Universidade Federal do Paraná
Setor de Ciências Exatas
Departamento de Estatística
Programa de Especialização em *Data Science* e *Big Data*

Márcio Venâncio Batista

A Comparison of Different Machine Learning Strategies for DIFOT Prediction

Curitiba
2020

Márcio Venâncio Batista

A Comparison of Different Machine Learning Strategies for DIFOT Prediction

Monografia apresentada ao Programa de Especialização em Data Science e Big Data da Universidade Federal do Paraná como requisito parcial para a obtenção do grau de especialista.

Orientador: Prof. Dr. Luiz Eduardo S. Oliveira

Curitiba
2020

Especialização em Data Science e Big Data
Universidade Federal do Paraná
dsbd.leg.ufpr.br

A Comparison of Different Machine Learning Strategies for DIFOT Prediction

Márcio Venâncio Batista[1]
Prof. Dr. Luiz Eduardo S. Oliveira[2]

Abstract

Improving results by optimizing processes execution is one of the major companies objectives. And for many of these companies, the main point to achieve better results is the good maintance of supply chain management. Optimizing the supply chain can help to ensure the customer satisfaction which can receives their order on the scheduled date and in the right quantity. Companies are constantly studying how to improve their performance in order to achieve this ideal scenario. Many management procedures are studied and many measurement techniques are tested, all of them with the aim of reducing costs and improving customer satisfaction. A widely used way that helps organizations to measure whether they are achieving this goal is the use of a KPI called Delivery In Full, On Time (DIFOT). There are several studies that deal with logistics and delivery optimization aiming to minimize delary risks and production problems, leadning to a better performance. In addition, other studies that aim to provide demand forecast and production capacity, are also widely used. These initiatives can helps companies to achieve DIFOT. A field not so explored until now is the appliance of data science techniques to assist in the process of optimizing all the production logistics and supplies delivery. This paper proposes the use of data science tools together with artificial intelligence and machine learning techiniques to bring an innovative vision to solve problems supply chain and process. This study uses a priori data, that is, sales orders features that are obtained right from the begining of sales process, and then, and one of the objetives is, to identify the correlation between features that are related to the occurrence of DIFOT. Subsequently, these features were used in machine learning models in order to identify how predictable the occurrence of DIFOT is when a new sales order is created. The results obtained demonstrate that it is possible to define predictive models that can assist in the decision-making process of organizations to ensure improvement in the supply chain and, consequently, improving KPI DIFOT performance.

[1]Aluno do programa de Especialização em Data Science & Big Data, marcio.venancio@ufpr.br.
[2]Professor do Departamento de Estatística - DEST/UFPR.

Keywrods: DIFOT, On time, In Full, Supply Chain, Big Data, KPI.

1 Introduction

The best way to describe the supply chain that is the series of steps and processes by which value is added to a product, and through which it is delivered to an end customer. This naturally leads to an effective tool for understanding the supply chain[1].

The role of manufacturing industry is to create wealth by adding value and selling products. Common to all manufacturing companies is the need to control the flow of material from suppliers, through the value adding processes and distribution channels, to customers. The supply chain, as shown in Figure1, is the connected series of activities which is concerned with planning, coordinating and controlling material, parts and finished goods from suppliers to the customer. It is concerned with two distinct flows through the organisation: material and information. The scope of the supply chain begins with the source of supply and ends at the point of consumption. It extends much further than simply a concern with the physical movement of material and is just as much concerned with supplier management, purchasing, materials management, manufacturing management, facilities planning, customer service and information flow as with transport and physical distribution.

In this scenario, as in any business activity, supply chain operations need to focus doggedly on improvement to compete in the market place, but how do you know if your supply chain performance is satisfactory, or if it is getting better, or worse? To help in this process, KPIs can be used. KPI stands for *Key Performance Indicator*, and can be defined as a practical and objective measurement of progress towards a predetermined goal or against a required performance standard.

Using KPIs for performance measurement ensures that the process is always being evaluated against a static benchmark. That makes fluctuations immediately visible, and if performance moves in the wrong direction, a quickly respond can be provided[3].

The Most Important Supply Chain KPI Metric is *Delivery in Full on Time* (DIFOT). This metric is the ultimate measure of the performance of your supply chain. The

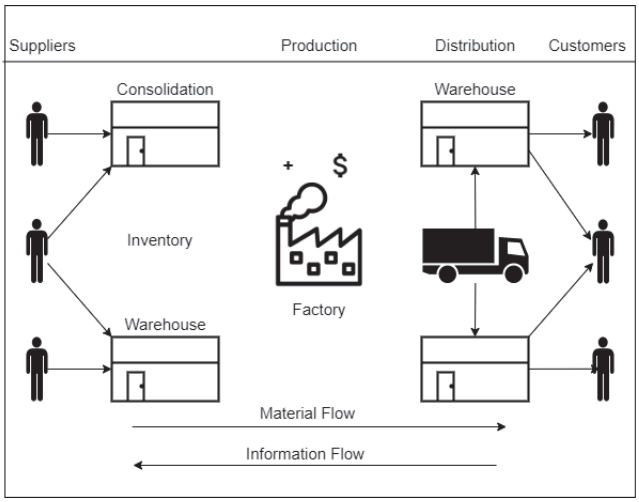

Figure 1: The scope of the Supply Chain [2]

purpose of a supply chain is to deliver to the customer the products they need in the quantity they need, when they need them. DIFOT directly measures how well a supply chain is fulfilling this core purpose. It would be inconceivable for a business not to measure profit or cash flow. It should be equally inconceivable for a manufacturing or distribution business not to manage DIFOT. DIFOT in its simplest form is simply the ratio of the number of orders that were delivered on time (NODOT), with all the ordered items supplied in the quantity required on the day that the customer required them, to the total orders shipped (TOS).

$$DIFOT = \frac{NODOT}{TOS} \cdot \frac{100\%}{1}$$

The DIFOT measurement reflects the actual performance of a company in the eyes of the customer. In many cases, particularly when the customer is a large business, they will measure company's DIFOT performance themselves [4]

Time is the most critical dimension in supply chains. The key to delivering products on time and in full is developing a process and a supply chain that can respond to customer demand as fast as or faster than the customer requires or expects. Lead time is the key measure of time.

The most important lead time is your expected customer order lead time. This is the length of time the customer is prepared to wait from placing their order to receiving their goods. This can vary enormously, depending on the type of product, the way it is distributed, the customer's location, market expectations (i.e., what the competitors do), and the degree to which a product is customized to meet a customer's individual needs.

One of the greatest challenges for DIFOT is hard to know what our customer will need and when they will need it to forecast customer need of the product. Most businesses take time to respond to changes in customer demand. This is because of lead times in production and lead times from their suppliers [1].

That said, it is possible to realize DIFOT KPI importance in supply chain management and how it is related to customer satisfaction and loyalty. Far beyond obtaining the demand forecast, and mitigating other problems in warehouses, the occurrence of DIFOT can be compromised by issues related to the transportation of goods, such as the chosen carrier, vehicle used for transportation, distance to be traveled between the distribution center and the customer, weather conditions, etc. In summary, just a forecast, however close to reality, is not enough to guarantee DIFOT.

Given the relevance of DIFOT KPI in Supply Chain, the aim of this work is to contribute to the framework of KPIs that helps to increase the performance of Supply Chain Management. In this paper is proposed the use of machine learning to predict the occurrence of DIFOT for a sales order using a priori data, that is, data which is obtained during the creation of the sales order. With that it is possible to give an answer if the sales order, will achieve the DIFOT, giving the manager the possibility to mitigate some weakness that may cause the failure of that sales order, this can contribut to the DIFOT final company score.

The DIFOT of a sales order is reached when the customer receives their products at the desired time and in the desired quantity, that is, *On time* and *In Full*. The success of these two indicators ends up causing DIFOT. Based on this understanding, the research focused on the interpretation of the problem in two different ways, the multiple class and binary forms or classification. About the data used for models design was obtained from real cases provided by a large multinational food company.

2 Literature Review

Supply Chain Management is defined by Christopher[5] as the management, across and within a network of upstream and downstream organisations, of both relationships and flows of material, information and resources.

Supply chain professionals are struggling to handle the large structured and unstructured data. They are surveying new techniques to investigate how data are produced, captured, organised and analysed to give valuable insights to industries. Big data analytics is one of the best methods to help them in overcoming their problem [6].

Together with Industry 4.0, Internet of Things (IoT), and other digital technologies, a massive amounts of data are now collated from several sources, including enterprise resource planning (ERP) systems, distributed manufacturing environments, orders and shipment logistics, social media feeds, customer buying patterns, product lifecycle operations, and technology-driven data sources such as global positioning systems (GPS), radio frequency based identification (RFID) tracking, mobile devices, surveillance videos, and others. The use of big data in Supply Chain Management is gaining popularity globally both to drive performance improvements and to benefit from new insights [7].

Recent studies in the field of big data analytics have come up with tools and techniques to make data-driven supply chain decisions. Analyzing and interpreting results in real time can assist enterprises in making better and faster decisions to satisfy customer requirements. It will also help organizations to improve their supply chain design and management by reducing costs and mitigating risks.

Recently, various research studies have indicated the benefits of using big data methods in logistics and supply chain management. *Tan et al.*[8] proposed a big data analytics infrastructure based on deduction graph theory to enhance supply chain innovation capabilities. Cakici *et al.*[9] used RFID data for redesigning an optimal inventory policy. *Mishra and Singh*[10] proposed a big data analytics approach for waste minimization in food supply chains. *Zhong et al.*[11] stated how big data information could be used in effective logistics planning, production planning, and scheduling. *Shukla and Kiridena*[12] introduced a fuzzy rough sets-based multi-agent model for configuring supply chains in dynamic environments. *Dutta and Bose*[13] presented the challenges of managing a big data project for a cement supply and logistics network. *Singh et al.*[14] proposed a cloud computing framework for reducing the carbon footprint of a supply chain. *Waller and Fawcett*[15][16] argued that use of data science, predictive analytics, and big data could help logistics managers to meet internal needs and adjust to changes in the supply chain environment.

Along with these studies, there are many areas within supply chain management that could benefit from big data methods and technologies, including mitigation of bullwhip effect, multi-criteria decision making [17], sustainable supply chain management [18][19], sensor data-based predictive maintenance in manufacturing, efficient logistics [20], forecasting and demand management [21], and planning and scheduling [22]. To improve operational efficiency, integrated production and distribution processes across different supply chain components, big data information and technologies need to be reassessed. Manufacturers, logistics, suppliers, and retailers should develop a holistic approach to add value to their customers and services[7].

This study intends to investigate an area that was not well focused in previous studies. The aim is to apply Advance Analytics techniques such as Predictive Analysis using Pattern Reconigtion to investigate more than statistical analysis and forecasting demands.

3 Dataset and Classification Approach

The dataset used in this paper was obtained from a food company historical data. The data is dated from the period of 2018-01 to 2019-08. Since it is intended to predict the occurrence of DIFOT at the time of registering a new sales order, the problem becomes difficult to solve because the features used are all obtained *a priori*, that is, at the time of a sales order is created, and at this moment, some information is still not defined, such as, the carrier that will deliver, or if it is possible to optimize production to meet the demand, or even if it will be possible to deliver the entire quantity requested. One of the aims of this paper is to investigate how each feature, such as these mentioned, can impact in different ways the occurrence of DIFOT.

Some data preprocessing were applied to the dataset, some of them are, data types were adjusted, null values was filled, categorical feature were balanced, more specifically, features with low distribution were combined. Date and time features have given rise to new features such as day, month, week and weekday. And in the end of the preprocessing phase, the dataset contained a total of 54 features. Among all the features, some are "Number of SKU per Sales Order", day, week and month of order creation date; day, week and month of preparation date for delivery; distance between the distribution center and customer. There are also features that classify the quality of roads in the location of the distribution center and client. There are also weather characteristics for the preparation date and the possible delivery date. The categorical features, which are: Sales Type, Sales Document Type, Sales Organization, Sales Channel, Customer Segmentation Code and Distribution Channel, all of them were transformed using a simple encoding strategy that assigns an integer value for each distinct feature category, that is, if a categorical feature has only three states, the encoding process will generate the values 0, 1 and 2. As a final step for feature preparation, all features were subjected to a scaling standardization algorithm.

The proposed technique to predict DIFOT ocurrence for a Sales Order, is the application of machine learning approach using the algorithms for classification problems. A classification problem with only two classes is known as a binary classification problem. An example of a binary classification problem is the medical diagnosis of a certain disease. In this example, the induced classifier uses clinical information from a patient to determine if he/she has a particular disease. The classes represent the presence or absence of the disease. In particular, binary classification problems where the class value denotes the presence or absence of some property are known as concept learning[23]. Many real problems, however, involve the discrimination of more than two categories or classes. Examples of such problems are the classification of handwritten digits and the distinction of multiple types of tumor[24].

Classification using ML techniques consists of inducing a function $f(x)$ from a dataset composed of pairs (x_i, y_i), $where\ y_j \in \{1, ..., k\}$

A multiclass classification problem is intrinsically more complex than a binary problem, since the generated classifier must be able to separate the data into a higher number of categories, which increases the chances of classification errors. As a result, its complexity increases for more classes.The most common approach for the generalization of binary classification techniques to

solve multiclass problems is to decompose the problem into several binary subproblems. The learning algorithm induces a classifier for each of one these subproblems. The outputs of these classifiers are then combined to obtain the multiclass prediction[24].

Table 1: Difot - Classes Decomposition

Class	On Time	In Full	DIFOT
Class 1 - DIFOT	1	1	1
Class 2 - On Time	1	0	0
Class 3 - In Full	0	1	0
Class 4 - NA	0	0	0

Analyzing the table 1, it is possible to suggest some strategies to solve the problem. The first strategy is to work with a binary classifier and disregard the *On Time* and *In Full* column and let the machine learning algorithm find any boundary that can separate positive and negative cases for the occurrence of DIFOT. The second strategy is to decompose the problem disregarding DIFOT column, transforming the problem into multiple classes created from the binary combination of *On Time* and *In Full* values, ending with four classes. Finally, the last possible strategy is to create two binary classifiers, one for *On Time* and one for *In Full* state, combine the results of the two classifiers and finally get the result of DIFOT occurrence.

4 Methodology

The problem was analyzed following the three strategies previously suggested, that is, for the first strategy the data were analyzed under the standpoint of a binary classifier with DIFOT being the objective variable that results ends with DIFOT true or false. In the second standpoint, a multiclass classifier was created for the four possible combinations that ends with DIFOT being the result of one of the four possible classes to which an example belongs. In the third standpoint, two classifiers were created, one for *On Time* and another one for *In Full* and, later, their results were combined to generate the final classification result for DIFOT.

4.1 Random Forest

The random forest is a model made up of decision trees ensembles. Rather than just simply averaging the prediction of trees (which we could call a forest), this model uses two key concepts that gives it the name random. The first concept, a random sampling of training data points is used when building trees. The second concept is that Random subsets of features are considered when splitting nodes. Prediction is made by aggregating (majority vote for classification or averaging for regression) the predictions of the ensemble. Generally, Random forest shows better performance over single tree classifier such as ID3, CART, C4.5, C5.0. It yields generalization error rate that compares favorably to Adaboost, yet is more robust to noise[25].

Random forests are built on decision trees, and decision trees are sensitive to class imbalance, then RF can suffer from the curse of learning from an extremely imbalanced training dataset. Each tree is built on a bag, and each bag is a uniform random sample from the data (with replacement). Therefore each tree will be biased in the same direction and magnitude (on average) by class imbalance. it will tend to focus more on the prediction accuracy of the majority class, which often results in poor accuracy for the minority class. In order to overcome this weekness, we work with the approach of balanced random forest (BRF)[26].

4.2 Balanced Random Forest

The most common way of dealing with imbalanced data is introducing appropriately weighted costs for specific classes or sampling the available training set[27]. Balanced Random Forest is a modification of RF, where for each tree two bootstrapped sets of the same size, equal to the size of the minority class, are constructed: one for the minority class, the other for the majority class. Jointly, these two sets constitute the training set [26, 27]. The approach of equalizing the influences of classes is not performed externally to classification algorithm by evaluating weights, but is integrated in the very process.

The Balanced Random Forest (BRF) algorithm is as follows[26]:

1. For each iteration in random forest, draw a bootstrap sample from the minority class. Randomly draw the same number of cases, with replacement, from the majority class.
2. Induce a classification tree from the data to maximum size, without pruning. The tree is induced with the CART algorithm, with the following modification: At each node, instead of searching through all variables for the optimal split, only search through a set of mtry randomly selected variables.
3. Repeat the two steps above for the number of times desired. Aggregate the predictions of the ensemble and make the final prediction.

4.3 Feature Selection

Feature Selection is a basic concept in machine learning that has a considerable impact on the performance of the model. Feature selection is important for classification because this process removes irrelevant features so that it can improve model performance, make the model easier to understand, and reduce running [28].

Features Selection is divided into three types in general, Filter methods, Wrapper methods, and Embedded methods[29].

- Filter method is commonly used in preprocessing data. This method applies a statistical measure to

assign a scoring to each feature, The features are ranked by the score and then selected to be kept or removed from the datase. The ranking method filters out irrelevant features before starting the classification process. Advantages of this method are simplicity, good results and relevant features, and independent of any machine learning algorithm. Some examples of some filter methods include the Chi squared test, information gain.

Figure 2: Filter Method

- In Wrapper method different set of features are selected and compared to other combinations. A predictive model is used to test and compare different sets of features. Each model is assigned with a score, which is the accuracy. The search process for the best set of features may be methodical such as a best-first, stochastic such as a random hill-climbing, or it can use heuristics, such as forward and backward to add or remove features[29].

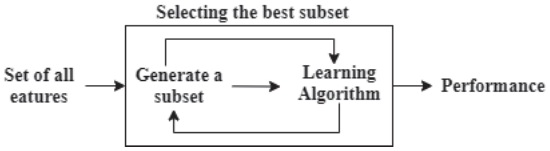

Figure 3: Wrapper Method

- Embedded methods find out which features can be more effective to the model final accuracy. This process occurs while the model is being trained. The most common embedded feature selection method are regularization methods. Also called penalization methods, they introduce addition constraints during the algorithm optimization which bias the model toward lower complexity. Examples of regularization are LASSO, Elastic Net and Ridge Regression. Another example of Embedded Methods is the process of Select from model, which, from a previous model, the most relevant features are selected and then, a new model is generated using the most important features[29, 28].

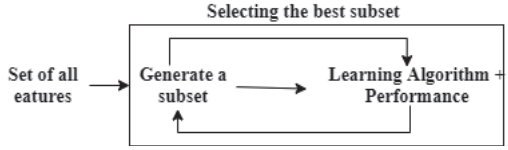

Figure 4: Embedded Method

In this paper we compared two different aproaches for feature selection, a Filter method *Mutual Information gain*, and the Embedded method *Tree Based Select From Model*.

4.3.1 Statistical & Ranking Filter Methods

Mutual information (MI) is a measure of the mutual dependence of two variables. It measures the information gain about a random variable through observing the other random variable. Mutual information is calculated between two variables and measures the reduction in uncertainty for one variable given a known value of the other variable[30]. This measure can express how much we can know about one variable by understanding the behavior of another variable. In machine learning, mutual information can measure how much information we gain or lose by adding or removing a variable during the prediction a an objective variable Y.

If X and Y are independent, their MI is Zero. If X is deterministic of Y, then MI is the entropy of X, which is a notion in information theory that measures or quantifies the amount of information within a variable[29].

$$I(X;Y) = \Sigma_{xy} P_{XY}(x,y) \log \frac{P_{XY}(x,y)}{P_X(x)P_Y(y)}$$

Where $P(X,Y)$ is the joint probability mass function of X and Y, and P_x and P_y are the marginal probability mass functions of X and Y respectively.

4.3.2 Tree Based Select From Model

Feature selection using Random forest comes under the category of Embedded methods. Embedded methods combine the qualities of filter and wrapper methods. They are implemented by algorithms that have their own built-in feature selection methods. Some of the benefits of embedded methods are highly accuracy, better generalization, interpretability. The tree-based strategies used by random forests naturally ranks by how well they improve the purity of the node. This mean decrease in impurity over all trees (called gini impurity). Nodes with the greatest decrease in impurity happen at the start of the trees, while notes with the least decrease in impurity occur at the end of trees. Thus, by pruning trees below a particular node, we can create a subset of the most important features[31].

4.4 Performance Metrics

When dealing with extremely imbalanced data, the overall classification accuracy is often not an appropriate metric to measure the performance. The accuracy can still be very high when the classifier predicts every case as the majority class. In this paper it was used metrics such as *Specificity'*, *G-mean*[32, 33], *Precision*, *Recall*, *F1-score* and *Index of Balanced Acuracy* [34] [35]. These metrics have been widely used for comparison. All the metrics are functions of the confusion matrix as shown in Table2. The rows of the matrix are actual classes, and the columns are the predicted classes. Based on Table2, the performance metrics are defined as:

Precision measures the precision of a predicted result

Table 2: Confusion Matrix

Class	Pred.Pos. Class	Pred.Neg. Class
Actual Pos.Class	TP	FN
Actual Neg.Class	FP	TN

as positive

$$\text{Precision: } P = \frac{TP}{TP + FP}.$$

Recall measures how many of predicted result are actual positive. positivo.

$$\text{Recall: } R = \frac{TP}{TP + FN}.$$

Specificity measures the proportion of negatives that are correctly identified.

$$\text{Specificity: } SP = \frac{TN}{TN + FP}.$$

F1-score is the harmonic mean between precision and sensitivity.

$$\text{F1-score: } F = 2 \cdot \frac{(Precision \cdot Recall)}{(Precision + Recall)}$$

G-mean is the root of the product of class-wise sensitivity. This measure tries to maximize the accuracy on each of the classes while keeping these accuracies balanced.

$$\text{G-mean: } G = (Sensitivity + Specificity)^1/2.$$

IBA balance any scoring function using the index balanced accuracy.

$$IBA\alpha(M) = (1 + \alpha \cdot Dom) \cdot M$$

The *Index of Balance Accuracy (IBA)* metric was proposed by [36] and is not wiedely know how the others, so in order to explain this metric, where *Dom* called dominance, is defined as $Dom = TPrate - TNrate$ within the range $[-1, +1]$, and it is weighted by $\alpha \geq 0$ to reduce its influence on the result of the particular metric M [35].

The dominance is here used to estimate the relationship between $TPrate$ and $TNrate$. The closer the dominance is to 0, the more balanced both individual class ccuracies are. The weighting factor $(1 + \alpha \times Dom)$ in IBA equation is within the range $[1 - \alpha, 1 + \alpha]$. Note that if $\alpha = 0$ or $TPrate = TPrate$, the IBA_α turns into the measure M. In practice, one should select a value of α depending on the metric used. In the present paper we will utilize $M = Gmean^2$ and $\alpha = 0.1$ [35].

In all different classification approaches, the classes were extremmelly inbalanced, and then The Balanced Random Forest was used as the Machine Learning algorithm which also deals very well with imbalanced data. All the algorithms used to perform each step over the data are from the scikit-learn package [37, 38].

4.5 Experiment for Binary Classifier for DIFOT

In the first experiment, a binary classifier for DIFOT was created. The data were splited into 80% for training and 20% for testing. The binary classification algorithm used was *Balanced Random Forest* present in the scikit-learn library version 0.23.1 for the Python 3.7.4 programming language [37, 38]. The balanced class mode uses the values of y to automatically adjust the weights inversely proportional to the frequency of the input classes in the way of $nsamples \div (nsamples \times np.bincount(y))$. Three test runs were performed using the same random seed. In each round, a different strategy of feature selection was used in order to identify which set of features is most relevant for each experiment round. In the first round, all 54 features were used. In the second round a selection was made based on *Tree Based Select From Model* strategy, and in the third round a statistical approach of *Ranking Filter K Best Features* that ranks each feature using an entropy measure called mutual information and then, K best features are selected.

Analyzing ROC curve graph (Figure 5), It is possible to verify a good performance of *Class* 0 and 1 (DIFOT True and False). It is not possible to accurately identify which class had the best performance, when analyzing, for instance, at the approximate point of $FPR \leq 0.1$, the negative class achieved better performance, however, when the value of $FPR = 0.2$, the positive class exceeds the negative in performance.

Analyzing the graph of the Precision-Recall curve (Figure 5), we can see that for the positive class it maintains high precision and high recall almost throughout the entire range of thresholds. For the negative class precision is starting to fall as soon as it starts recalling 0.3 of true positives and by the time it hits 0.8, it decreases to around 0.6. It is possible to identify that the high imbalance of *Class* 0 was one of the reasons for the instability of its accuracy. Although class balancing was introduced in the model used, an observation that can be stressed is that the features used are not able to represent *Class* 0 accurately when compared to *Class* 1.

Analyzing the Table3, it is possible to identify that the first round model with 54 features was the one that obtained the best result for $G - mean$ and IBA metrics. The dataset with 54 features used by this model is presented in Figure6. In the first graph, the 2D projection was made from 2 components created of a Principal Component Analysis (PCA) over the the 54 features. In the second graph of Figure6, the 2 components previously created were submitted to the t-SNE algorithm, which is specialized in multidimensional data projection, which generated 2 new components. The *t-Distributed Stochastic Neighbor Embedding (t-SNE)* is a technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets[39].

In the third graph of Figure6, from the Dataset and its 54 features, a PCA generated 15 components, and then, these 15 components were submitted to *t-SNE* algorithm that generated 2 new components.

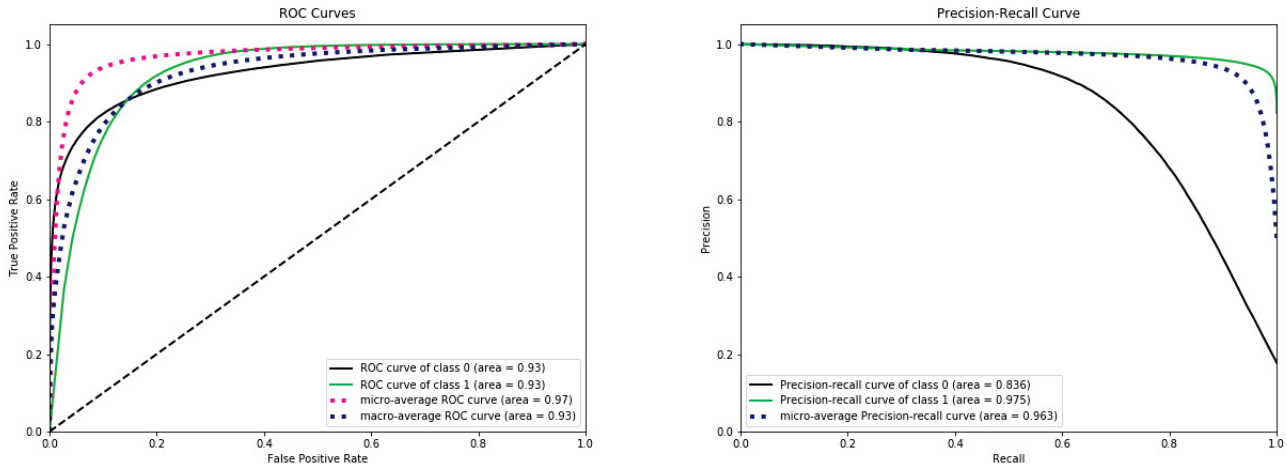

Figure 5: ROC and Precision/Recall curve of DIFOT Classification.

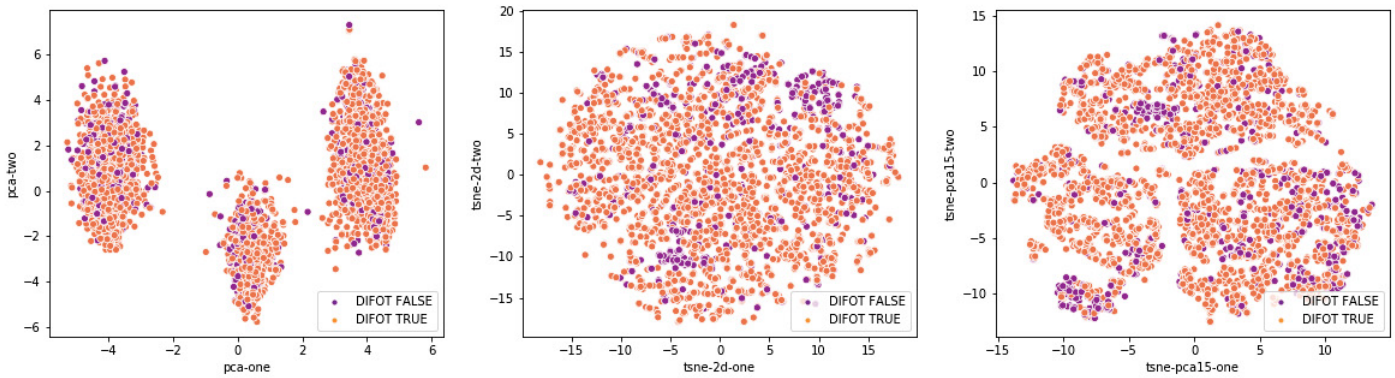

Figure 6: Data Distribution using PCA and t-SNE for DIFOT Classification.

Table 3: Results for Binary classifier for DIFOT

54	First run - All Features						
Features	***Prec.***	***Rec.***	***Spec.***	***F1***	***Gmean***	***IBA***	***Sup.***
Difot F	0.86	0.68	0.98	0.76	0.81	0.64	42550
Difot T	0.93	0.98	0.68	0.95	0.81	0.68	197062
Avg/Total	*0.92*	*0.92*	*0.73*	*0.92*	***0.81***	***0.67***	*239612*

11	Second run - Feature selected from model						
Features	***Prec.***	***Rec.***	***Spec.***	***F1***	***Gmean***	***IBA***	***Sup.***
Difot F	0.86	0.63	0.98	0.73	0.79	0.60	42550
Difot T	0.92	0.98	0.63	0.95	0.79	0.64	197062
Avg/Total	*0.91*	*0.92*	*0.69*	*0.91*	*0.79*	*0.63*	*239612*

27	Third run - K best features						
Features	***Prec.***	***Rec.***	***Spec.***	***F1***	***Gmean***	***IBA***	***Sup.***
Difot F	0.83	0.62	0.97	0.71	0.78	0.58	42550
Difot T	0.92	0.97	0.62	0.95	0.78	0.62	197062
Avg/Total	*0.91*	*0.91*	*0.68*	*0.90*	*0.78*	*0.61*	*239612*

In the first graph of the Figure6 it is possible to identify that *Class* 0 and *Class* 1 are distributed in 3 groups, and the classes are superimposed, with its data with hard separation frontier. The second graph, created from $t-SNE$ analysis of 2-component PCA, shows a small agglomeration of *Class* 0 in some data regions, but it is still possible to see too much overlap of *Class* 1. In the third graph, where 2 components$t-SNE$ were created from 15-component PCA, again the data is shown to be basically distributed in 3 groups, and in these groups the classes are shown overlapping. *Class* 0 shows a small proximity in the 3 groups, but it is still possible to verify that the data are difficult to separate when interpreted under the view of the DIFOT objective.

4.6 Experiment for Multiclass DIFOT Decomposition

In the second experiment, the DIFOT class was decomposed into four possible intermediate classes, as shown in Table1, which are, DIFOT, *On Time*, *In Full* and NA, all binary classes. The dataset were splited into 80% for training and 20% for testing. The binary classification algorithm used was *Balanced Random Forest*. Three rounds for testing were performed using the same random seed for all models. In each round, the same strategy for feature selection was used as discussed in the first experiment. The results presented are as follows.

Analysing the results from Table4, the first round with 54 features and the second round with just 18 features, they presented practically the same results for $G-mean$ and IBA mean. Analyzing the results of the $G-mean$ metric individually, the first round was a little better. That being said, it is important to note that the results of the second round were obtained using only 33% of the features. Thus, for data analysis, the model chosen was the one from the second round, with only 18 features. The second round model had an average $G-mean$ of 0.78 and an average IBA of 0.63. Below are the graphs of the ROC curve and Precision-Recall. Difot is identified as *Class* 0, *On Time* as *Class* 1, *In Full* as *Class* 2 and NA as *Class* 3.

Analyzing ROC curve in Figure7, it is possible to verify a good performance of *Class* 2 and *Class* 1 (*In full* and *On time* respectively), followed by *Class* 0 and *Class* 3 (Difot and NA respectively). It is possible to observe a slower growth of *Class* 0 when compared to *Class* 2 and *Class* 1 for average $FPR \leq 0.3$. Even (*Class* 3), which denotes events that are neither *On Time* nor *In Full*, had an unsatisfactory performance compared to the other Classes. Still, it is possible to observe that from $FPR > 0.3$, *Class* 0, *Class* 1 and *Class* 2 stabilize and resemble each other.

Analyzing the Precision-Recall curve in Figure7, we can see that for the *Class* 0 it maintains high precision and high recall almost throughout the entire range of thresholds. For the *Class* 1 and *Class* 2, both have similar behavior, but yet *Class* 1 performs slightly better compared to *Class* 2. Both start to decrease as the recall increases. Both are not maintained with good accuracy, reaching around an accuracy of 0.85 when the recall reaches 0.8. For *Class* 3, the precision is completely unprecise as soon as the recall starts to increase.

Folowing the same approach of the first experiment to plot the data, now the best model chosen is the from the second round with 18 features which good results for $G-mean$ and IBA metrics. The dataset with 18 features used by this model is presented in Figure8. In the first graph, the 2D projection was made from 2 components created of a Principal Component Analysis (PCA) over the the 18 features. In the second graph of Figure8, the 2 components previously created were submitted to the $t-SNE$ algorithm that generated 2 new components.

In the first graph of Figure8 it is possible to verify that the data are grouped in just one single large block without any significant subgroup, however it is possible

Table 4: Results for Multiclass DIFOT Decomposition

54	First run - All Features						
Features	***Prec.***	***Rec.***	***Spec.***	***F1***	***Gmean***	***IBA***	***Sup.***
Difot	0.93	0.98	0.64	0.95	0.80	0.65	197062
On Time	0.81	0.71	0.99	0.76	0.84	0.69	18886
In Full	0.79	0.65	1.00	0.71	0.80	0.62	6673
NA	0.69	0.36	0.99	0.47	0.60	0.33	16991
Avg/Total	*0.90*	*0.91*	*0.71*	*0.90*	*0.78*	*0.63*	*239612*

18	Second run - Feature selected from model						
Features	***Prec.***	***Rec.***	***Spec.***	***F1***	***Gmean***	***IBA***	***Sup.***
Difot	0.93	0.99	0.63	0.96	0.79	0.65	197062
On Time	0.82	0.72	0.99	0.77	0.84	0.69	18886
In Full	0.82	0.64	1.00	0.72	0.80	0.62	6673
NA	0.73	0.34	0.99	0.46	0.58	0.31	16991
Avg/Total	*0.90*	*0.91*	*0.70*	*0.90*	***0.78***	***0.63***	*239612*

27	Third run - K best features						
Features	***Prec.***	***Rec.***	***Spec.***	***F1***	***Gmean***	***IBA***	***Sup.***
Difot	0.92	0.97	0.59	0.94	0.76	0.59	197062
On Time	0.74	0.67	0.98	0.71	0.81	0.64	18886
In Full	0.67	0.49	0.99	0.57	0.70	0.46	6673
NA	0.54	0.31	0.98	0.40	0.55	0.29	16991
Avg/Total	*0.87*	*0.88*	*0.66*	*0.87*	*0.75*	*0.57*	*239612*

to observe that in the axis $pca - one < 1$ there is a visibly greater concentration of *Class* 0, while starting from the axis $pca - one > 1$ the other classes 1, 2 and 3 are spread out. It is also possible to observe that the values of *Class* 2 are slightly more grouped from $pca - one = 4$ and $pca - two \leq 0$.

In the second graph of Figure8, it is already possible to identify a larger grouping of *Class* 1 data from the axis $tsne - 2d - one \geq 10$ and the $axistsne - 2d - two \leq 5$. It is also possible to verify a higher occurrence of *Class* 2 on the axis $tsne - 2d - one \leq 5$ and axis $tsne - 2d - two \leq -5$. In chart 3 of Figure8, the data for *Class* 1 are more grouped on the axis $tsne - pca15 - one > 5$. *Class* 2 data, on the other hand, are more dispersed from the axis $tsne - pca - 15 - one \geq -5$ and the axis $tsne - pca15 - two \geq 0$. In all graphs, *Class* 3 did not show any identifiable distribution pattern, which may suggest that there may be some outlier.

4.7 Experiment for DIFOT intermediary composition

In the third experiment, the strategy used was to combine binary classifiers. When Difot occurs which means that *On Time* and *In Full* also occurred, this means that, for any other combination between *On time* and *In Full*, DIFOT does not occur. That said, two classifiers were created, one for *On time* and the other for *In Full*. The data were splited into 80% for training and 20% for testing, while the classification algorithm was the Balanced Random Forest. For each objective, *On Time* and *In Full*, three test rounds were performed using the same feature selection strategies as the previous experiments.

4.7.1 Classifier for "On Time" state

The resultos for *On Time* state is shown in Table 4 and Figure 9 and 10.

Analyzing ROC curve in Figure 9 it is possible to verify the performance of *Class* 1 and *Class* 0 (*On time* true and false respectively). It is not possible to accurately identify which class performed best. When analyzing, for example, the approximate point of $FPR \geq 0.2$, the negative class achieved better performance, however, when the value of $FPR > 0.2$, the positive class exceeds the negative in performance.

Analyzing the Precision-Recall in 9 curve graph, it is possible to see that for the positive class it maintains high precision and high recall almost throughout the entire range of thresholds. For the negative class precision is starting to fall as soon as recalling true positives and by the time it hits 0.8, decreases to around 0.3. It is possible to identify that the high imbalance of *Class* 0 may have been one of the reasons for the instability of its accuracy. Although class balancing was introduced in the model, an observation that can be made is that the features used cannot accurately explains *Class* 0 compared to *Class* 1.

Folowing the same approach of the previous experiments to plot the data, now the best model chosen is the third one with 27 features which have obtained good results for $G - mean$ and IBA metrics. The dataset with 27 features used by this model is presented in Figure10. In the first graph, the 2D projection was made from 2 components created of a Principal Component Analysis (PCA) over the the 27 features. In the second graph

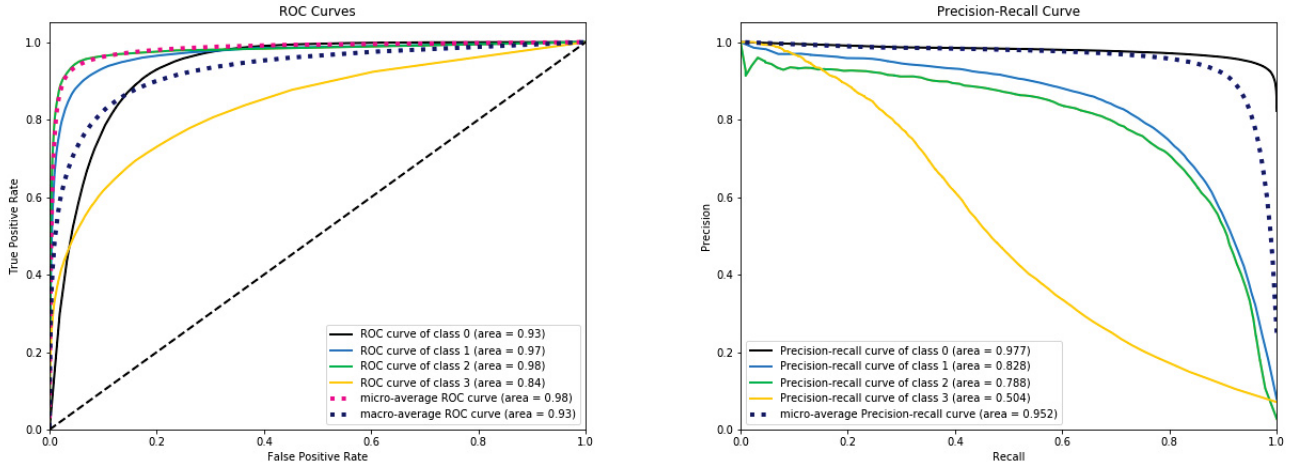

Figure 7: ROC and Precision/Recall curve of Multiclass Classification.

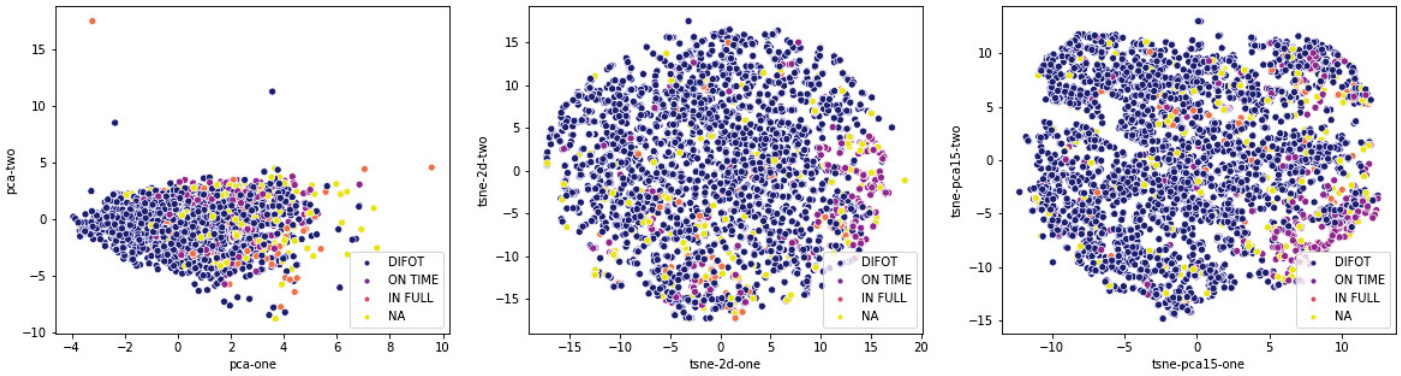

Figure 8: Data Distribution using PCA and t-SNE for Multiclass Classification.

Table 5: Results for On Time state

54	**First run - All Features**						
Features	***Prec.***	***Rec.***	***Spec.***	***F1***	***Gmean***	***IBA***	***Sup.***
OnTime F	0.83	0.50	0.99	0.63	0.71	0.47	23664
OnTime T	0.95	0.99	0.50	0.97	0.71	0.52	215948
Avg/Total	*0.94*	*0.94*	*0.55*	*0.93*	*0.71*	*0.52*	*239612*

15	**Second run - Feature selected from model**						
Features	***Prec.***	***Rec.***	***Spec.***	***F1***	***Gmean***	***IBA***	***Sup.***
OnTime F	0.85	0.47	0.99	0.61	0.68	0.44	23664
OnTime T	0.94	0.99	0.47	0.97	0.68	0.49	215948
Avg/Total	*0.94*	*0.94*	*0.52*	*0.93*	*0.68*	*0.49*	*239612*

27	**Third run - K best features**						
Features	***Prec.***	***Rec.***	***Spec.***	***F1***	***Gmean***	***IBA***	***Sup.***
OnTime F	0.69	0.55	0.97	0.61	0.73	0.51	23664
OnTime T	0.95	0.97	0.55	0.96	0.73	0.55	215948
Avg/Total	*0.93*	*0.93*	*0.59*	*0.93*	***0.73***	***0.55***	*239612*

of Figure10, the 2 components previously created were submitted to the t-SNE algorithm that generated 2 new components.

In the first graph of Figure10, it is possible to identify

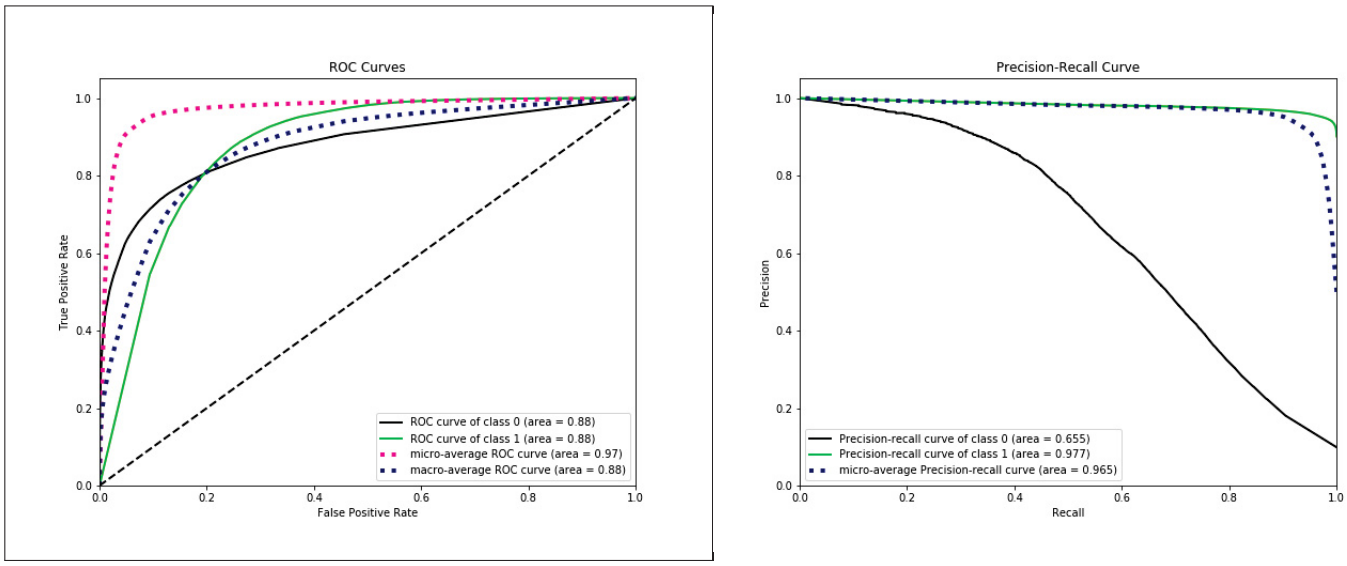

Figure 9: ROC and Precision/Recall curve of *On Time* Classification.

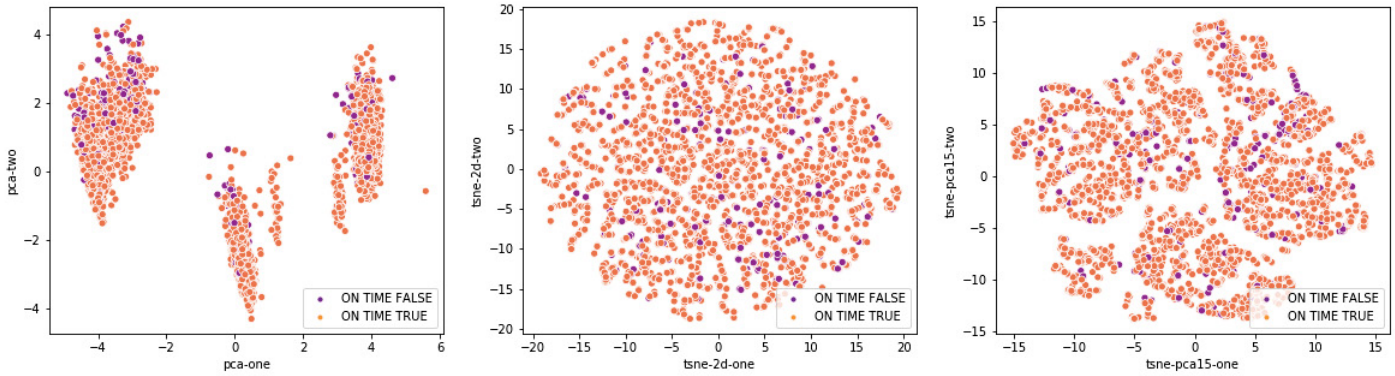

Figure 10: Data Distribution using PCA and t-SNE for *On Time* Classification.

that the data is extremely imbalanced. It is possible to observe the greater recurrence of *Class* 1(*On time* true). Also, it is possible to identify 3 minor groups. There is no clear overlap of the data, but it is also not possible to identify that each class has its own grouping with its peers. The second graph, made from $t - SNE$ created from PCA of 2 components, shows a large dispersion of the data. In the third graph, created from PCA of 15 components that were summarized in $t - SNE$ of 2-component, it is possible to verify in some plane, albeit timidly, *Class* 0 and *Class* 1 are closer to their counterparts, it is also possible to observe small clusters that shows similarity and within these small clusters there is a recurrence of both classes, however, *Class* 0 is closer to its similars, information that is difficult to be observed when analyzing the two previous graphs.

4.7.2 Classifier for "In Full" state

The results for *In Full* state is shown in Table 5 and Figure 11 and 12.

Analyzing ROC curve graph in Figure11 it is possible to verify the performance of *Class* 1 and *Class* 0 (*In Full* true and false). *Class* 0 performs best when $FPR < 0.2$, and *Class* 1 performs best when $FPR > 0.2$ and maintains the best performance throughout the entire range of FPR thresholds.

Analyzing the Precision-Recall curve in Fiture12, we can see that for the positive class it maintains a high precision and high recall almost throughout the entire range of thresholds. For the negative class precision is starting to fall when recalling 0.3 and when hit recalls 0.8, it decreases to around 0.55. It is possible to identify again that there is a high class imbalance for *Class* 0 and this may have impacted the model's performance.

Folowing the same approach of the previous experiments to plot the data, the best model chosen is from the first round with all 54 features which have obtained the best results for $G - mean$ and IBA metrics. The dataset with 54 features used by this model is presented in Figure12. In the first graph, the 2D projection was made from 2 components created of a Principal Component Analysis (PCA) over the the 54 features. In the second graph of Figure10, the 2 components previously created

Table 6: Results for In Full state

54	**First run - All Features**						
Features	***Prec.***	***Rec.***	***Spec.***	***F1***	***Gmean***	***IBA***	***Sup.***
InFull F	0.84	0.63	0.98	0.72	0.78	0.59	35877
InFull T	0.94	0.98	0.63	0.96	0.78	0.63	203735
Avg/Total	*0.92*	*0.93*	*0.68*	*0.92*	***0.78***	***0.63***	*239612*

11	**Second run - Feature selected from model**						
Features	***Prec.***	***Rec.***	***Spec.***	***F1***	***Gmean***	***IBA***	***Sup.***
InFull F	0.84	0.58	0.98	0.69	0.75	0.55	35877
InFull T	0.93	0.98	0.58	0.95	0.75	0.59	203735
Avg/Total	*0.92*	*0.92*	*0.64*	*0.91*	*0.75*	*0.58*	*239612*

27	**Third run - K best featuress**						
Features	***Prec.***	***Rec.***	***Spec.***	***F1***	***Gmean***	***IBA***	***Sup.***
InFull F	0.84	0.57	0.98	0.68	0.75	0.54	35877
InFull T	0.93	0.98	0.57	0.95	0.75	0.59	203735
Avg/Total	*0.92*	*0.92*	*0.64*	*0.91*	*0.75*	*0.58*	*239612*

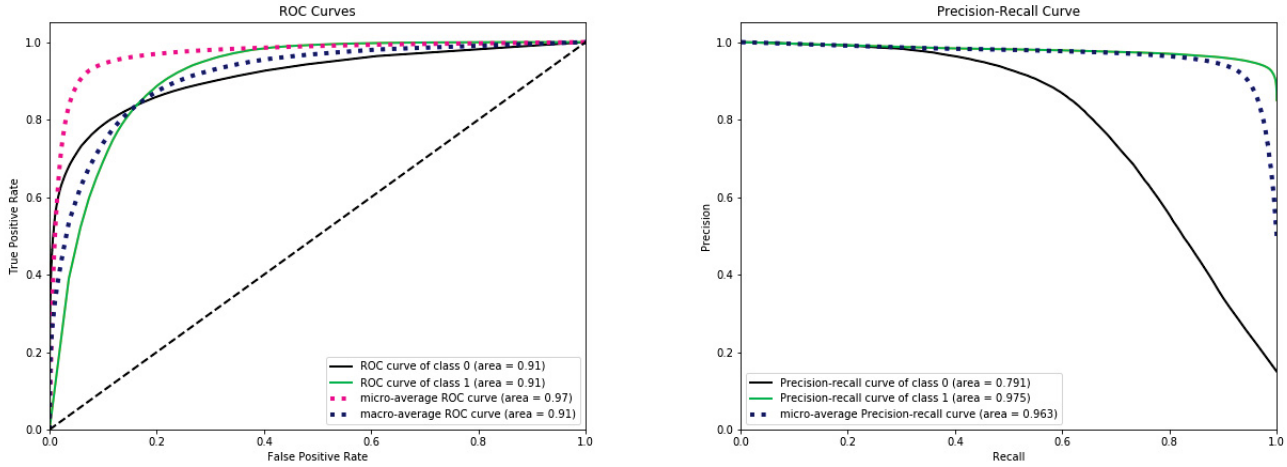

Figure 11: ROC and Precision/Recall curve of *In Full* Classification.

were submitted to the t-SNE algorithm that generated 2 new components.

In the first graph of Figure12 it is possible to verify that *Class* 1 is much more recurrent than *Class* 0. The data is divided in 3 minor groups that are not exclusive to each class. There is, apparently, data overlapping, it is also possible to identify that each class is not grouped only with its peers. The second graph, made from $t - SNE$ analysis created from a PCA of 2-component, shows data dispersion, it is possible to verify this fact when we analyze the behavior of *Class* 0, but items from *Class* 0 are close together in small clusters. In the third graph, created from a PCA of 15 components that were summarized in $t - SNE$ of 2 component, there is a clear division into 4 clusters, 3 larger ones with mixed classes between 0 and 1, and a smaller cluster with a predominant occurrence of the *Class* 0. Again, the classes are dispersed in all clusters, but as we are analyzing through a 2D projection, maybe in some dimension these data are more separated. It is also possible to observe that *Class* 0 normally tends to group into small clusters within the larger clusters dominated by *Class* 1.

4.7.3 On Time and In Full combinaton results

Combining the two classifiers (*On time* and *In Full*) and submitting the same testing data, it was possible to obtain all classes of the second experiment (Multiclass) as shown in Table1. That said, after combining the results of the two classifiers, the result is as follow.

The results presented in 7 is the best results of all previous experiments shown in Experiment A and B, reaching a better $G - mean$ and IBA values.

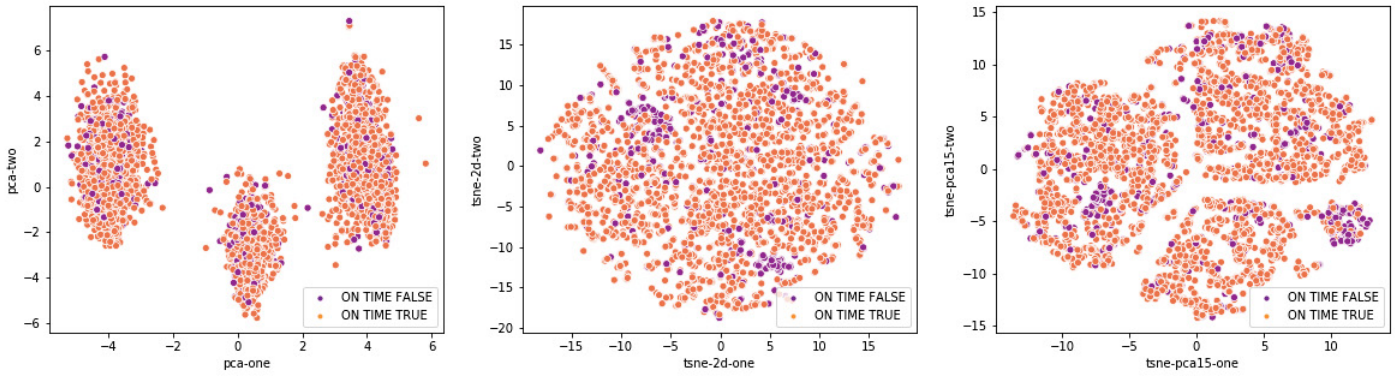

Figure 12: Data Distribution using PCA and t-SNE for *In Full* Classification.

Table 7: Results for DIFOT (Combined by On Time and In Full Classifiers)

	First run - All Features						
	Prec.	***Rec.***	***Spec.***	***F1***	***Gmean***	***IBA***	***Sup.***
Difot	0.94	0.96	0.70	0.95	0.82	0.69	197062
On Time	0.72	0.72	0.98	0.72	0.84	0.69	18886
In Full	0.44	0.72	0.97	0.55	0.84	0.69	6673
NA	0.71	0.33	0.99	0.45	0.57	0.30	16991
Avg/Total	*0.89*	*0.89*	*0.75*	*0.88*	***0.81***	***0.67***	*239612*

5 Conclusions

Analysing all the results presented, it is possible to verify that the Experiment 4.5 and Experiment 4.7 obtained practically the same results. It is interesting to note that two considerably different techniques have achieved similar results. Whereas the Experiment 4.6 got the worst result. It is quite possible that the data imbalance, which was accentuated in Multiclass4.6, may have contributed to the model low performance. From the two Experiments, 4.5 and 4.7, the Experiment 4.7 can still be considered better because it use just a fewer features to build the model, with only 27 features the model was able to represent the *On time* state.

When we analyze the problem and the data deeply, it is possible to infer that the strategy applied in Experiment 4.7 is a good approach since *On Time* is related to transport, distance, departure and arrival days, and *In Full* is related to stock, quantity, and production. That being said, the concepts are totally different and both can use different features. From this point of view, the *In Full* information can hinder the classification of *On Time* data and vice versa. The dataset analyzed did not have much information related to stock and production, relying only on the correlation of other features to identify a good frontier that separates *In Full* data.

Finally, it is possible to conclude that the DIFOT classification problem is viable, and in order that get a good classifier, it is necessary to focus on features inherent to each of the intermediate states composition, *On Time* and *In Full* states. The dataset must have features for *On Time* tails such as distance, expected arrival date, departure date and climatic information, all of these features can affect the product transport, impacting in *On Time* delivery. For the *In Full* state, the necessary features are linked to production, inventory, due date, promotion, and product losses. A dataset that has such features will have success classifying DIFOT.

References

[1] T. McLean. *On Time, In Full: Achieving Perfect Delivery with Lean Thinking in Purchasing, Supply Chain, and Production Planning*. Taylor & Francis, 2017.

[2] Mark Johnson and Graham Stevens. Integrating the supply chain... 25 years on. *International Journal of Physical Distribution & Logistics Management*, 46, 02 2016.

[3] Rob O'Byrne. Kpi key performance indicators in supply chain & logistics, 2020.

[4] Ganda Boonsothonsatit. Supply chain causal linkage-based strategic map design. *Journal of Advanced Management Science*, 5, 03 2016.

[5] M. Christopher. *Logistics & Supply Chain Management*. Financial Times Series. Financial Times Prentice Hall, 2011.

[6] Sunil Tiwari, H.M. Wee, and Yosef Daryanto. Big data analytics in supply chain management be-

tween 2010 and 2016: Insights to industries. *Computers & Industrial Engineering*, 115:319 – 330, 2018.

[7] Kannan Govindan, T. C. E. Cheng, Nishikant Mishra, and Nagesh Shukla. Big data analytics and application for logistics and supply chain management. *Transportation Research Part E: Logistics and Transportation Review*, 114, 05 2018.

[8] Kim Tan, Yuanzhu Zhan, Guojun Ji, Fei Ye, and Ching-Ter Chang. Harvesting big data to enhance supply chain innovation capabilities: An analytic infrastructure based on deduction graph. *International Journal of Production Economics*, 165, 01 2015.

[9] Ozden Cakici, Harry Groenevelt, and Abraham Seidmann. Using rfid for the management of pharmaceutical inventory-system optimization and shrinkage control. *Decision Support Systems*, 51:842–852, 11 2011.

[10] Nishikant Mishra and Akshit Singh. Use of twitter data for waste minimisation in beef supply chain. *Annals of Operations Research*, 270, 11 2018.

[11] Ray Zhong, George Huang, Shulin Lan, Q.Y. Dai, Chen Xu, and Ting Zhang. A big data approach for logistics trajectory discovery from rfid-enabled production data. *International Journal of Production Economics*, 165, 02 2015.

[12] Nagesh Shukla and Senevi Kiridena. A fuzzy rough sets-based multi-agent analytics framework for dynamic supply chain configuration. *International Journal of Production Research*, pages 1–13, 02 2016.

[13] Debprotim Dutta and Indranil Bose. Managing a big data project: The case of ramco cements limited. *International Journal of Production Economics*, 165, 01 2015.

[14] Akshit Singh, Nishikant Mishra, Syed Ali, and Nagesh Shukla. Cloud computing technology: Reducing carbon footprint in beef supply chain. *International Journal of Production Economics*, 164, 09 2014.

[15] Matthew Waller and Stanley Fawcett. Click here for a data scientist: Big data, predictive analytics, and theory development in the era of a maker movement supply chain. *Journal of Business Logistics*, 34, 12 2013.

[16] Matthew Waller and Stanley Fawcett. Data science, predictive analytics, and big data: A revolution that will transform supply chain design and management. *Journal of Business Logistics*, 34, 06 2013.

[17] Kannan Govindan, Roohollah Khodaverdi, and Ahmad Jafarian. A fuzzy multi criteria approach for measuring sustainability performance of a supplier based on triple bottom line approach. *Journal of Cleaner Production*, 47:345–354, 05 2013.

[18] Devika Kannan. Role of multiple stakeholders and the critical success factor theory for the sustainable supplier selection process. *International Journal of Production Economics*, 195, 04 2017.

[19] Kannan Govindan, Milosz Kadzinski, and Sivakumar Rajendran. Application of a novel prometheebased method for construction of a group compromise ranking to prioritization of green suppliers in food supply chain. *Omega*, 71, 11 2016.

[20] Kannan Govindan, Hamed Soleimani, and Devika Kannan. Reverse logistics and closed-loop supply chain: A comprehensive review to explore the future. *European Journal of Operational Research*, 240:603–626, 02 2015.

[21] Tobias Schoenherr and Cheri Speier[U+2010]Pero. Data science, predictive analytics, and big data in supply chain management: Current state and future potential. *Journal of Business Logistics*, 36, 02 2015.

[22] Felix Chan, Anuj Prakash, and Nishikant Mishra. Priority-based scheduling in flexible system using ais with flc approach. *International Journal of Production Research*, 51, 08 2013.

[23] T.M. Mitchell. *Machine Learning*. McGraw-Hill International Editions. McGraw-Hill, 1997.

[24] Ana Lorena, Andre Carvalho, and João Gama. A review on the combination of binary classifiers in multiclass problems. *Artificial Intelligence Review*, 30:19–37, 12 2008.

[25] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.

[26] Chao Chen and Leo Breiman. Using random forest to learn imbalanced data. *University of California, Berkeley*, 01 2004.

[27] Łukasz Kobyliński and Adam Przepiórkowski. Definition extraction with balanced random forests. volume 5221, pages 237–247, 01 2008.

[28] Mia Huljanah, Zuherman Rustam, Suarsih Utama, and Titin Siswantining. Feature selection using random forest classifier for predicting prostate cancer. *IOP Conference Series: Materials Science and Engineering*, 546:052031, jun 2019.

[29] Jason Brownlee. An introduction to feature selection, 2014.

[30] LF Kozachenko and Nikolai N Leonenko. Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii*, 23(2):9–16, 1987.

[31] Chris Albon. Feature selection using random forest, 2017.

[32] M. Kubat. Addressing the curse of imbalanced training sets: One-sided selection. *Fourteenth International Conference on Machine Learning*, 06 2000.

[33] Ricardo Barandela, Josep Sánchez, Vicente García, and E. Rangel. Strategies for learning in class imbalance problems. *Pattern Recognition*, 36:849–851, 03 2003.

[34] V. García, J.S. Sánchez, and R.A. Mollineda. On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. *Knowledge-Based Systems*, 25(1):13 – 21, 2012. Special Issue on New Trends in Data Mining.

[35] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017.

[36] Vicente García, Ramón Mollineda, and Josep Sánchez. Theoretical analysis of a performance measure for imbalanced data. pages 617–620, 08 2010.

[37] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[39] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.