

# Back Orders Prediction



Group T1E:

Nuno Bolas 20211052

João Magalhães 20211044

Mariana Teixeira 20211058

Maria Trindade 20211049

# Agenda

## 1. PROJECT CONTEXT & OBJECTIVES

Introduce the Back Orders Prediction challenge and the current project's objectives

## 2. METHODOLOGY

Describe the methodology followed to tackle the challenge and achieve the objectives defined

## 3. RESULTS & CONCLUSIONS

Present the best model results and the project's main findings

## 4. RECOMMENDATIONS & NEXT STEPS

Outline recommended course of action and further steps



# Organizations face the challenge of managing Back Orders while ensuring customer retention

In Inventory Management, Back Orders are a good strategy to avoid stock-outs while keeping healthy inventory levels.

However, organizations face the challenge to find the balance in managing Back Orders so customers do not wait too long for delivery and cancel their orders.

**How can organizations manage Back Orders in a competitive market while ensuring customer satisfaction and retention?**



# This project aims to find the best approach to predict Back Orders so organizations can plan and act accordingly

Project objectives:

- 1 Analyze different models to predict the occurrence of Back Orders for products
- 2 Compare the models and select the best one by evaluating the results obtained



# The methodology followed comprises 5 main steps:

## 2. Data Preparation

- Creation of **Dummy variables**;
- Missing values' filling with KNN and Mean Imputer;
- Outliers' handling with z-score;
- **Dataset standardisation** with StandardScaler
- **Imbalanced classes' treatment** with ADASYN, SMOTE and Tomek.

## 1. Data Understanding

- Initial exploration with Power BI;
- Dataset analysis with pandas-profiling;
- Identification of main dataset's issues:
  - **High correlation among variables**;
  - **Missing values**;
  - **Imbalanced classes**.

## 3. Feature Selection and Engineering

3 different methods were tested:

- **Feature selection** by joining the 3 highly correlated variables' groups into 3 separate variables;
- **Feature engineering** by adding 2 new variables created with PCA and SVD;
- Both methods combined.



# The methodology followed comprises 5 main steps:

## 5. Results and Evaluation

- Selection of the evaluation metrics: recall, precision, f1-score and specificity;
- Models performance assessment;
- Comparison of results and selection of the best model.


## 4. Modelling


- Resulting treated datasets analysis to find the best treated dataset;
- Grid-search analysis with the best dataset and the f1-score (micro) to find the best parameters;
- Models selected: Logistic Regression, Neural Networks (MLP), Random Forest, Gradient Boosting, Adaboost, Extra Trees and Balanced Bagging.




# The results showed that Balanced Bagging is the model with best performance in predicting Back Orders occurrence

Backorders would occur in 0.57% of the SKUs

**Precision: 0.70**  
Items that did not go on back order are classified correctly

**Specificity: 0.40**  
Items that did go on back order are classified correctly

**F1 Score: 0.68**  
Best for imbalanced classes

Confusion Matrix Balanced Bagging		
	Positive Prediction	Negative Prediction
No Back Order	500860	1653
Back Order	2281	1082



**According to the predictions obtained, backorders would occur in 0.57% of the SKUs thus, the risk of occurring a backorder is low for the majority of the items**

Main conclusions:

- Most of the SKUs of the dataset don't have backorders;
- Imbalanced datasets require the adoption of different ML techniques in order to find a model that predicts correctly the minority class (i.e., class with less observations);
- The market varies rapidly and new data is constantly increasing, so it is important to guarantee flexibility and scalability to the models. With our model, the variables and datapoints it can easily be tunable and scalable.

Managing back orders is a challenging task when defining the inventory strategy to follow, impacting the customer's service levels. Thus, the importance of having a model that predicts its occurrence, improving overall inventory strategy performance and customer satisfaction.





# Given the insights retrieved, SKUs with higher back ordering probability should be closely monitored

The following course of actions are suggested:

- Apply a more **conservative inventory level strategy** for the SKUs identified with a **higher probability of having backorders** and further monitor this SKUs in order to **avoid delays** and **customer dissatisfaction**;
- Further analyse the products with higher sales volume or higher revenue and define a specific marketing strategy for these products;
- Deployment and adoption of the model proposed to support inventory strategy;
- Future work includes exploring **misclassification costs** when **training the models** considering inventory costs and the revenue of the products and analyse the impact on performance.



# Thank you!

## Do you have any questions?



João Magalhães  
20211044



Maria Trindade  
20211049



Mariana Teixeira  
20211058



Nuno Bolas  
20211052



# Annexes



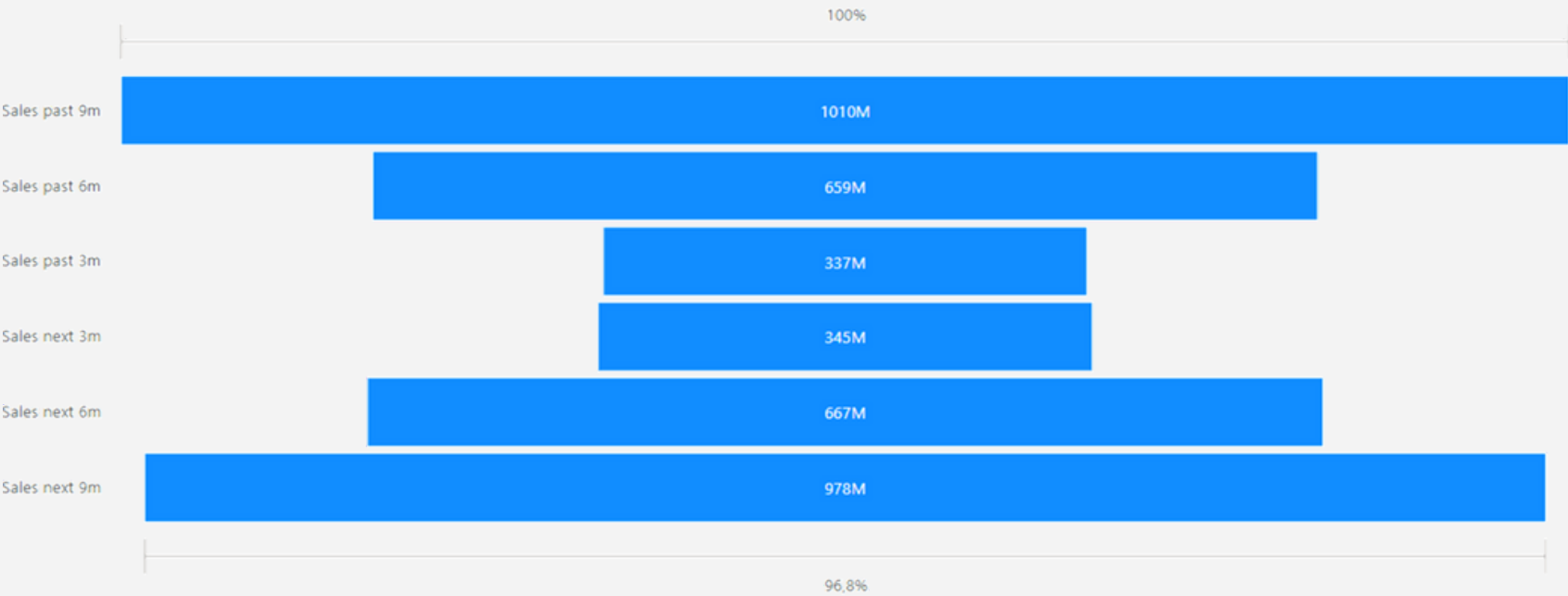
# 1. Methodology

## Data Understanding

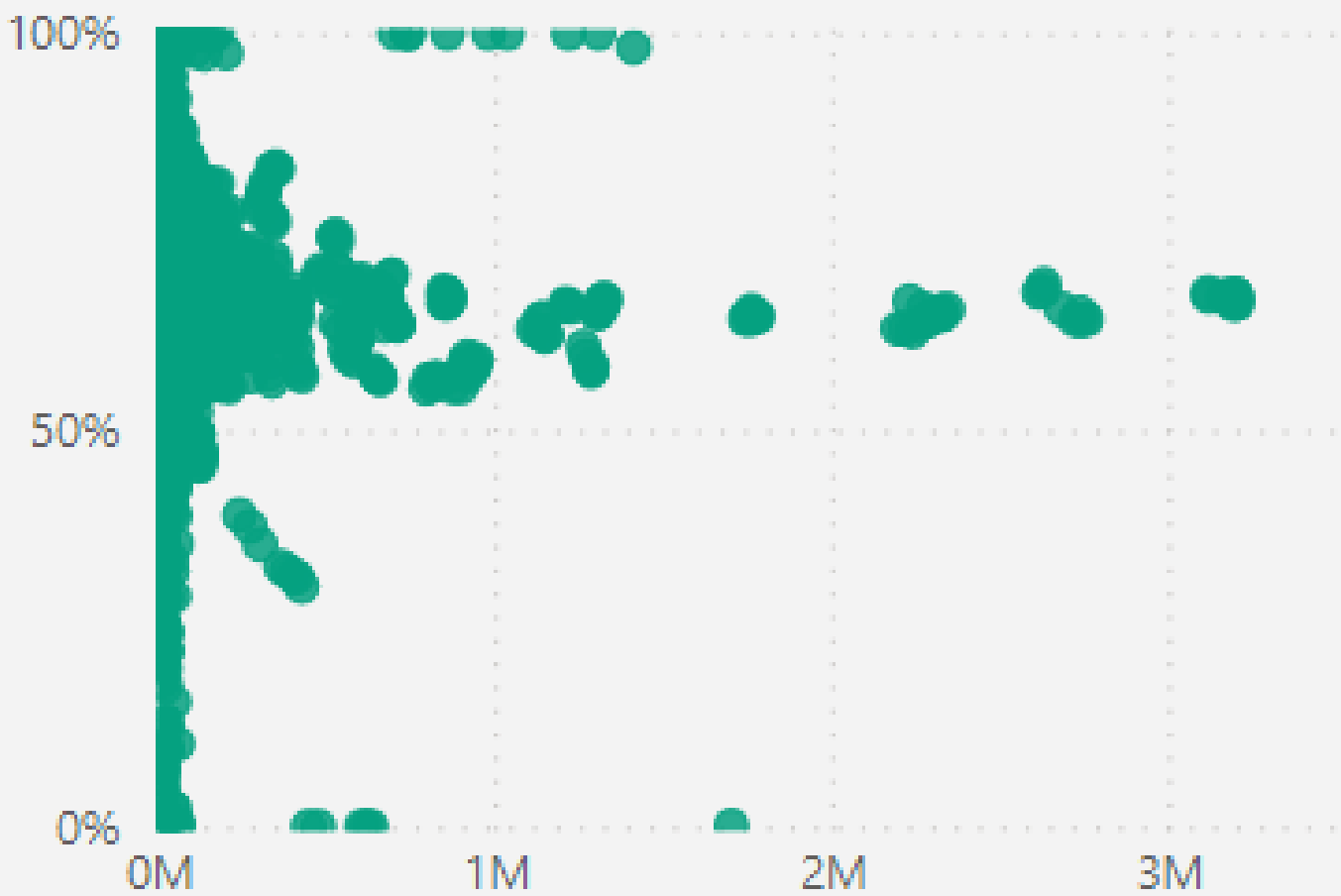
During the initial exploration of the dataset provided we discovered that:

- The dataset has a total of **1,93M SKUs**
- **34,7%** of the SKUs have no sales during the analysed time period.

Past sales are very similar to the forecasted sales showing that the dataset is homogeneous.



Comparing the sales in the last 6 months with the sales in the last 9 months, we noticed that for the majority of the SKUs, the last 6 months corresponds to around 67% of the last 9 monthsshowing that the sales behavior have little seasonality and variability.





# 1. Methodology

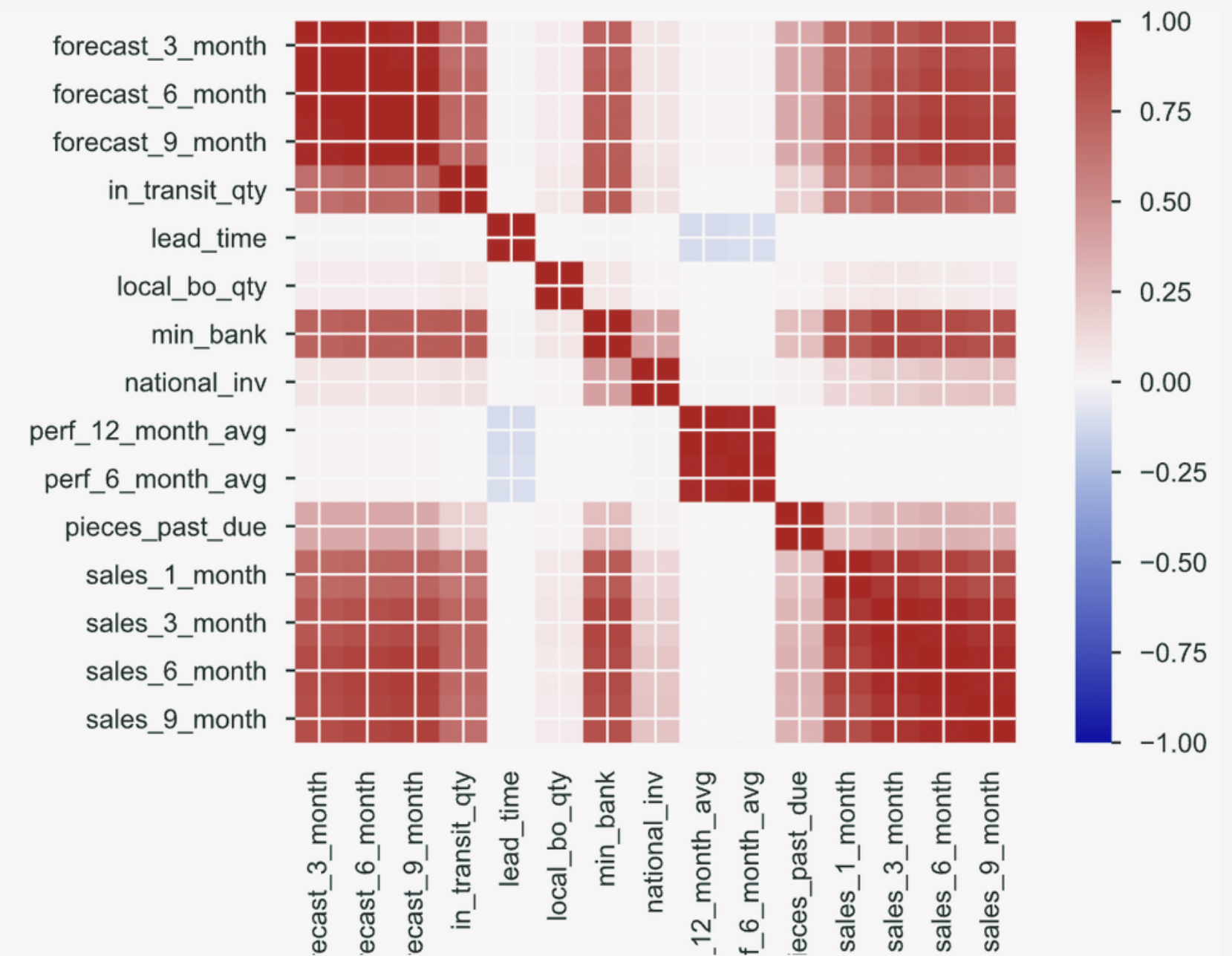
## Data Understanding

Through our analysis of the dataset, using Power BI and Pandas profiling we discovered the main issues of the dataset.

Starting with the correlation analysis, where we could understand that there were groups of **highly correlated variables** mainly:

- forecast\_3\_month , forecast\_6\_month and forecast\_9\_month
- sales\_1\_month, sales\_3\_month, sales\_6\_month and sales\_9\_month
- perf\_6\_month\_avg and perf\_12\_month\_avg

Additionally, forecast and sales columns are highly correlated which is normal since forecast is calculated based on past sales.



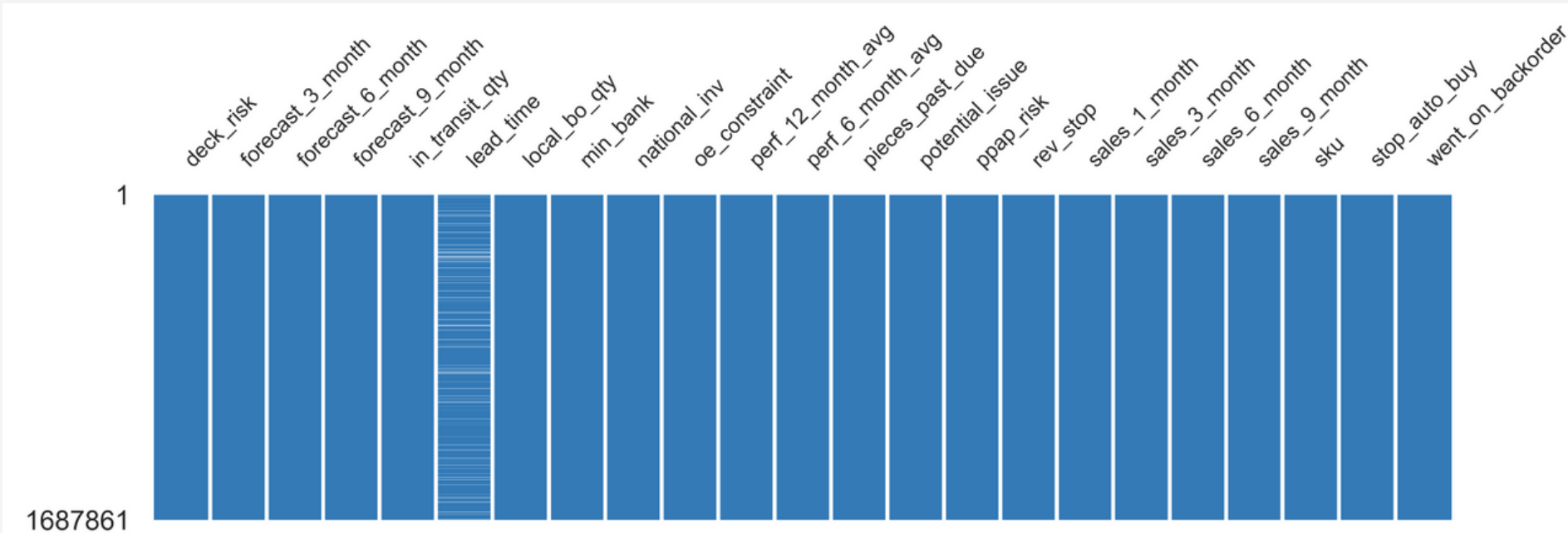
# 1. Methodology

## Data Understanding

Through our analysis of the dataset, using Power BI and Pandas profiling we discovered the main issues of the dataset.

The value lead time had 6% of the values missing, and there was a common missing row for all of the variables.

Finally, we discovered that the classes were highly unbalanced:



	Total	Percentage
Went backorder	1676567	99,3%
Did not go backorder	11293	0,67%



# 1. Methodology

## Data Preparation

To prepare the data for modelling we followed the next few steps:

- Starting by deleting the empty row and imputing the missing values. To impute the missing values, we used two different methods to test which would work best with this dataset. Starting with the univariate imputer with the mean and then followed by the KNN imputer. After testing against the models, we understood that the latter was the right option for this project;
- To deal with the outliers, since most operations are influenced by these, the z-score method was used, which means that 18.88% of unusual data points were removed;
- With the imputed datasets, we created dummy variables and proceeded to standardise the dataset. The chosen scaling method was Standard Scaler, which removes the mean and scales each feature/ variable to unit variance;
- The variables perf\_6\_month\_avg and perf\_12\_month\_avg had a high percentage (around 7%) of a single value (-99), which we considered to be a single value imputation for the missing values, as such, we proceeded to impute these columns with the KNN imputer as well.
- To conclude the preparation, we resampled the dataset with ADASYN (oversampling), SMOTE (oversampling) and Tomek+SMOTE (oversampling + undersampling), so we would have the same values for both classes, and the model could learn the minority class as well as the majority.



# 1. Methodology

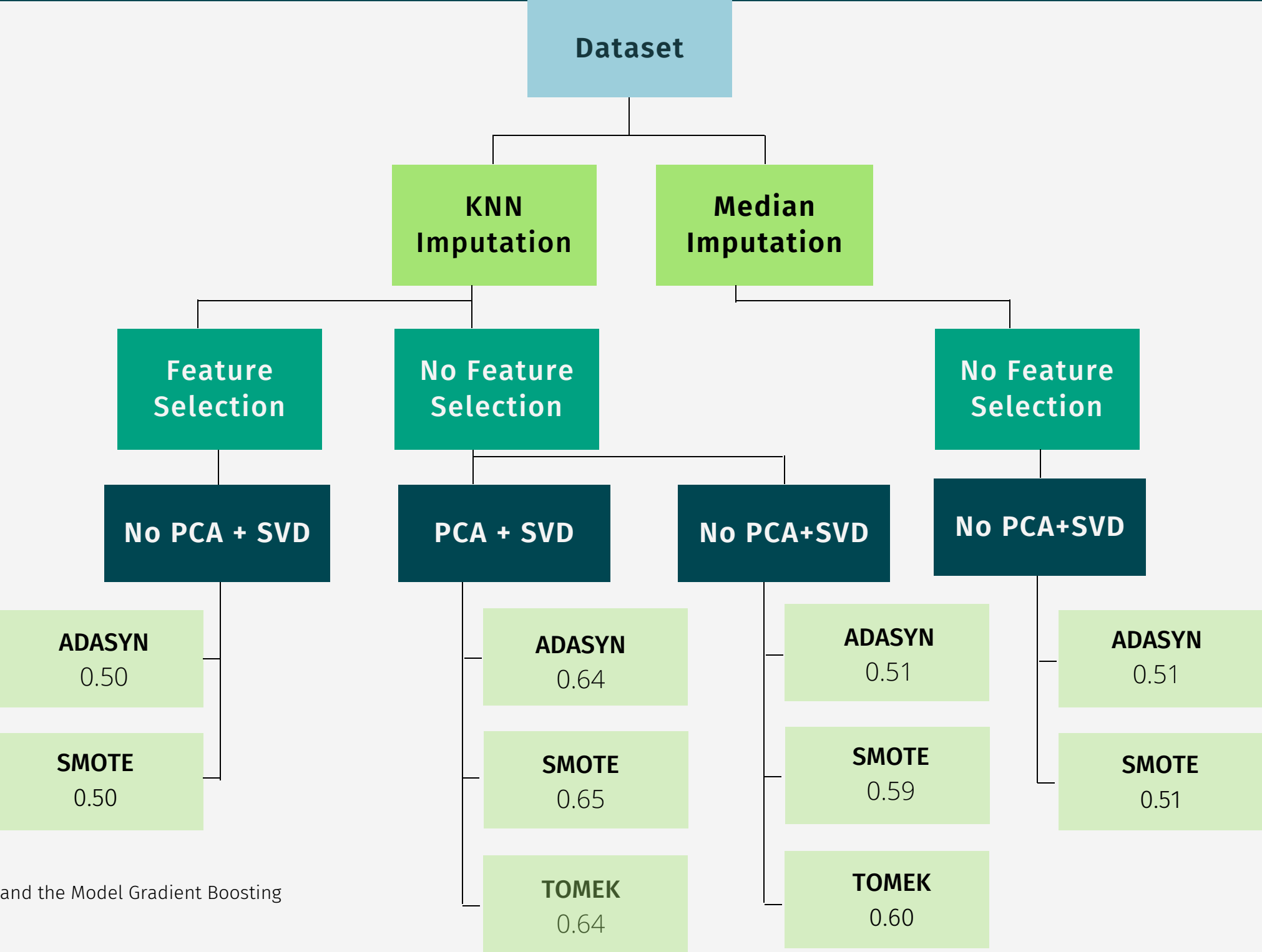
## Feature Selection and Engineering Tree

Data Preparation

Feature Selection

Feature Engineering

Balancing



\*Scores are calculated with the metric Precision and the Model Gradient Boosting



# 1. Methodology

## Feature Selection and Engineering

For better understanding of our process, we can take a look at our tree in the previous slide.

In order to selection the variables, we studied the correlation and Kolmogorov–Smirnov test for numerical features, as well as created the Stochastic/Probability Matrix for categorical features. As mentioned in data understanding section, we found that there were 3 groups of variables that demonstrated to be highly correlated so we decided to create one variable for each of the groups:

1. sales\_month (sales\_9\_month divided by 9): representing sales\_1\_month, sales\_3\_month, sales\_6\_month, sales\_9\_month;
2. forecast\_month (forecast\_9\_month divided by 9): representing forecast\_3\_month, forecast\_6\_month, forecast\_9\_month;
3. perf\_month (same as perf\_12\_month\_avg): representing perf\_6\_month\_avg, perf\_12\_month\_avg.

This dataset was tested in modelling, but unfortunately the results were worse than the dataset without feature selection, therefore, we continued with the latter one.

In feature engineering we created new variables using SVD and PCA, the intent for this was giving the model variables that would encapsulate the information of all of the other variables and lead it to the best prevision.



# 1. Methodology

## Modelling

For the modelling section, we started by experimenting with the different datasets, quickly understand that the feature selection dataset was not performing as well as the others, so we excluded it from the training. As for the the PCA + SVD dataset, we found that it was performing well, so we kept it as the final training dataset. As mentioned in the previous slide, the KNN imputer proved to be a better imputation method.

With the selected dataset (PCA + SVD with KNN imputation), the Gridsearch was conducted for the smote balancing method. For each Gridsearch different parameters were used, which, through our research, we found to be the most effective. The scoring metric was f1-micro, which optimizes the model according to the minority class.

The final models were: Logistic Regression (LR), Neural Networks (NN), Random Forest (RF), Gradient Boosting Decision Tree (GB), Adaboost (AB), Extra Trees (ET) and Balanced Bagging (BB).

	Hyperparameters tested
GB	n_estimators, max_depth
ET	n_estimators, max_depth
RF	n_estimators,max_depth
NN	hidden_layer_sizes, activation, solver, learning_rate
AB	n_estimators, learning_rate
LR	-
BB	n_estimators, max_features, max_samples, warm_start



# 1. Methodology

## Results

With our Gridsearches complete, and the best hyperparameters found, we proceeded to the training of our balanced dataset to form the model which we then applied to our validation set.

We experimented with different metrics, though for our project we found specificity to be the most important as we were trying to predict the true negative class. We paid special attention as well to the macro averaged precision and f1-score, as this was a highly imbalanced dataset.

In the end, our best model was Balanced Bagging which is understandable as it performs a random undersampling strategy on the majority class within a bootstrap sample so that the two classes are balanced.

In this case, we can see a lift on macro averaged f1-score, specificity and precision, therefore this was our final model.



Results for the  
Precision and  
Specificity  
metrics

	KNN imputation and No Feature Selection				Mean imputation and No Feature Selection		No Feature Selection
	PCA + SVD			No PCA + SVD	No PCA + SVD		PCA + SVD
	Adasyn	Smote	Tomek	Smote	Adasyn	Smote	Smote
GB	0.64 / 0.29	0.65 / 0.30	0.64/ 0.28	0.59 / 0.19	0.58 / 0.19	0.62 / 0.24	0.50 / 0.01
ET	0.52 / 0.04	0.52 / 0.04	0.51 / 0.03	0.51 / 0.02	0.51 / 0.02	0.51 / 0.02	-
RF	0.54/ 0.09	0.54 / 0.09	0.54 / 0.09	0.54 / 0.08	0.54 / 0.08	0.54 / 0.08	-
NN	0.00 / 0.01	0.00 / 0.01	0.5 / 0.01	0.52 / 0.03	0.52 / 0.03	0.52 / 0.03	-
AB	0.52/0.04	0.52/0.04	0.53/0.06	0.52 / 0.04	0.52 / 0.04	0.52 / 0.04	-
LR	0.5 / 0.01	0.5 / 0.00	0.50 / 0.01	0.51 / 0.01	0.50 / 0.01	0.50 / 0.01	-
BB		0.70 / 0.40	-	-	-	-	-

## II. Definitions and applications to our project

**Unbalanced data:** data in which observed frequencies are very different across the different possible values of a categorical variable. Basically, there are many observations of some type and very few of another type. In our dataset, our majority class is when it didn't go backorder; and our minority class when it did go back order. This is a problem because we want to predict when we will have a backorder, with very little data about this occurrence.

**High correlation:** means that two or more variables have a strong relationship with each other, meaning that you can predict one variable using second predictor variable. This is called the problem of multicollinearity. This results in unstable parameter estimates of regression which makes it very difficult to assess the effect of independent variables on dependent variables.

**Outliers:** data points that are far from other data points. In other words, they're unusual values in a dataset. Outliers are problematic for many statistical analyses because they can cause tests to either miss significant findings or distort real results

**Missing values:** this means that one or more of variable doesn't have the same number of values than the others. This is a problem because most models can't run with variables with different numbers of entries.

## II. Definitions and applications to our project

**Missing values imputation:** The imputer is an estimator used to fill the missing values in datasets.

- Mean Imputer: It's one of many univariate imputations, which means that the missing values are imputed by one value only, in this case, it would be the the mean of each variable;
- KNN Imputer: It's one of the most known multivariate imputations, unlike the one above, it fills in the missing values with multiple values, instead of a fixed one. This model identifies 'k' samples in the dataset that are similar or close in the space. Then it uses these samples to estimate the value of the missing data points. Each sample's missing values are imputed using the mean value of the 'k'-neighbors found in the dataset.

We chose the KNN imputer, as it takes into account other features in the dataset when performing imputation instead of an only column. The scores with KNN dataset proved to be better than with the mean imputation.

**Z-score for Outliers:** The Z-score measures how far a data point is away from the mean as a signed multiple of the standard deviation. Large absolute values of the Z-score suggest an anomaly. This is one of the simplest and one of the most popular techniques for outlier detection that works well for most usecases.





## II. Definitions and applications to our project

**PCA (Principal component analysis)** : It's a technique for reducing the dimensionality of data. It increases interpretability yet, at the same time, it minimizes information loss.

**SVD (singular value decomposition)** : SVD is basically a matrix factorisation technique, which decomposes any matrix into 3 generic and familiar matrices.

In our case, we used these techniques as a feature engineering method, creating of an extra variable that would encapsule the information of our other variables, hopefully giving the model useful information for the classification of our target.

**Clustering the minority class:** technique that groups a set of entries together in such a way that objects in the same group are more similar to each other than to those in other clusters.

In our case, we decided to do this as a treatment for the imbalancing on our dataset, so we clustered the minority class.



## II. Definitions and applications to our project

**Gridsearch:** technique to search through the best parameter values from the given set of the grid of parameters. It is basically a cross-validation method. The model and the parameters are required to be fed in. Best parameter values are extracted and then the predictions are made.

**Re-sampling:** resampling is the creation of new samples based on one observed sample.

- ADASYN: is an algorithm that generates synthetic data, and its greatest advantages are not copying the same minority data, and generating more data for “harder to learn” examples.
- SMOTE: SMOTE stands for Synthetic Minority Oversampling Technique. SMOTE is an improved method of dealing with imbalanced data in classification problems.
- SMOTE-TOMEK + Clustering : First uses SMOTE algorithm to generate new samples from minority class and then downsample data with the help of clusters centroid. The main advantage here is that we’re maintaining diversity during both upsampling and down sampling phases which leads to higher quality results.





## II. Definitions and applications to our project

- **Logistic Regression:** statistical technique used to predict the relationship between the dependent variable and the independent variable;
- **Neural Networks:** set of algorithms, modelled loosely after the human brain, that are designed to recognise patterns.
- **Random Forest:** consists of many decisions trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.
- **Gradient Boosted Decision Tree:** Similar to Random Forest, but aggregates the results of each decision tree along the way to calculate the final result.
- **Adaboost:** ensemble learning method (also known as “meta-learning”) which was initially created to increase the efficiency of binary classifiers. AdaBoost uses an iterative approach to learn from the mistakes of weak classifiers, and turn them into strong ones.
- **Extra Trees:** an ensemble supervised machine learning method that uses decision trees and is used by the Train Using AutoML tool.
- **Balanced Bagging:** A Bagging classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction.

## II. Definitions and applications to our project

- **Accuracy:** measures how often the classifier correctly predicts. Accuracy can be defined as the ratio of the number of correct predictions and the total number of predictions.
- **Precision:** explains how many of the correctly predicted cases turned out to be positive. It is useful in the cases where false positives are more important to understand in the context than false negatives.
- **Recall:** also known as Sensitivity, explains how many of the actual positive cases we were able to predict correctly with our model.
- **Specificity:** measures the proportion of actual negative cases that have gotten predicted as negative by our model. It is the ratio of true negatives to all negatives. Is also known as the true negative rate.
- **F1-Score:** is the harmonic mean of precision and recall. It gives a combined idea about Precision and Recall metrics.
- **Confusion Matrix:** is a performance measurement for the machine learning classification problems where the output can be two or more classes. It is a table with combinations of predicted and actual values. A Confusion Matrix is used to describe the performance of a classification model on a set of the test data for which the true values are known.



### III. Bibliography

Dalvi, C. (2021, June). Backorder Prediction using Machine Learning. Retrieved from Medium:

<https://chinmaydalvi.medium.com/backorder-prediction-using-machine-learning-cbe2a7d2cfa4>

Elabd, M. (2020, February). Imbalanced dataset, Here are 5 regularization methods which can help. Retrieved from Medium:

<https://medium.com/@arch.mo2men/imbalanced-dataset-here-are-5-regularization-methods-which-can-help-5acdb8d324e3>

Islam, S., & Amin, S. H. (2020). Prediction of probable backorder scenarios in the supply chain using Distributed Random Forest and Gradient Boosting Machine learning techniques. Journal of Big Data .

Korstanje, J. (2021, August). SMOTE. Retrieved from Towards Data Science: <https://towardsdatascience.com/smote-fdce2f605729>

LinWei-Chao, Tsai, Chih-Fong, Hu, Ya-Han, Jhang, & Jing-Shang. (2017, October). Clustering-based undersampling in class-imbalanced data. Information Sciences, pp. 17-26.

Mohit, P. (2021, September). Predicting Material Backorders in Inventory Management. Retrieved from Medium:

<https://medium.com/analytics-vidhya/predicting-material-backorders-in-inventory-management-90e4d0ece6ba>

Raja, S. (2021, March). Backorder Prediction. Retrieved from Medium: <https://medium.com/analytics-vidhya/backorder-prediction-d4f1c5362f18>

Santis, R. B., Aguiar, E. P., & Goliatt, L. (2017). Predicting Material Backorders in Inventory Management using Machine Learning. 4th IEEE Latin American Conference on Computational Intelligence.

