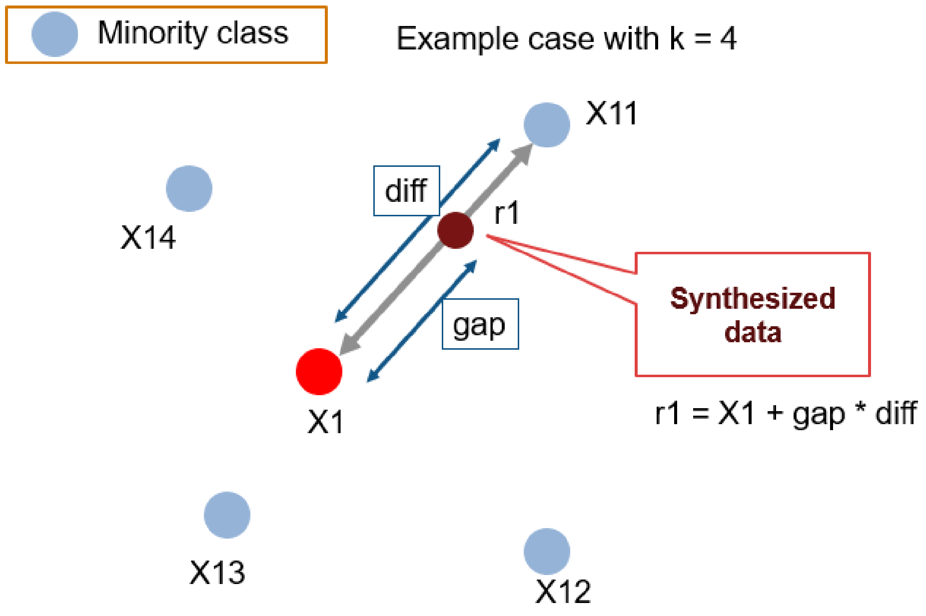
OVER-SAMPLING TECHNIQUES - SMOTE

In terms of our optimizing metric (**recall**), copulas are the best option, but they tend to classify too many points as anomalies.

To reduce the effect of the class imbalance, we worked on some over-sampling algorithms, namely **Naive random oversampling** (the standard sample with replacement of the points in the anomaly class), **SMOTE, Borderline SMOTE, ADASYN**. In particular, SMOTE works in the following way:

The SMOTE algorithm goes on until a desired ration between anomalies and normal points is reached. For each point in the anomaly class, SMOTE selects the K-nearest points belonging to the same anomaly class (K in our code is equal to 3) and adds a new anomaly as a random convex combination of the selected anomalous points and one random point contained in the neighborhood.

Borderline SMOTE and ADASYN are straightforward variations of SMOTE.



SMOTE – RESULTS

We decided to use k-NN coupled with SMOTE to offer a probability measure to the prediction produced by the copula, which always seems to favor the anomaly class: SMOTE should correct the class imbalance that would undermine the performance of k-NN. In particular, whenever we want to classify a point we first run the copula and use its prediction as the official one. Then we over-sample the training set with SMOTE to obtain an estimation of the probability for that point to belong to the class selected by the copula.

We tried all the over-sampling techniques and tuned parameter k, choosing the one which maximized a linear combination of precision and recall on the validation set.

Here are the results when using this procedure and SMOTE as an over-sampling technique:

```
confidence estimate: [0.4 0.  0.8 0.4 0.  0.  0.  0.  0.  0.2]
true labels: [0 0 0 0 0 0 0 0 0 0]
predicted labels: [1 1 0 1 1 1 1 1 1 1]
```

Using SMOTE

```
confidence estimate: [0.4 0.  0.8 0.6 0.2 0.  0.  0.2 0.  0.2]
true labels: [0 0 0 0 0 0 0 0 0 0]
predicted labels: [1 1 0 1 1 1 1 1 1 1]
```

Using Naive random over-sampling

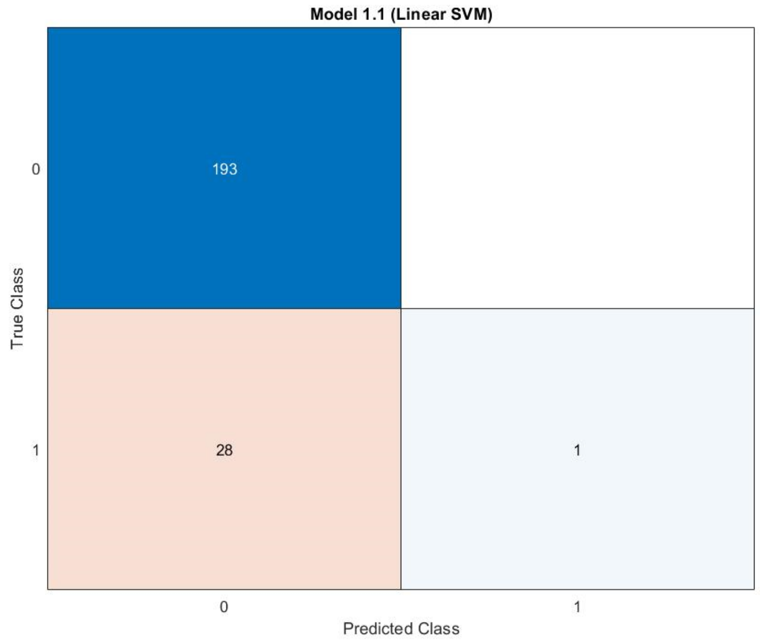
confidence estimate: estimate of the confidence of the copula prediction
true labels: true classes of the given samples
predicted labels: predicted labels by the copula model

OTHER MODELS

As explained in the previous slides we have considered many different models before reverting to copulas but none of them presented satisfying results.

Many models did not actually learn the relationship and ended up classifying almost all the test set samples as normal ones. For instance, we tried to implement:

- Isolation forests
- COPOD
- Artificial Neural Networks
- Many others from the Classification Learner App in Matlab



We report in the image on the left-hand side of the slide the confusion matrix on the test set of a linear SVM model. It is clear that the algorithm classifies all the samples into the dominant class of normal ones.

CONCLUSIONS

ISSUE #1 : Non stationary data

- We were able to stationarize the dataset features

ISSUE #2 : Too many features for interpretability

- We were able to construct a feature selection algorithm that allows for simple and easy-to-interpret models

ISSUE #3 : Unbalanced Dataset

- This has been the real challenge since most of the models we have tried ended up being biased towards one specific class and this is what made us revert to the novelty detection algorithm of copula that does not need to be trained on a balanced dataset and to extend our model with a Borderline-SMOTE oversampling k-NN algorithm

GOAL #1 : Optimize Recall

- We have been able to optimize the recall classification performance metric of our model

GOAL #2 : Maintain a good Precision level

- We have a fairly satisfying precision value in the end but we recognize that this is primarily due to the fact that our recall-optimized copula model has a tendency to overestimate the probability of being an anomalous sample.

POLITECNICO
MILANO 1863