UNIVERSITY OF PISA

Master's Degree Artificial Intelligence and Data Engineering

Business and Project Management Course

Project Feasibility Prediction Using Artificial Intelligence

Work Group:
Arduini Luca
Liu Chang

ACADEMIC YEAR 2022/2023

INTRODUCTION

Our project for the Business and Project Management course aims to implement an artificial intelligence model that assists project managers in predicting project outcomes based on selected parameters during the planning phase. These parameters include budget, estimated timelines, and various others. The objective is to provide an output that indicates the predicted outcome of the project, whether it will fail, encounter issues (such as delays or cost overruns), or proceed smoothly. Additionally, we intend to include an infographic that highlights the variables that have the most significant impact on each prediction.
This interactive approach will empower project managers to make data-driven decisions and optimize project planning and execution.

The code is available on GitHub at: github.com/LucaArduini/ProjectFeasibilityPrediction

CONTENTS

1 PROJECT MANAGEMENT AND ARTIFICIAL INTELLIGENCE

In this chapter, we will delve into the fundamental concepts of Project Management and Artificial Intelligence. Our exploration begins with an introduction to the concept of ”project management” and an examination of the valuable data it generates. We will then explore how the integration of artificial intelligence enhances project management practices, enabling intelligent decision-making and optimizing project outcomes.
By harnessing the power of AI and combining it with the principles of project management, we will be able to implement the predictive tool that is the objective of this project.

1.1 PROJECT MANAGEMENT

The prediction of project development outcomes plays a fundamental role for project managers as it allows them to make crucial decisions regarding cost estimates and expected duration. These decisions are of critical importance to ensure project success and ensure proper allocation of financial, time, and human resources.

Cost estimation is a key element in predicting the outcome of a project. A project manager must be able to accurately assess the costs associated with different project phases, including personnel costs, equipment costs, material costs, and external service costs. Evaluating estimated costs helps determine the economic feasibility of the project and identify potential financial problems that may arise along the way.
Furthermore, accurate estimation of the expected duration is essential for project success. The project manager must carefully assess the time required to complete each activity and define a realistic schedule. The estimated duration depends on various factors, such as project complexity, available resources, and dependencies between activities. An accurate assessment of the expected duration helps ensure that the project is completed within time limits and adequately planned.

To make realistic cost and duration estimates, the project manager must rely on accurate and relevant data. This may include analyzing similar projects undertaken in the past, involving industry experts, and collecting detailed information on the resources needed for each activity.

Choices regarding estimated costs and expected duration have a significant impact on the overall success of the project. Accurate cost estimation helps avoid unforeseen costs or excessive expenses that could jeopardize the financial sustainability of the project. Similarly, a realistic estimate of the expected duration allows for proper resource planning and helps avoid delays and setbacks that could negatively affect project objectives.

It is important to emphasize that cost and duration estimates are based only on information available at the time of project outcome prediction.

1.1.1 Parameters Derived During the Design Phase

During the project management phase, various variables are calculated that provide detailed indicators about the status, progress, and performance of the project. The calculated variables may vary depending on the project characteristics, industry, and specific objectives of the organization.
Below, we list the main parameters that are typically calculated during the project management phase.

One key aspect is **cost**, which includes the total budget allocated to the project, the actual cost incurred so far, the cost variance between the planned budget and the actual cost, as well as the estimated total cost at project completion.
Furthermore, **time** is evaluated, which includes the planned duration of the project, the actual time spent on completing tasks, the schedule variance between the expected duration and the elapsed time, as well as the task status indicating progress compared to the plan.
Resources are another important variable and involve the utilization of resources such as personnel, equipment, or materials, assessing how effectively they are being used, ensuring proper allocation of resources for different project activities, and identifying critical resources that could impact project performance or duration if not managed properly.
Additionally, it is crucial to analyze and manage **project risks**. This entails identifying and assessing potential risks that could have a negative impact on the project, including the probability of occurrence and the impact of such risks. Mitigation plans are developed to proactively address risks and reduce their negative impact on the project.
Finally, **project performance** is evaluated through indicators such as earned value, which measures the value achieved compared to what was planned, and planned performance and cost indices, which respectively compare actual time spent and cost with what was planned.

Analyzing these variables provides a comprehensive overview of the project's status and performance, enabling project managers to make informed decisions, make changes, or implement corrective measures if necessary, to ensure the project's success within the established constraints of time, cost, and quality.

1.2 PROJECT FEASIBILITY PREDICTION EXPLOITING AI

The main concept of this project is to develop an AI-powered tool to support project managers in predicting the outcome of project development using their own calculated data. Additionally, the tool will provide infographics highlighting the variables that have the greatest influence on the prediction. By leveraging this tool, project managers will have a valuable resource to make informed decisions and proactively address potential challenges, ultimately increasing the chances of project success.

To achieve this goal, our model will be trained on a dataset of projects, where the defined parameters we discussed in the previous paragraph are recorded, along with the corresponding outcomes. These outcomes indicate whether a project failed, encountered issues such as delays or cost overruns, or was successfully completed without major setbacks.
By leveraging artificial intelligence algorithms and machine learning techniques, our model will analyze this project data and uncover patterns and correlations between the chosen parameters and project outcomes. This analysis will provide valuable insights into the factors that contribute to project success or failure. In particular, when issues arise during project development, the model will help identify which parameters were underestimated or inadequately accounted for, leading to various problems.

The classification of projects based on their outcomes is our ultimate objective. To accomplish this, we have employed a class of neural networks known as classifiers. In our case, the classifier will take as input the parameters of a project and, based on them, assign the project to the appropriate class, where the classes represent the potential outcomes of the project's development.

1.2.1 Classifier

A classifier in machine learning is an algorithm that automatically orders or categorizes data into one or more predefined classes. These algorithms are designed to accurately assign input objects to different classes or categories based on specific features or attributes. The ultimate goal is to develop models that can learn from labeled training data and make precise predictions or decisions regarding unseen or new data.

The functioning of classification algorithms is based on the creation of a mathematical or statistical model that learns from the training data. During the training phase, the algorithm analyzes a set of labeled data where each object is associated with a known reference class. The algorithm aims to identify patterns or relationships between the attributes of the training data and their respective class labels.

The effectiveness of classification algorithms relies on the quality of the training data, the appropriate choice of the algorithm itself, and the correct selection of relevant features for classification. By carefully selecting and preparing the training data, we can enhance the classifier's ability to generalize and accurately classify unseen instances.

Once trained, the classification algorithm can be used to predict the class of new input objects that were not included in the training dataset. The algorithm applies the rules and models learned during the training phase to correctly assign a class to the object based on its characteristics. This ability to classify new data enables us to make informed decisions and gain valuable insights in various domains, ranging from image recognition to customer segmentation and fraud detection.

Depending on your needs and data, there are several types of classification algorithms to choose from. Let's now provide a brief overview of the ones that have been used in the implementation of this Project.

Decision tree

A decision tree uses a tree-like model to make predictions or decisions based on input features. It's like a flowchart where each step leads us to a different outcome.

We start with a dataset that has examples labeled with different classes. The decision tree algorithm looks at the features of these examples and creates a tree structure based on them. Each internal node of the tree represents a feature, and each branch represents a possible value or decision based on that feature. The leaf nodes of the tree represent the predicted class labels. To make a prediction using a decision tree, we start at the root node and follow the branches based on the values of the features of the example we want to classify. We keep moving down the tree until we reach a leaf node, which gives us the predicted class label for the example.

Random forest

Random Forest is a powerful classification algorithm that combines the predictions of multiple decision trees to make accurate predictions.

Instead of relying on a single decision tree, Random Forest creates an entire forest of decision trees. Each tree is built on a randomly selected subset of the training data and a random subset of the features. This randomness helps create diverse trees that can capture different aspects of the data.
When making a prediction with Random Forest, each tree in the forest independently predicts the class label for the input. The final prediction is determined by taking a majority vote among all the trees. In other words, the class label that receives the most votes becomes the predicted label.

AdaBoost

AdaBoost, short for Adaptive Boosting, is a popular classification algorithm that combines the predictions of multiple weak classifiers to create a strong classifier. It is a boosting algorithm that iteratively improves the model's performance by focusing on the misclassified samples.

Initially, each sample in the training set is assigned equal weights. A weak classifier is trained on the data, and its performance is evaluated. The algorithm then adjusts the weights of the misclassified samples, placing more emphasis on those that were classified incorrectly. In the next iteration, another weak classifier is trained, giving more attention to the previously misclassified

samples. This process is repeated for a predetermined number of iterations or until the desired level of accuracy is achieved.
The final prediction is made by combining the predictions of all the weak classifiers. Each weak classifier is assigned a weight based on its performance, and these weights are used to determine the importance of each classifier's prediction. The combined result is a strong classifier that performs better than any individual weak classifier.

In the next chapters, we will explore how these classification algorithms have been utilized within this project to implement the classification tool.

2 IMPLEMENTATION

In this chapter, we will analyze the main phases of the development of this project and how we obtained the models to predict project outcomes.

2.0.1 Dataset Acquisition

The first phase of the project involved searching for a dataset to train the model. We needed a dataset that contained data from real completed projects, where for each project, the estimated parameters for its development and, most importantly, its outcome were available. Fortunately, we found a suitable dataset at this link!
This dataset consists of baseline scheduling data, risk analysis data, and project control data from projects developed between 2011 and 2018. The majority of these projects are related to the construction of infrastructure such as residential buildings or the development of IT devices. By utilizing this dataset, we aimed to capture a diverse range of project scenarios and outcomes to enhance the effectiveness and accuracy of our prediction model.

Having access to such comprehensive project data allowed us to analyze the relationship between various project parameters and their corresponding outcomes. We could uncover patterns and trends that shed light on the critical factors influencing project success or failure. This information served as the foundation for training our classifier algorithm, enabling it to learn from historical project data and make predictions about future projects.

2.0.2 Data pre-processing and model training

As mentioned earlier, depending on your needs and data, there are several types of classification algorithms to choose from. Since we didn't know in advance which one would be the best fit for our project, we decided to proceed with a trial and error approach. We tested one classification algorithm at a time, out of the three algorithms discussed in the previous chapter, by training a prediction model with it and conducting evaluative tests.
After conducting the evaluative tests and analyzing the results, we compared the performance of the various algorithms. This comparison allowed us to determine which algorithm was the most suitable for our specific project requirements and objectives.

To train a model, the first step is dataset preparation. In our case, we began by removing columns

that did not contain relevant values for prediction. Specifically, we eliminated columns such as the project's name, code, submitting company, and sector. Next, we focused on analyzing the rows, where further modifications were necessary. We had to delete rows that had missing values (NaN).
After completing the phase of removing unnecessary rows and columns, the dataset was ready to be used for model training. The second step involved splitting the dataset columns into two parts: one group containing the columns to be predicted (`y features`), which held the outcome values of the project, and another group containing the columns that serve as the basis for prediction (`x features`).
The third step involved dividing the dataset once again, this time based on rows. Around 70% of the rows were allocated for training the model, while the remaining 30% were reserved for testing it. This division ensured that we had a sufficient amount of data for training the model while also allowing us to evaluate its performance on unseen data during the testing phase.

Now that the dataset has been properly prepared to train the model, we proceed to actually train it. As previously mentioned, we will train it using the three classification algorithms that were introduced earlier, one at a time.

2.0.3 Performance evaluation

Once the three models have been trained, our objective was to evaluate their performance and determine which one among the three would be the best fit for our predictive model.
Multiple tests were conducted to assess their performance, but we prefer to provide a detailed overview of the results in the following chapter. However, we can give a small spoiler for now: all three algorithms resulted in the creation of a good predictive model, but among them there is an algorithm that leads the model trained with it to perform better than the others.

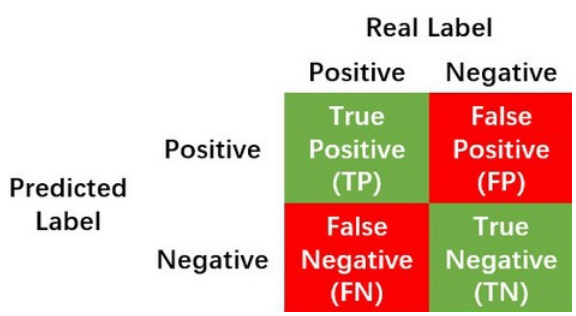
3 RESULTS

In this chapter, we will analyze the tests conducted to evaluate the performance of the three models trained with three different classification algorithms.

3.0.1 Accuracy score

The first evaluative test we conducted was the accuracy score of the classifiers, as it provides an immediate measurement of how well they are able to predict the given cases. We assessed the performance of the three models using the confusion matrix.

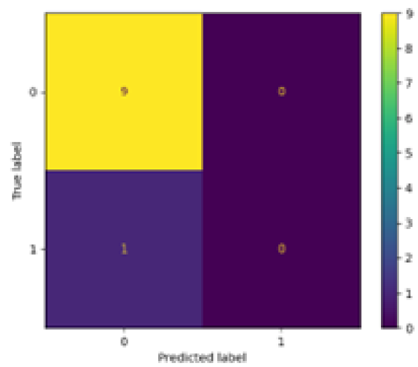
Before presenting the results obtained, let's first explain how they should be interpreted. In the cells of the matrix, the results of some classification tests will be reported. On the diagonal of the matrix, the counts of tests where the prediction correctly matches the outcome will be shown, while the cells outside the diagonal represent tests that are predicted incorrectly.



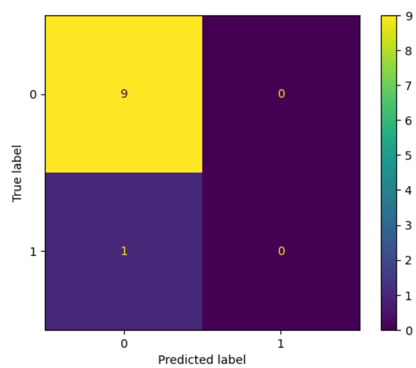
Once the tests have been conducted and the matrix is constructed, the accuracy score is calculated using the following formula:

$$\text{accuracy} = \frac{TN + TP}{TP + FP + TN + FN}$$

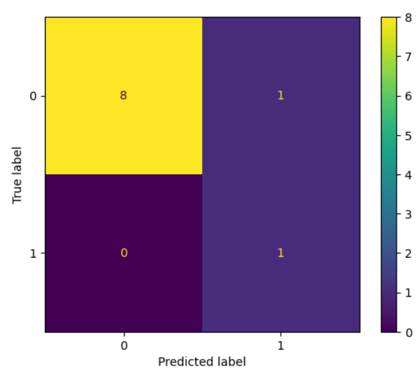
Let's now proceed to analyze the results obtained.



The decision tree model achieved an accuracy score of 0.9.



The random forest model achieved an accuracy score of 0.9.



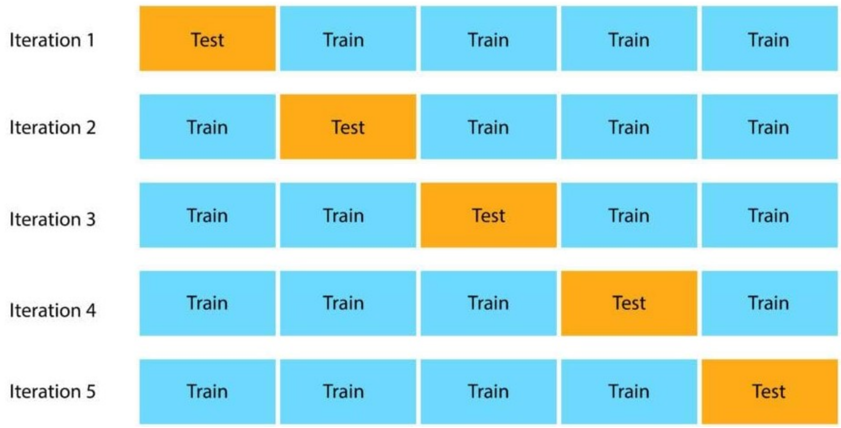
The AdaBoost model achieved an accuracy score of 0.9.

The three classifiers have the same accuracy rate. However, we notice that the three confusion matrices are different. This is because even though the errors made differ between models, the formula used evaluates them in the same way, as it should be.

Since no classifier outperforms the others in terms of accuracy, let's proceed with conducting further tests.

3.0.2 Cross Validation

The second test we conducted is k-Fold cross-validation. This involves first dividing the dataset into k subsets. Now, k iterations will be performed, where each time one of the k subsets is used as the test/validation set, while the remaining k-1 subsets are combined to form the training set. The error estimation is then averaged over all k trials to obtain the overall effectiveness of our model.



This approach ensures that each data point is included in the validation set exactly once and is included in the training set k-1 times. This significantly reduces bias as most of the data is used for fitting, and it also reduces variance as a substantial portion of the data is used in the validation set. By interchanging the training and test sets, the effectiveness of this method is further enhanced.

We have presented the results obtained from the execution of this test in the following table.

Classifier	**Average score**
Decision Tree	0.825
Random Forest	0.925
AdaBoost	0.899

From the results of this test, it can be observed that the Random Forest model exhibits slightly better performance compared to the other classifiers.

In the field of machine learning, we cannot rely solely on the results of a single test, as a model that performs well in one test may have inferior performance in other evaluation tests. Therefore, let's proceed with conducting additional evaluation tests to gain a more comprehensive understanding.

3.0.3 Paired T-Test

The paired t-test is designed to compare two classifiers simultaneously, allowing us to assess their performance relative to each other. This statistical test is commonly used in machine learning to determine if there is a significant difference between the performance of two classification models.

To conduct a paired t-test, we need to evaluate both classifiers on multiple test sets. While it may be impractical to have a large number of distinct test sets, we can utilize k-fold cross-validation as a workaround. By employing this technique, we can effectively evaluate each model on multiple subsets of the data, resulting in more reliable and robust results, particularly when working with limited data.

In our case, focusing on accuracy as the performance metric, we begin by calculating the difference in accuracies between the two models.
For each test set, we pair the accuracy of Classifier A with the corresponding accuracy of Classifier B. We make the assumption that the accuracies of both classifiers are sampled from normal distributions, each with an unknown variance.

The null hypothesis states that the accuracies of Classifiers A and B are drawn from the same underlying distribution. In other words, if the expected value of the difference in accuracies (E[diff]) is equal to zero, it suggests that there is no significant difference between the two models. However, if the accuracies are indeed sampled from two distinct distributions, the expected value of the difference in accuracies (E[diff]) will not be equal to zero. This implies that the models are truly different, and one may outperform the other.

The test follows these steps:

1. Take a sample of N observations (given from the k-fold cross-validation). The assumption is that these results come from a normal distribution with a fixed mean and variance.
2. Calculate the sample mean and sample variance based on the collected observations.
3. Compute the t-statistic, which measures the difference between the sample means of the two classifiers and takes into account the sample variances.
4. Utilize the t-distribution with N-1 degrees of freedom to estimate the probability that the true mean falls within a specific range. This provides an indication of the significance of the observed differences
5. Reject the null hypothesis at a chosen significance level (often denoted as p) if the computed t-statistic falls outside the critical region defined by the t-distribution. This critical region represents the interval of t-values that would lead to the rejection of the null hypothesis.

$$\text{Interval} = [-t_{\frac{p}{2},n-1}; +t_{\frac{p}{2},n-1}]$$

Formulas used:

$$\mathrm{diff}_i = \mathrm{acc_i(A)} - \mathrm{acc_i(B)}$$

$$\mathrm{sample\_mean} = m = \frac{1}{N}\sum_{n=1}^{N} \mathrm{diff}_i$$

$$sd = \sqrt{\frac{\sum_{n=1}^{N}(\mathrm{diff}_i - m)^2}{N-1}}$$

$$\mathrm{t\_statistic} = t = \frac{\sqrt{N} * m}{sd}$$

Achieved results:

Classifiers versus	Score
Decision tree vs Random forest	0.499
Decision tree vs AdaBoost	0.656
Random forest vs AdaBoost	0.499

Based on the results of this test, no model has prevailed over the others, as we followed the standard and chose a p-value threshold of 0.05. None of the models reached a p-value below this threshold, indicating that there is no significant difference in their performance.

3.0.4 Precision, Recall, and F-Score

Precision, Recall, and F-Score are performance metrics commonly used in machine learning and information retrieval to evaluate the effectiveness of a classification or information retrieval system. These metrics provide insights into the model's performance in terms of accuracy, completeness, and the balance between them.

The values used to calculate these parameters are obtained from the first test, specifically from the confusion matrices. In particular, we will refer to the values using the following acronyms: TP (True Positive), FP (False Positive), FN (False Negative) from the confusion matrix.

Precision: Precision is the measure of accuracy that indicates the percentage of correctly identified positive results out of the total predicted positive results. It quantifies how precise a classifier is in correctly predicting true positives relative to all the cases it identified as positive.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall: Recall, also known as sensitivity or true positive rate, is the measure that indicates the percentage of true positives correctly identified out of the total true positives in the dataset. It measures the classifier's ability to correctly detect positive cases.

$$\text{Recall} = \frac{TP}{TP + FN}$$

F-Score: The F-Score, also called F1-score, is a measure that combines precision and recall into a single value to provide an overall assessment of a classifier's performance. It is a harmonic mean of precision and recall and provides a balanced measure between precision and the ability to correctly detect positive cases.

$$\text{F-Score} = \frac{(1 + b^2) \cdot \text{Precision} \cdot \text{Recall}}{b^2 \cdot \text{Precision} + \text{Recall}}$$

Classifiers	Precision	Recall	F Score (b=2)
Decision Tree	1.00	0.9	0.92
Random Forest	1.00	0.9	0.92
AdaBoost	0.88	1	0.97

Even from this test, there is no clear winner.
Therefore, let's try using another parameter based on the graph, which is the AUC (Area Under the Curve) value calculated as the integral under the ROC (Receiver Operating Characteristic) curve.

3.0.5 AUC - ROC Curve

AUC - ROC curve is a performance measurement for the classification problems at various threshold settings. ROC (Receiver Operating Characteristic) is a probability curve and AUC (Area Under the Curve) represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1. By analogy, the Higher the AUC, the better the model is at distinguishing between patients with the disease and no disease.

When the AUC is 0.7 it means there is a 70% chance that the model will be able to distinguish between positive class and negative class, when is approximately 0.5, the model has no discrimination capacity to distinguish between positive class and negative class, when AUC is approximately 0, the model is actually reciprocating the classes, it means the model is predicting a negative class as positive class and vice versa.

The ROC curve is plotted with TPR against the FPR where TPR is on the y-axis and FPR is on the x-axis. TPR is the true positive rate, FPR is false positive rate.

$$\text{TPR} = \frac{TP}{TP + FN}$$

$$\text{FPR} = \frac{FP}{TN + FP}$$

These followings pictures show the ROC curve and the relative AUC score.

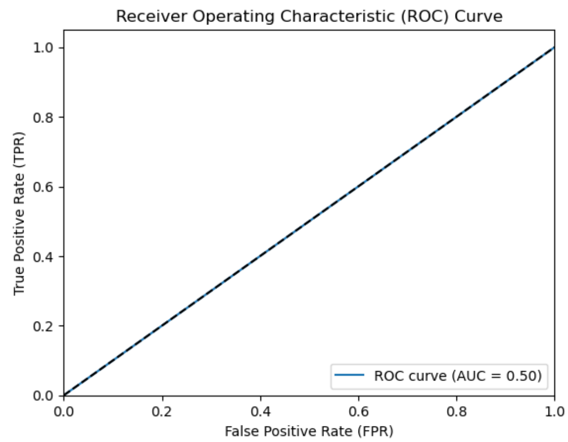


Figure 3.1. AUC-ROC curve for the Decision Tree model

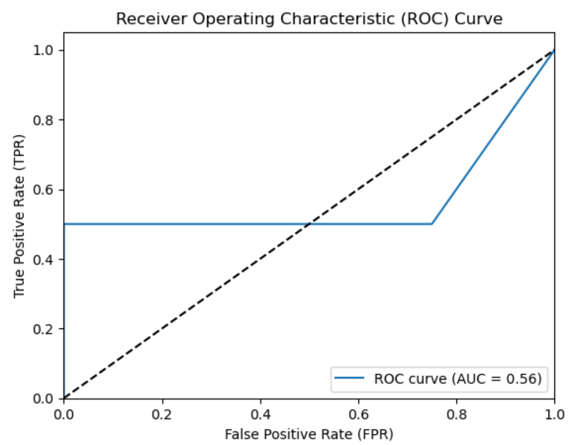


Figure 3.2. AUC-ROC curve for the Random Forest model

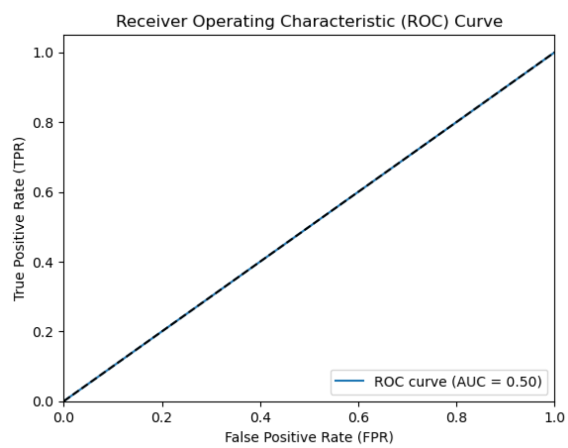


Figure 3.3. AUC-ROC curve for the AdaBoost model

From the results of this test, it is evident that the Random Forest classifier performs better compared to the other models.

3.1 EXPLAINABLE AI

As previously mentioned, in addition to predicting the outcome of a project, our intention is to provide an infographic as an output that highlights the variables that have the most significant impact on each prediction. This infographic will help project managers understand how the prediction is computed and provide insights into the variables that have been identified as influential in determining the outcome.

By visualizing the key variables, project managers can easily identify the factors that contribute to the predicted outcome and gain valuable insights into potential areas of concern or improvement. This information empowers project managers to make informed decisions and take necessary actions based on the identified problematic variables.

The variables that could potentially pose challenges are obviously among those provided as input to the model. Let's now examine the following tables to identify the variables that the project manager needs to provide as input to obtain a prediction from our model.

Real Cost	The actual cost incurred for the completion of a project.
Real Duration	The actual time taken to complete a project or activity.
P-Factor	A performance factor that measures the degree to which a project is meeting its planned objectives.
SPI (Schedule Performance Index)	A ratio that indicates the efficiency of the project's schedule by comparing the earned value (EV) with the planned value (PV).
SPIT (Schedule Performance Index Trend)	A trend analysis of the project's schedule performance over time.
CPI (Cost Performance Index)	A ratio that measures the cost efficiency of the project by comparing the earned value (EV) with the actual cost (AC).
SVT (Schedule Variance Trend)	A trend analysis of the project's schedule variance over time.
SV (Schedule Variance)	The deviation between the planned value (PV) and the earned value (EV), indicating whether the project is ahead or behind schedule.
CV (Cost Variance)	The deviation between the earned value (EV) and the actual cost (AC), indicating whether the project is under or over budget.
CRI-TAU (Cost-Related Index-TAU)	An index that measures the cost efficiency of a project by considering the actual cost (AC) and the target cost (TC) at a specific point in time.

CRI-RHO (Cost-Related Index-RHO)	An index that evaluates the cost performance of a project by comparing the earned value (EV) with the target cost (TC) at a specific point in time.
CRI-R (Cost-Related Index-R)	An index that assesses the cost efficiency of a project by comparing the earned value (EV) with the actual cost (AC) at a specific point in time.
SSI (Schedule Slack Index)	An index that measures the amount of flexibility or slack available in the project schedule.
CI (Cost Index)	A ratio that compares the actual cost (AC) with the budget at completion (BAC) to evaluate the cost performance of a project.
RI (Schedule Risk Index)	A measure of the level of uncertainty or risk associated with meeting project deadlines.
TF (Total Float)	The maximum amount of time that a schedule activity can be delayed without affecting the project's overall duration.
LA (Late Activities)	Activities that are scheduled to be completed after their planned completion date.
AD (Activity Dependencies)	Relationships between project activities that determine their sequencing and interdependencies.
SP (Schedule Performance)	A measure of how well the project is adhering to the planned schedule.
Under Over Budget	A comparison of the actual cost (AC) with the budget at completion (BAC) to determine if the project is under or over budget.
Early Late	A comparison of the planned start and finish dates with the actual start and finish dates of project activities to assess if they are early or late.
BAC (Budget at Completion)	The total planned budget for a project, representing the estimated cost of completing all project activities.
PD (Project Duration)	The total time required to complete a project, considering the planned start and finish dates of all activities.
Project Control	The process of monitoring and regulating a project to ensure it stays on track and meets its objectives.
Risk Analysis	The evaluation and assessment of potential risks and uncertainties that may impact a project's success.
Baseline Schedule	The original or approved version of the project schedule against which actual progress is measured and deviations are assessed.

One technique to achieve model explainability is to utilize feature importance, which provides insights into how input features influence the model's decision and final output. Each model has its own approach for calculating feature importance, as it is based on different algorithms.

In the following pictures, we can observe the feature importance calculated by different classifiers.

Decision tree

The feature importance of a decision tree is calculated by measuring the decrease in node impurity, weighted by the probability of reaching that node. The node probability can be determined by dividing the number of samples that reach the node by the total number of samples. The importance of a node, denoted as nij, is calculated as follows:

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)}$$

where:

- wj represents the weighted number of samples reaching node j
- Cj denotes the impurity value of node j
- Cleft(j) is the impurity value of the left child node of node j
- Cright(j) is the impurity value of the right child node of node j

The higher the value of nij, the more important the corresponding feature is considered by the decision tree model.

As shown in the Figure 3.4, the decision tree model has placed significant importance on the TF feature in making its prediction. If the model predicts a project failure, there is a high probability that considering the TF value could contribute to achieving project success. We recommend re-running the model with the same project, taking into account the TF, to assess if the model produces a more favorable outcome.

Random Forest

The feature importance in the Random Forest model is calculated by applying the formula for feature importance of a decision tree to each individual tree in the ensemble, and then taking the average of the feature importance values across all the trees.

As shown in the Figure 3.5, the decision trees in the Random Forest model have made their decisions based on different feature values. As mentioned earlier, the Random Forest model combines the decisions of multiple decision trees, each focusing on different features. Therefore, if the final prediction of the project is classified as a failure, it is highly likely that the values of these particular features played a significant role. To increase the chances of achieving a successful project outcome, it is advisable to consider and prioritize these specific values. Additionally, we recommend re-running the model with the same project data to evaluate if the model can provide a more favorable outcome.

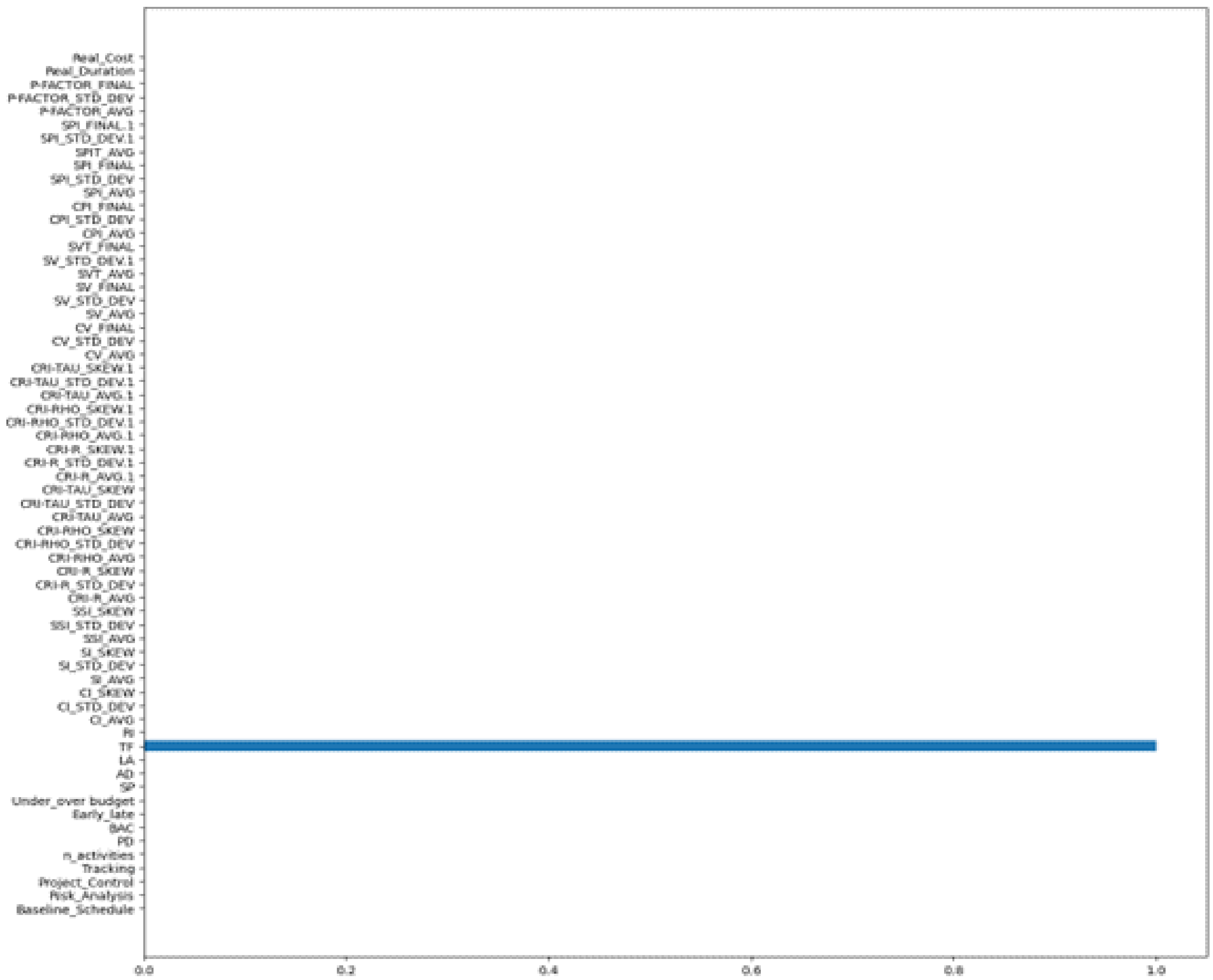


Figure 3.4. Most significant variables for the Decision Tree model

AdaBoost

The feature importance in AdaBoost can be calculated similarly to Random Forest. Each weak classifier's feature importance is weighted based on its performance in the array of classifiers. While both AdaBoost and Random Forest use decision trees as base estimators, there are key differences. In Random Forest, decisions are made by voting with equal weights, whereas in AdaBoost, each vote is weighted based on the performance of the corresponding classifier. The underlying principles of these two classifiers are distinct. To determine the feature importance in AdaBoost, we calculate a weighted mean of the feature importance values across all classifiers. This weighted mean represents the overall feature importance in AdaBoost.

As depicted in the Figure 3.6, the classifiers within AdaBoost have made their decisions based on different values. As mentioned before, AdaBoost relies on the weighted vote of multiple classifiers. Each classifier may prioritize different parameters in making its decision. Therefore, if the final project prediction indicates failure, there is a high likelihood that considering these specific values will be crucial for achieving project success. We recommend rerunning the model with the same

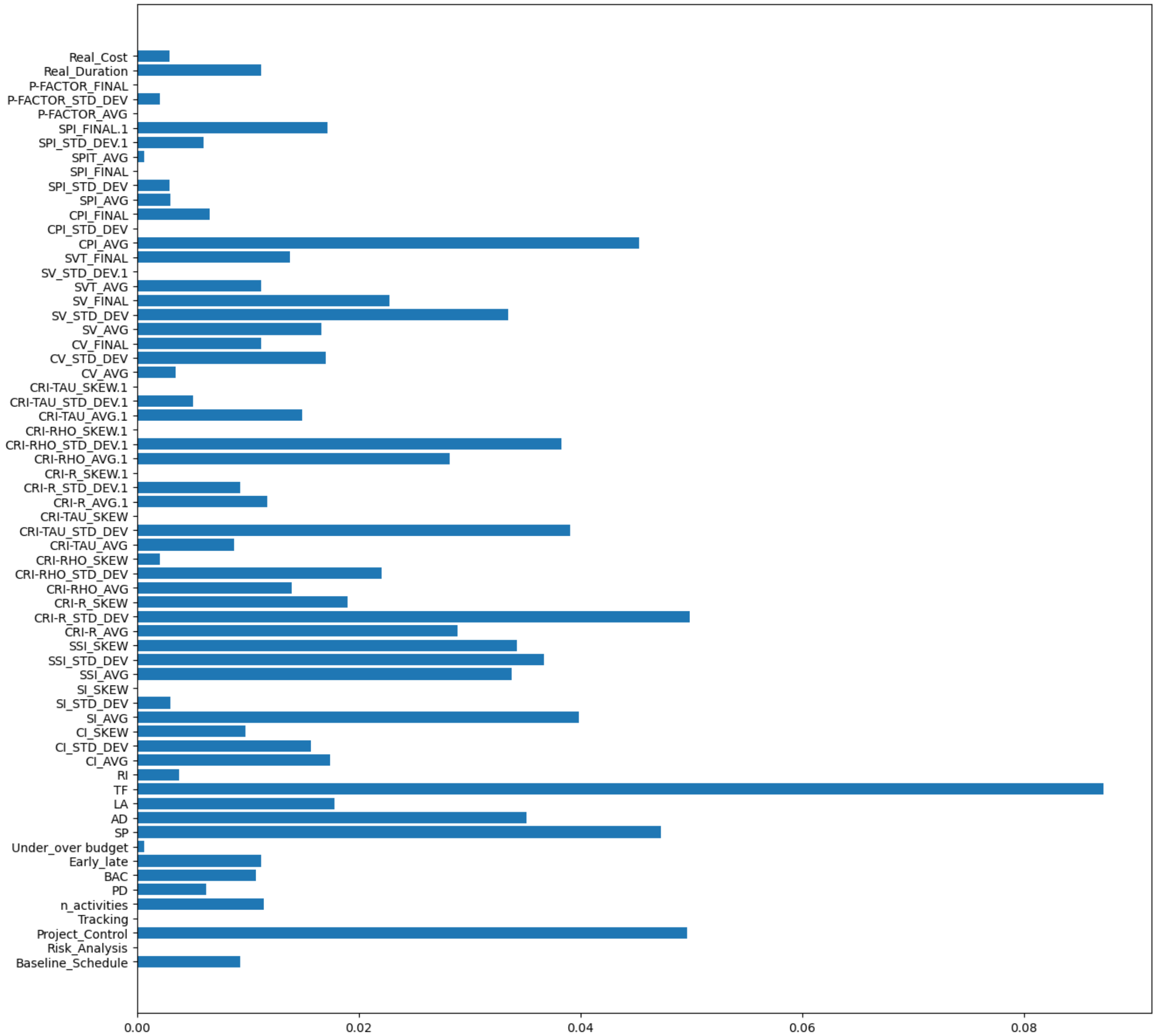


Figure 3.5. Most significant variables for the Random Forest model

project inputs to determine if the model yields more favorable results.

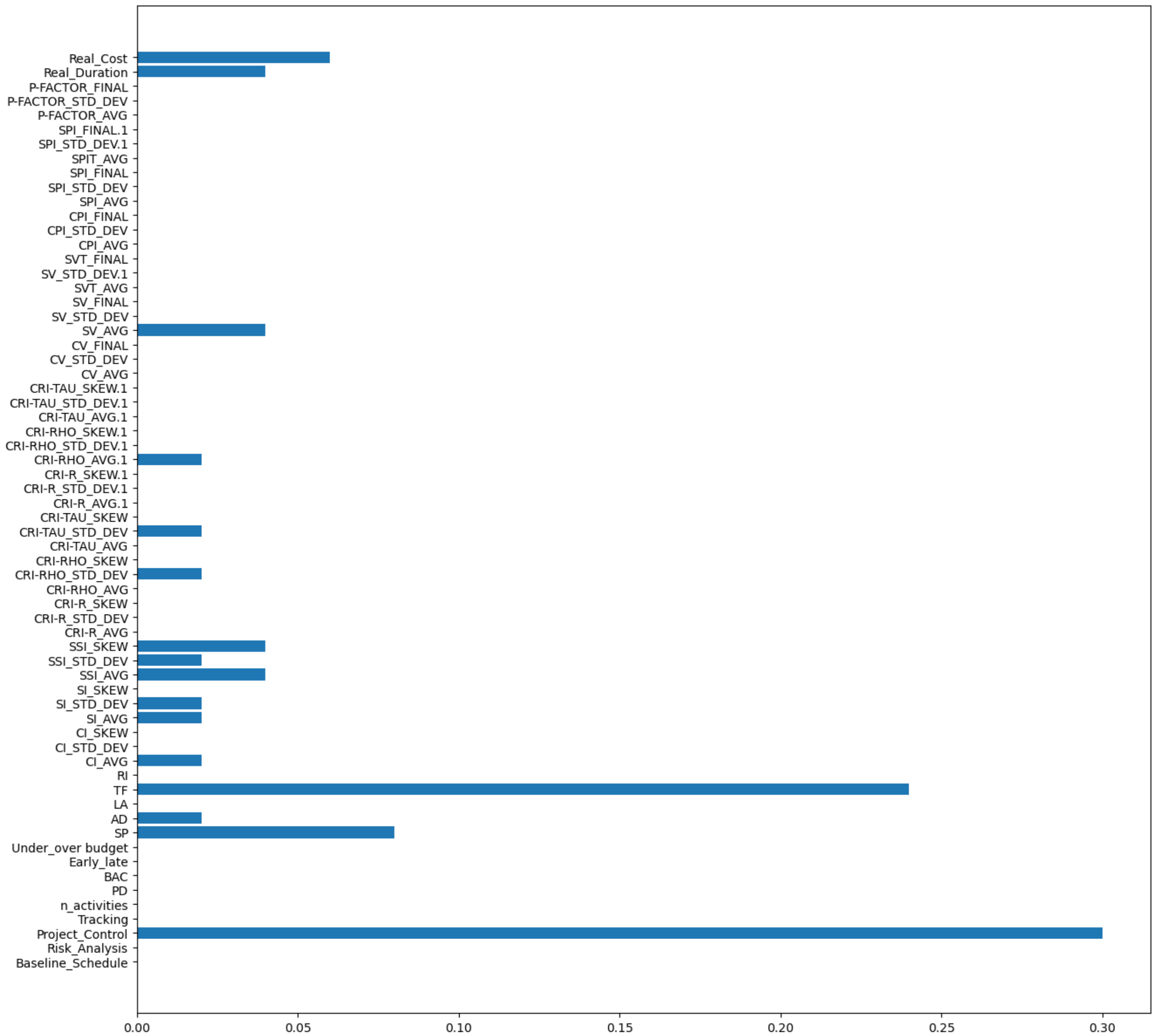


Figure 3.6. Most significant variables for the AdaBoost model

3.2 FINAL CONCLUSION

We conclude our project with satisfaction regarding the results obtained from the conducted tests. All three trained models demonstrate excellent performance in predicting the outcomes of the submitted projects. Among the three models, the Random Forest classifier stands out as having the best performance. However, as reiterated throughout this document, if this tool is to be used in practice, we recommend utilizing all three classifiers to obtain a comprehensive understanding of the predictions.

Regarding the explainability of the models we have developed, we believe that they provide a valuable tool for assisting project managers in reassessing parameters that our model identifies as influential. However, it is crucial to emphasize that the responsibility for making these important project decisions ultimately lies with the project manager. Our tool should be seen as a supportive tool rather than a decision-making authority. Its purpose is to aid the project manager in reevaluating their choices and gaining a deeper understanding of the factors that contribute to project outcomes. Considering the significant investments of time and resources involved, it is essential for the project manager to exercise their judgment and expertise in conjunction with the insights provided by our models.