The George Washington University

DNSC 6314: Machine Learning II

Professor Dr.Long He

Supplier Risk Classification by using the SIFOT

Group 10

Manuel Chavez
Anam Husain
Kylie Loudermilk
Oyuna Johnson
Cameron Stopak

May 9, 2024

Introduction

Mining Equipment Company (MEC) is a global productivity partner for mining and construction customers, and accelerates the transformation toward a sustainable society. With ground-breaking technology, MEC develops and provides innovative and safe equipment, such as drill rigs, rock excavation and construction equipment and tools for surface and underground applications. The company also offers world-class service and other aftermarket support as well as solutions for automation, digitalization and electrification, based in Europe, had revenues of more than $5.5 billion in 2023, and has around 18,000 employees supporting and collaborating with customers in around 150 countries.

MEC has 8 divisions divided into five areas: Surface, Underground, Parts & Services, Digital Solutions, Tools and Attachments. These divisions and areas of focus demonstrate MEC's comprehensive approach to serving the needs of the mining and construction industries, covering everything from equipment manufacturing to maintenance, digital innovation, and support services.

As a global company operating in the mining and construction industries, MEC sources products and components from suppliers worldwide. It has a global supply chain network that includes sourcing materials, components, and finished products from various countries. This allows them to access specialized expertise, cost-effective manufacturing capabilities, and diverse product offerings. MEC often works with a diverse supplier base to mitigate risks, improve flexibility, and enhance innovation. Also, they source products from different countries to leverage the strengths of suppliers in terms of quality, price, technology, and reliability.

Moreover, each MEC headquarter collaborates with Intercompany's manufacturer centers and as well with strategic partners and suppliers globally to develop new technologies, improve product performance, and address customer needs effectively. In addition, MEC focuses on building a resilient supply chain by diversifying sourcing locations, assessing geopolitical risks, ensuring continuity of supply, and implementing contingency plans to mitigate any disruptions.

I. <u>Business understanding</u>

Business Problem

In the modern global economy, businesses depend heavily on a complex web of suppliers to ensure efficient, competitive delivery of goods and services. This interdependence, while beneficial, also subjects organizations to substantial risks that can affect their operations, profitability, and reputation. Without a systematic approach to identifying and managing supplier risks, organizations encounter challenges like quality issues, financial instability, geopolitical issues, supplier bankruptcies, and unexpected events. Such challenges can result in production delays, shortages of essential components, and customer dissatisfaction, posing severe risks to the company's overall success.

By this approach, in the Procurement area, acquisitions of goods must be made, either with Intercompany manufacturing centers or third-party suppliers that help meet demand. Monitoring the arrival of the purchased goods is the responsibility of the Expediting team, which activates tracking of each purchase line, once the order has been issued and accepted. However, given the high load in terms of the number of purchase lines and considering contractual commitments and even more so the level of service committed to DIFOT (Delivery in Full on Time), they require adequate control with the SIFOT (Supply in Full On Time). , and finally this results in obtaining deliveries aligned with the commitment date. This is why the Procurement and Expediting team requires centralizing efforts on both specific expediting lines that impact revenue (considering rejection probabilities), as well as patterns that can help develop efforts for adequate contractual closure, mitigation of risks or sourcing with new suppliers or manufacturing centers.

II. Exploratory Analysis

Data Overview:

Datasets and Sources:

Basically we will work with all the datasets related to the procurement process , as follows:

- **Purchase Orders Dataset :** Record of purchase orders from the previous 4 years.
- **Vendor Performance / Master Dataset :** Records of performance, built by a mixture of External assessment (Dun & Bradstreet) and internal score by the Procurement team (Expediting & Purchasing).
- **Part Master Dataset:** Description of specific features according to each part number.
- **Contract Agreement Dataset :** Description of terms and conditions (fixed price, penalties and warranty), associated with the agreement with one of the customers.

Overview:

We obtained purchase data from the MEC in previous years. The data contains 56 columns and 45,241 rows providing detailed insights into order delivery timelines. Specifically, it indicates whether each order arrived on schedule and quantifies the number of days it took for orders to be fulfilled.

- The data relates to purchase orders (POs) and procurement activities within a company.
- Each row represents a specific item within a purchase order, identified by a unique PO number and line number.
- Key information includes order quantities, received quantities, unit of measure, order values, received values, open quantities, vendor details, delivery status, approval status, and dates (order date,promise date, due date, receipt date, etc.).
- The data also includes information about product groups, part numbers, divisions, departments, charts, and buyer IDs.
- It appears that the data covers multiple purchase orders, each containing several items, and tracks their progress from order placement to receipt and approval.

- There are indicators for on-time delivery, late delivery, vendor performance, and inspection requirements.

Overall, the data provides a comprehensive view of procurement activities, vendor relationships, inventory management, and order fulfillment processes. By analyzing this dataset, the company gains insights into its purchasing patterns, vendor performance, inventory management effectiveness, and adherence to delivery timelines. The structured nature of the data enables robust monitoring, trend analysis, and decision-making to optimize procurement processes, enhance supply chain efficiency, and ensure timely and cost-effective acquisition of goods and services, and reduce the supplier's risk.

Data preprocessing was a crucial initial step. It involved the following process:

- Extracting unique values from the Order Type column in the dataset
- Filtering the target variables from the unique values
- Dropping irrelevant or redundant features to streamline the datasets for analysis
- Create a new primary key by using the Purchase Order Number + Purchase Line to identify the every single purchase order and positioned at beginning of the dataframe
- Restricting the time frame to include only entries in the 'Days Late' column that are less than or equal to 250 days.
- Merging datasets by adding relevant features such as supplier performance features that can be used for the evaluation of the supplier risk, part master categories by using the part master, fixed prices and penalties by using Contract agreement dataset.
- Encoding categorical variables into a machine-readable format using dummy variables.
- Dropping the missing values
- Dropping imprecisely purchasing transactions, namely the ones that are with dates out of range, or without complete information, mainly due to ERP migration, as we confirmed with the Procurement team.

III. Modeling & Performance Evaluation

Feature Selection:

Basically, we are going to take attributes of the main actors in the procurement process, the same ones that actively intervene in each transaction, such as:

-Part Number: Type of material, taxonomy.

-Supplier: Performance.

-Specific Transaction information: Lead time expected, order value, quantity, location.

After that we will test the models, in order to find the ideal model according to the scope.

Performance Evaluation:

After the dataset is ready, we split the data into training and testing data to evaluate our models; 30% will be allocated to the test set while 70% will be used in the training set. We ensure that our splits are randomized by using the 'random_state' function which will determine reproducibility. We then initialize the multiple classifiers and evaluate each model's performance using k- fold cross validation. Using classification models such as logistic regression (for binary classification) which estimates the probability that an instance belongs to a particular class, KNN - which classifies instances based on majority class among their k nearest neighbors. Gaussian Naive Bayes, Decision Tree Classifier and Random Forest methods are included in this model as we want to check which method gives us the most optimal results. Cross validation evaluation for each classifier is checked using k-fold cross validation and the AUC ROC scores are calculated using 10-fold cross validation.

Trade off - Performance and Interpretability

By using more features we increase performance (not consistently), but we decrease the interpretability. Since the dummy features that we consider to increase the performance is

Taxonomy, we can simplify it by using just Order Type , as we achieve an appropriate performance, preferring a simpler and efficient model, later on we can take the taxonomies for a trend distribution.

Results , selected Model:

Our results (Table 1.1.) show that Random Forest achieved the highest AUC mean (91.25) and accuracy mean (84.43) among the evaluated classifiers, indicating strong performance in both metrics. It also has relatively low standard deviations, suggesting consistent performance across folds. Decision Tree Classifier achieved a lower AUC mean compared to Random Forest (78.99) but still showed respectable performance. However, it has a slightly higher standard deviation in both AUC and accuracy compared to Random Forest. KNN achieved lower AUC mean (76.27) and accuracy mean compared to Random Forest and Decision Tree Classifier. It also exhibits higher standard deviations, indicating more variability in performance across folds. However, Gaussian Bayes and Logistic Regression are giving the lowest AUC mean measures hence we will use hyperparameter tuning for the first 3 models only.

Table 1.1.

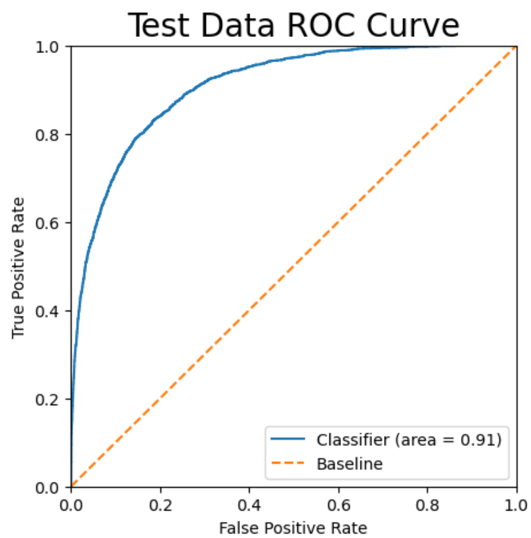
	Algorithm	AUC Mean	AUC STD	Accuracy Mean	Accuracy STD
4	Random Forest	90.83	0.63	83.87	0.60
3	Decision Tree Classifier	80.65	1.06	81.25	0.69
1	KNN	76.27	1.27	73.22	1.06
0	Logistic Regression	59.87	5.36	66.64	1.21
2	Gaussian NB	51.41	0.83	64.82	3.83

For hyperparameter tuning for the random forest model, we are using randomized search or grid search cross-validation. Each hyperparameter is specified with a range of values to explore during the hyperparameter tuning process. By defining this random grid, we can perform a hyperparameter search to find the combination of hyperparameters that optimizes the performance of the Random Forest classifier. There are 100 different combinations of hyperparameters (n_iter = 100), and for each combination, the model is trained and evaluated using 10-fold cross-validation (cv = 10). Since each combination is evaluated using 10-fold cross-validation, there are 10 fits (or training sessions) for each combination. Therefore, for 100 combinations, the total number of fits is 100 * 10 = 1000. Once the search is complete, we access the best hyperparameters found using 'rf_random.best_params'.

The best Random Forest classifier obtained from the randomized search is instantiated with the hyperparameters that yielded the best performance on the training data. The instantiated Random Forest classifier (best_rf) is trained on the entire training dataset (X_train, y_train) using the fit method. The predict_proba method is used to obtain probability predictions for the training data. This method returns the probability estimates for all classes. Additionally, the predict method is used to obtain class predictions based on the trained model. Similar to the training data, probability predictions and class predictions are obtained for the test dataset (X_test) using the predict_proba and predict methods, respectively.

We then calculate the AUC score on the training data to evaluate how well the Random Forest classifier discriminates between the positive and negative classes; a value of 0.9946 shows the classifier has near-perfect discrimination ability and it assigns higher predicted probabilities to positive instances compared to negative instances. The AUC score on training data demonstrates the high quality and effectiveness of the Random Forest classifier in the classification task at hand. Testing out the F1 score to check the accuracy on the binary classification model - a value of 0.946 shows that the Random Forest classifier is highly accurate and reliable in its predictions on the training data. It rarely misclassified negative samples as positive (low false positive rate) and also rarely misses positive samples (low false negative rate). The model is performing very well on the training data, achieving a high level of accuracy and effectively distinguishing between the positive and negative classes. Coming to the test data measures, we can evaluate how well the Random Forest classifier generalizes unseen data and its ability to discriminate between the positive and negative classes in the test dataset based on the AUC and F1 score. An AUC score of 0.906 (Table 1.2) shows that our model is performing very well on the test data, demonstrating its ability to effectively discriminate between the positive and negative classes and its strong generalization capability. We can evaluate that the model has captured meaningful patterns and relationships in the training data that are applicable to unseen instances.

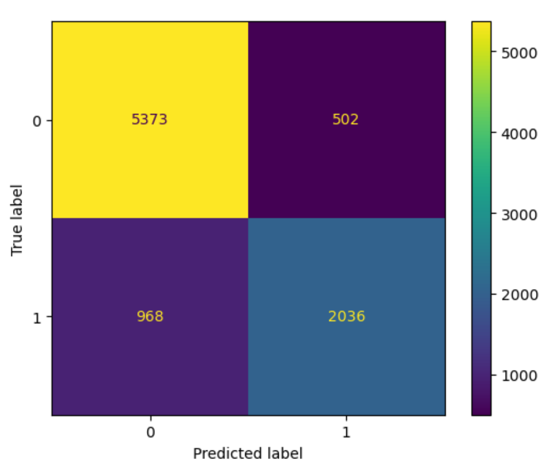
Table 1.2



The f1 score for the test data is 0.735 which is significantly less than our training data - however it might be useful to note that it might have been a result of overfitting before tuning our model and now gives a reflection of how accurately the data will be predicted via our model.

We have used the confusion matrix to visualize our true positive, true negative, false positive and false negative classes (Table 1.3). Just by looking at the graph, we can see that true positives and true negatives scores are much higher, showing that the random forest model is dividing the classes accurately. The accuracy score for the model (5383+2030/5383+2030+492+974) - 0.83, showing 83% of the instances in the dataset were correctly classified by the model.

Table 1.3



We will now check the other two models; decision tree classifier and KNN to see if they perform better or equivalent to the Random Forest Classifier. We tune the decision tree and use the grid to search for the best hyperparameters. There are 432 different combinations of hyperparameters in the grid (random_grid_dt), and for each combination, the model is trained and evaluated using 10-fold cross-validation (cv = 10). Since each combination is evaluated using 10-fold cross-validation, there are 10 fits (or training sessions) for each combination. Therefore, for 432 combinations, the total number of fits is 432 * 10 = 4320. The best decision tree classifier is instantiated with the hyperparameters that yielded the best performance on the training data. The decision tree classifier is trained on the entire training dataset using the 'fit' method. The 'predict_proba' method is used to obtain probability predictions for the training data and similarly, probability predictions and class predictions are obtained for the test dataset. The training AUC score is 0.998 which states the decision tree classifier has almost perfect discrimination ability. It shows exceptional quality and effectiveness of the model on the training data. The f1 score shows that the Decision Tree classifier performs very well on the training data, achieving a high level of accuracy and effectively distinguishing between the positive and negative classes.

Comparing this to the test data, AUC for the testing data for Decision Tree; it decreases to 0.80 and the f1 score dorps to 0.706 - there may be several reasons for this, overfitting might be the prevalent one, the model have adjusted to fit the training data closely, capturing noise specific to the training dataset which does not fit with the new, unseen data.

The confusion matrix shows that the accuracy score for the model has decreased - 0.80 (5120+2055)/(5120+2055+755+949). 80% which states that the model is a good fit but Random Classifier might be better based on these scores.

Coming to the last model, we tried KNN testing for our model; A base KNN classifier is initialized and 'GridSearchCV' is used to perform an exhaustive search over a specified parameter grid. The grid search model (kn_random) is fitted to the training data (scale(X_train.values), y_train). This will train and evaluate the KNN classifier using different values of n_neighbors and identify the best value. We can see there are 98 different values of n_neighbors in the grid, and for each value, the model is trained and evaluated using 10-fold cross-validation (cv = 10). Since each value of n_neighbors is evaluated using 10-fold cross-validation, there are 10 fits (or training sessions) for each value. Therefore, for 98 values of n_neighbors, the total number of fits is 98 * 10 = 980. The best KNN classifier obtained from the grid search is instantiated with the optimal value of n_neighbors (n_neighbors = 3). The instantiated KNN classifier (best_kn) is trained on the entire training dataset (scale(X_train.values), y_train) using the fit method. The high AUC score (0.95) indicates that the KNN classifier accurately distinguishes between the two classes in the training data. This suggests that the model has captured meaningful patterns and relationships in the data, leading to strong predictive performance. A high f1 score of 0.83 indicates that the classifier maintains a good balance between minimizing false positives and false negatives.

However, when we use the KNN model on our test data, the AUC and f1 score reduce to 0.70 and 0.54 respectively, showing that the KNN might not be the optimal model when it comes to our dataset. Looking at the accuracy score from the confusion matrix - 0.72 (4905+1498/4905+1498+970+1506) shows that the KNN classifier made correct predictions for approximately 72% of the instances in the test dataset.

Based on the metrics discussed above for the three different models; we will be using the Random Forest Classification Model since it has the highest accuracy ratios for the training and test data.

IV. Findings, trends and Managerial decisions

In this project, we are focusing on how to measure the criticality in terms of purchase transactions, which occur in terms of compliance with the Supply In Full On Time – SIFOT, a key piece and indicator in procurement management and the main indicator for monitoring the Expediting team. Taking into consideration the attributes of each transaction such as Supplier (Supplier History), purchase category, Amount, etc.

We will now create risk probabilities to store the test data along with the predicted probabilities of belonging to certain classes. We add a new column 'probability' to the DataFrame, which stores the predicted probabilities of the positive class - of being late, obtained from test_probs[:, 1]. These are the probabilities predicted by our Random Forest classifier model for each sample in the test data to belong to the ‘is late’ class. Sorting risk_prob_df by the 'probability' column in ascending order will rearrange the DataFrame so that the rows are ordered based on increasing probabilities of belonging to the positive class. Now, risk_prob_df is sorted with the samples having the lowest predicted probabilities at the top and those with the highest predicted probabilities at the bottom. This sorted DataFrame can provide insights into the distribution of predicted probabilities across the test dataset. By using the rank() method with the method='first' parameter, we are ensuring that ties are broken by assigning each value a unique rank in the order they appear in the DataFrame. Now, each row in risk_prob_df will have an associated rank based on its predicted probability.

We are using the np.select() function from NumPy to categorize the predicted probabilities into different risk levels based on predefined conditions. We created a list of conditions based on ranges of predicted probabilities where each condition checks whether the probability falls within a certain range. We have also created a list of values corresponding to each condition. These values represent the risk levels assigned to samples based on which condition they satisfy. We are adding a new column 'Risk Level' which stores the risk level assigned to each probability. We grouped the DataFrame risk_prob_df by the 'Risk Level' column using the groupby() function. This operation splits the DataFrame into groups based on the unique values in the 'Risk Level' column. Each group contains rows that have the same risk level assigned. We then extract individual groups from the grouped DataFrame grouped based on their risk levels using the get_group() method. Each resulting DataFrame (df1, df2, df3, df4, df5) contains the rows corresponding to a specific risk level. We input the vendor_dist(vendor_id) name that takes a vendor_id as input and aims to compute the distribution of occurrences of that vendor across different risk levels (represented by DataFrames df1 to df5). This function effectively provides the distribution of occurrences of a specific vendor across different risk levels.

We create a dictionary named 'vendor_dist_dict' to store the distributions of occurrences of each vendor in the list of vendors across different risk levels. We will iterate over each vendor ID in the vendors list, computing their distribution across risk levels using the vendor_dist() function we defined earlier, and then storing the results in the dictionary. This approach allows us to analyze the involvement of specific vendors across various risk levels predicted by the model.

Managerial Decision:

General Scenario

After creating the risk probability dataframe, we want to check whether certain suppliers should be replaced or not, based on their delivery services. If a shipment arrives late, MEC will be forced to sell at a discount rate of 7% whereas on time deliveries are sold at full price. The expected discount rate would be the probability of late delivery multiplied by the 7% discount rate - this will help in estimating the potential discount considering the probability of risk associated with each row in the dataset. We are estimating a $1400 cost if we do decide to change the vendor and we have to compare whether the profit of replacing the vendor will outweigh the costs or not.

We calculate total orders, total order value and quantity-weighted average discount for each vendor, selecting vendors whose weighted average discount exceeds or is equal to 0.05 (5%) which is MEC's average resale margin. These vendors are considered tardy and unprofitable on average, likely indicating they have higher discounts relative to their total order quantities. The resulting data frame 'tardy_vendors' contains information about vendors with potentially higher-than-average discounts relative to their order quantities, helping to identify the ones that might require further attention or investigation.

We calculate the total average order discount which is 1.8%. Selling deliveries at a 5% of their premium value, we will check if any vendor has an average expected discount rate greater than 5%, and then further investigate the expected profit from switching to a market average vendor. After calculating the profit increase from switching to the average vendor, 5 vendors will have profit greater than $1400: 140198, 15293, 105596, 65306, and 128121 (*Table 1.4*). The total profit increase to MEC would be $138,074.63 which outweighs the cost by a large margin.

Table 1.4

Vendor ID	Order Value	Weighted Discount	Current Vendor Total Sales	Average Vendor Discount	Possible Vendor Total Sales	Profit Increase from Switching to Average Vendor
140198	1.498228e+06	0.067372	1.467154e+06	0.018435	1.544139e+06	76984.858984
15293	1.144425e+06	0.065743	1.122647e+06	0.018435	1.179494e+06	56847.286363
105596	1.190029e+05	0.059434	1.175267e+05	0.018435	1.226496e+05	5122.937314
65306	8.892000e+04	0.058762	8.787962e+04	0.018435	9.164481e+04	3765.183214
128121	6.103305e+04	0.055173	6.054894e+04	0.018435	6.290331e+04	2354.366193

Contract Agreement Case Scenario

We know that one of the main contracts that the company has basically lies in ensuring a fixed price according to each part number, under a delivery time commitment.

Therefore we consider:

- Filter the previous dataset, with the part numbers associated with the contract number
- Include fixed price
- We do not include the delivery time, since this is previously associated with the calculation of the probability of being outside the promise date.

For the managerial decision, we should follow the following formulas :

Expected revenue = *[Revenue - [((Revenue *P(Late) * 7%)-(Revenue *P(No-Late)) * 7%) + Order Value]]* Qty*

Ideal Revenue = *(Price - Order Value)* Qty*

By using the expected revenue and the ideal revenue, we can measure the impact, as follow:

% Variation = $(Ideal\ Revenue\ -\ Expected\ Revenue) /\ Ideal\ Revenue$

Δ = *Ideal Revenue -Expected Revenue*

As we can see in the table 1.5, by using the impact in the revenue, the Procurement and Expediting team, can make operational and strategic decisions, regarding the transactions (purchase lines) and the stakeholders(suppliers).

Table 1.5

```
             KEY  #P_Order  PO_Line  PORel Num  Order Qty_x  Received Qty IUM  \
1203  160154_22    160154       22          1            6             0  EA
1206  160154_14    160154       14          1            6             0  EA
1207  160154_14    160154       14          1            6             0  EA
1245   144228_9    144228        9          4            1             1  EA

      Order Value  Received Value  Total Order  ... Weighted Discount  \
1203      44.2962           0.000    116075.42  ...      3.785461e-06
1206      40.0842           0.000    116075.42  ...      6.578014e-06
1207      40.0842           0.000    116075.42  ...      6.578014e-06
1245     124.2540         124.254    486613.57  ...      5.354670e-08

      Total_Orders  Weighted Discount for Total       Origen   Price  \
1203     295507549                 1.131894e-10  Contrato MV  131.98
1206     295507549                 1.966898e-10  Contrato MV  292.50
1207     295507549                 1.966898e-10  Contrato MV  292.50
1245     295507549                 9.482463e-09  Contrato MV  497.02

     Expected Revenue Baseline Revenue % Difference     Difference  \
1203        74.619980          87.6838    14.898784    1227.999111
1206       223.682550         252.4158    11.383301    2700.925500
1207       223.682550         252.4158    11.383301    2700.925500
1245       323.614069         372.7660    13.185733  355565.069948

     Impact Category
1203      Low Impact
1206      Low Impact
1207      Low Impact
1245        Critical
```

V. Conclusions & Recommendations :

- The presented model develops adequate accuracy and performance, considering interpretability, under the framework of relationships and patterns in the main actors of the acquisition process.
- The model provides insights, which consider beyond SIFOT, materializing the impact on revenue of the delay in the delivery of spare and wear parts.
- At an operational level, considering the workload of the Expediting team, the model provides necessary insights (probability) and impact, so that the team centralizes efforts in monitoring and appropriate postures for the treatment of these transactions.
- At a strategic level, the model provides insights regarding inadequate suppliers for the respective sourcing. Thus, it also provides adequate tools for long term agreements with suppliers that can help transmit the risk in reference to the penalties applied, ultimately ensuring an adequate level of service.

Contribution Statement

CS - Model Selection, General Scenario in Management Decision, Code Revision

KL - Target Variable Collection, Feature Selection/Dummy Variables, Model Selection/CV, Model Performance and Evaluation, Risk Level Classification, Vendor ID vs. Risk Level Trends

MC.- Business and problem understanding, exploratory analysis and data cleaning, Managerial decision -contract agreement scenario, conclusions.

OJ-Background information, business overview, understanding and data cleaning report.

AH-Performance evaluation, risk probability, managerial decision analysis report