Introduction

A Data Science consulting company needs to help reduce trucks' air system maintenance costs, from a recent outsourced company, using its own historical data most efficiently.

When a truck is sent to maintenance, the following scenarios can happen:

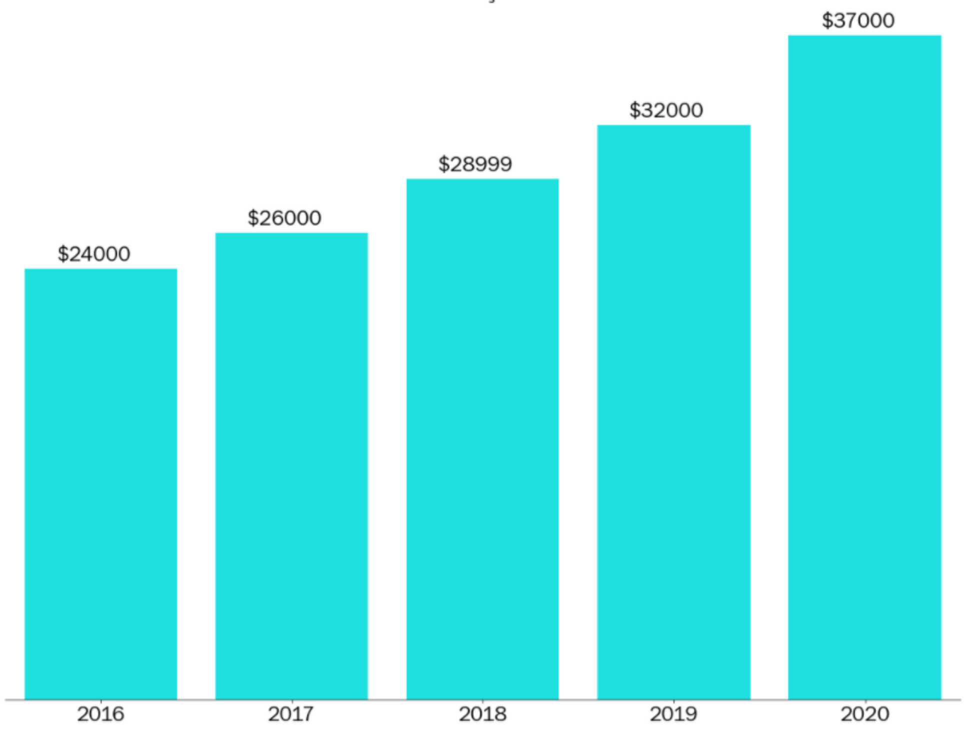
a. $10 will be charged if the truck doesn't show any defect on its air system;

b. $25 will be charged if the truck actually shows defect on its air system;

c. $500 will be charged if the truck has a defect on its air system but doesn't get sent to maintenance.

With this information in mind, the company wants to apply a machine learning model to predict its trucks' maintenance needs.

The annual maintenance cost so far is shown in the graph below.



About the data

Two datasets were provided: The training dataset (air_system_previous_years.csv), which includes data up to the year 2022 with a total of 178 columns, and the validation dataset (air_system_present_year.csv).

The Solution

The solution to the problem was designed and executed following the steps of CRISP-DM.

After the business understanding and data understanding phases, the Data Preparation phase began.

In this phase, the target variable was separated from the training dataset, and its values were converted from text ('pos' and 'neg') to integer values (1 for the 'pos' class and 0 for the 'neg' class). At this point, a significant imbalance in the data was identified, as the 1 class represents only 1.6% of the total observations.

Considering that there is no information about the meaning of each variable, unsupervised methods were used to create additional variables that could potentially help the models identify air system failures in trucks. Due to the significant class imbalance, the first model used to generate a new feature was the Isolation Forest, an unsupervised tree-based method for anomaly detection. A pipeline was then created to prepare the data, and the model was applied to generate the 'anomaly_scores' feature. After application, the Isolation Forest was saved as an artifact (in .pkl format) for later use in the validation dataset.

The next step was to create clusters that could capture the maximum number of observations with the 1 class. For this step, another pipeline was necessary, this time for the application of Principal Component Analysis (PCA) to reduce the data before applying clustering techniques. To evaluate the optimal number of clusters, the Silhouette method was used, adjusted to K-Means. The result of this step indicated that 2 or 3 clusters would be ideal, but it was decided to validate this information using a more robust model, such as the Gaussian Mixture Model (GMM). Using GMM and the Bayesian Information Criterion

(BIC) as a metric, the optimal number for the cost-benefit between clusters and model complexity was found: 9 clusters.

The GMM was then applied to the training data and saved as an artifact for use in the validation set.

With the new features created, the training data was submitted to a new pipeline to prepare the data for the modeling phase later on. In this pipeline, the data was standardized and transformed using quartiles to try to transform their distributions as close as possible to a Gaussian distribution. With the data prepared, a simple feature selection step was performed to remove variables with high correlation and low variance. After this step, the data was ready for the modeling phase.

The models to be tested were based on boosting, as they are highly robust to class imbalance. Three baseline models were then created: XGBoost, GradientBoosting, and LightGBM. The models were evaluated based on the maintenance cost their predictions would generate, penalizing false negatives (FN). For this, a custom cost function was created as follows:

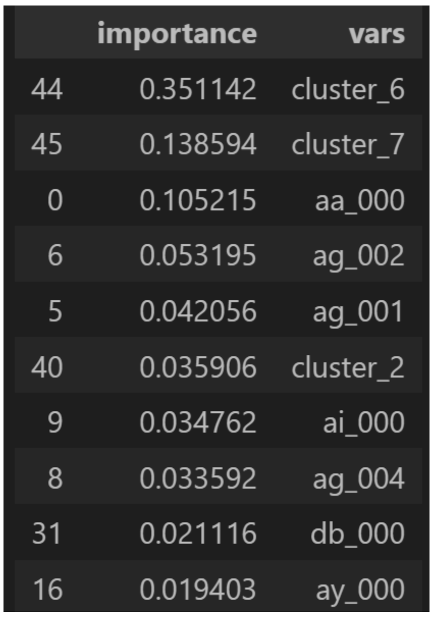
Cost = False Positive * 10 + True Positive * 25 + False Negative * 500

This function was also used in the hyperparameter tuning stage, where Optuna was used along with stratified cross-validation to better assess model generalization.

At the end of this stage, the model that best managed to minimize maintenance costs was XGBoost, generating a maintenance cost based on the training set of $15,570.

Finally, using the artifacts generated during the development, the same transformations were applied to the validation set so that the winning model could be applied using its best configurations. After application, the maintenance cost on the validation set was $29,065, representing a reduction of approximately 21%, which answers the proposed question “Can we reduce our expenses with this type of maintenance using AI techniques?”.

For the second proposed question “Can you present to me the main factors that point to a possible failure in this system”, the developed model choose the following variables as the most important to predict the maintenance costs:



	importance	vars
44	0.351142	cluster_6
45	0.138594	cluster_7
0	0.105215	aa_000
6	0.053195	ag_002
5	0.042056	ag_001
40	0.035906	cluster_2
9	0.034762	ai_000
8	0.033592	ag_004
31	0.021116	db_000
16	0.019403	ay_000

Challenge Activities

To solve this problem we want you to answer the following questions:

1. What steps would you take to solve this problem? Please describe as completely and clearly as possible all the steps that you see as essential for solving the problem.

   The answer to this question was presented in “The Solution” section of this document.

2. Which **technical** data science metric would you use to solve this challenge? Ex: absolute error, rmse, etc.

   To solve this challenge, I would use the cost function metric. Specifically, I would develop a cost function that incorporates the different costs associated with preventive maintenance, unnecessary inspections, and corrective maintenance. This cost function would allow us to optimize the decision-making process regarding when and which trucks should be sent for maintenance, aiming to minimize the overall maintenance costs.

3. Which business metric ***would*** you use to solve the challenge?

   Some custom metrics could help me measure the effectiveness of my solution on a business perspective, such as:

   Cost per Truck: The average maintenance cost per truck, which helps in understanding the cost efficiency on a per-unit basis.

   Maintenance Frequency: The number of maintenance events over a given period, that helps balancing between preventive and corrective maintenance.

   Downtime Reduction: The reduction in truck downtime due to breakdowns, which impacts overall operational efficiency and delivery reliability.

4. How do technical metrics relate to the business metrics?

Technical metrics and business metrics are interconnected, as improvements in technical metrics typically lead to better business outcomes. One solid example would be the relation between Cost Function Optimization (Technical metric) and the Total Maintenance Cost Reduction (Business Metric). By optimizing the cost function, which includes costs of preventive maintenance, unnecessary inspections, and corrective maintenance, we aim to minimize the total maintenance costs.

5. What types of analyzes would you like to perform on the customer database?

Trend Analysis, Descriptive Statistics, Distribution Analysis, Segmentation using clustering, Operational Efficiency Analysis (Maintenance Efficiency/Downtime Analysis).

6. What techniques would you use to reduce the dimensionality of the problem?

Since this is a supervised problem, I would use Feature Selection techniques.

7. What techniques would you use to select variables for your predictive model?

   Techniques such as Recursive Feature Elimination (RFE), removing high correlated features, removing low variance features, information value or removing low feature importance features through Random Forests/Decision Trees.

8. What predictive models would you use or test for this problem? Please indicate at least 3.

   XGBoost, LightGBM, Random Forests and Logistic Regression

9. How would you rate which of the trained models is the best?

   The model that produces the lowest maitenance cost (minimizes the loss function).

10. How would you explain the result of your model? Is it possible to know which variables are most important?

I would explain by pointing out how changes in each of the features impacts que final maintenance costs or the probability of a air system failure. By using a tree-based model, I could use the feature importance from the model to explain which variables are most important. On a Regression model (logistic regression), I could use the beta (weights).

11. How would you assess the financial impact of the proposed model?

I would compare it with the current model (even if there isn't a current model) using business metrics.

12. What techniques would you use to perform the hyperparameter optimization of the chosen model?

I've used the Optuna library with a StratifiedKFold, but Bayesian Optimization would solve the problem aswell.

13. What risks or precautions would you present to the customer before putting this model into production?

Before putting the model into production, I would inform the customer about potential risks such as model performance drift, where the model's effectiveness may degrade over time due to changes in data patterns, data quality and continuous monitoring.

14. If your predictive model is approved, how would you put it into production?

Using cloud services, like Amazon SageMaker to create an inference endpoint. Then, configure the model in SageMaker using the model stored in S3. Once the endpoint is deployed, test it using the SageMaker SDK or boto3 by sending data to the endpoint and obtaining predictions.

15. If the model is in production, how would you monitor it?

I would monitor it through business and costs metrics, using visualization tools (like Grafana) and logging systems (like AWS CloudWatch)

16. If the model is in production, how would you know when to retrain it?

Besides cost and business metrics, Data Drift Detection using statistical tests to compare changes in data distribution could be a good indicator to retrain the model. Another solution to this problem is a fixed scheduled retraining, either event based or time based.