[Data preparation 3](#_qbocegggw4pz)

[Materialising minable view 4](#_dtkpvx9qszjy)

[Model building 5](#_bco6pbipqusw)

[Evaluation 6](#_wlw1tljj6chm)

[Deployment mockup 8](#_m2shvs6jw3fv)

[Use of technology 9](#_db2mrmenjbsr)

[Seminar use 11](#_vxgdwvnm2qhf)

# Data preparation

The purpose of our study was to analyze Kickstarter projects launched in the US and identify the factors that contribute to their success. To prepare the data for analysis, we carried out several stages of data preparation.

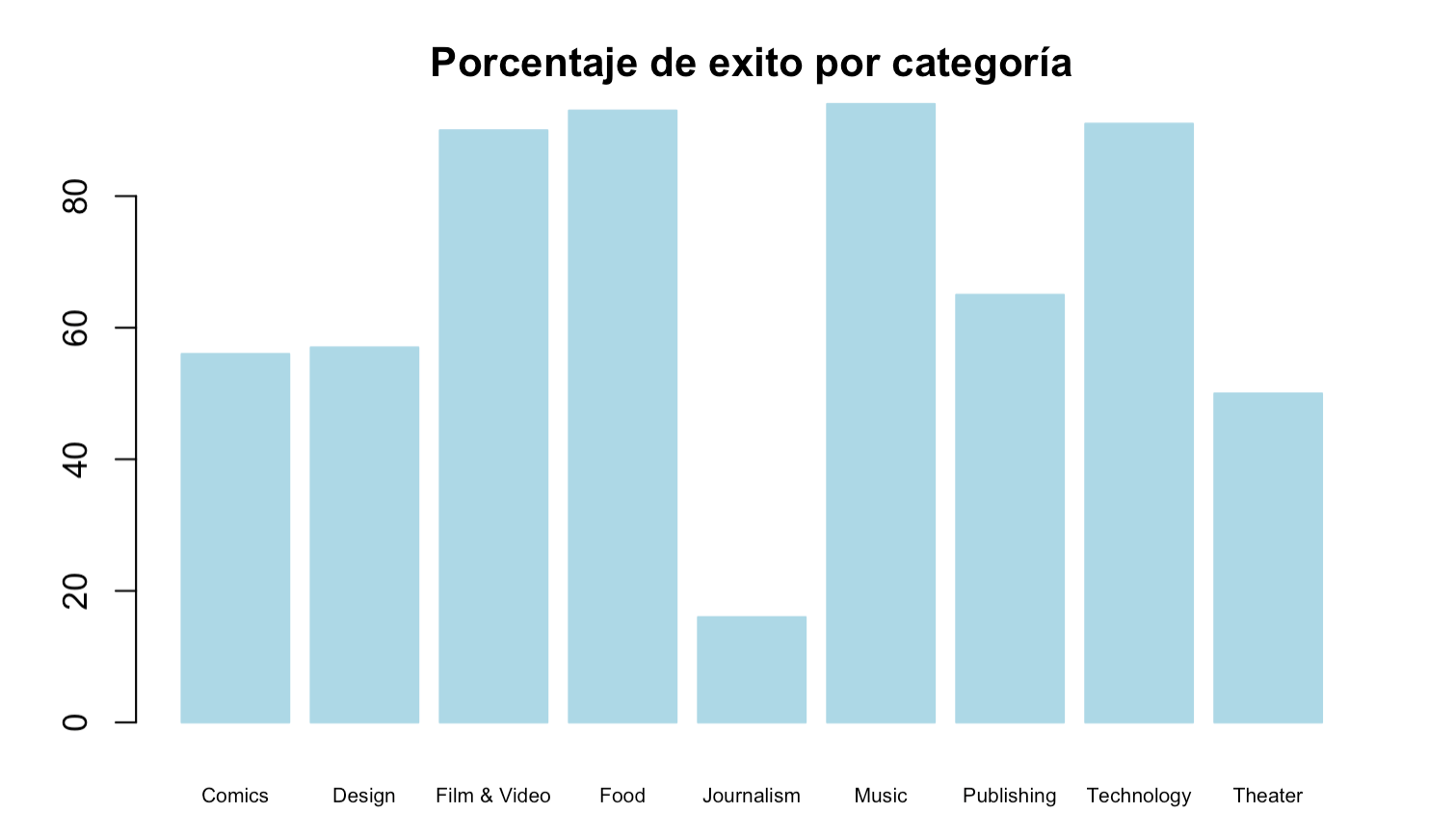
Firstly, we extracted the variables from the dataset in dictionary format and stored them in different variables. This allowed for greater clarity in the organization of variables and facilitated their subsequent manipulation. Additionally, we identified the most relevant variables for the study and removed those that did not contribute significant information. In M1, we provided a more comprehensive explanation of the process, outlining all the steps that were taken to clean and integrate the data for analysis.

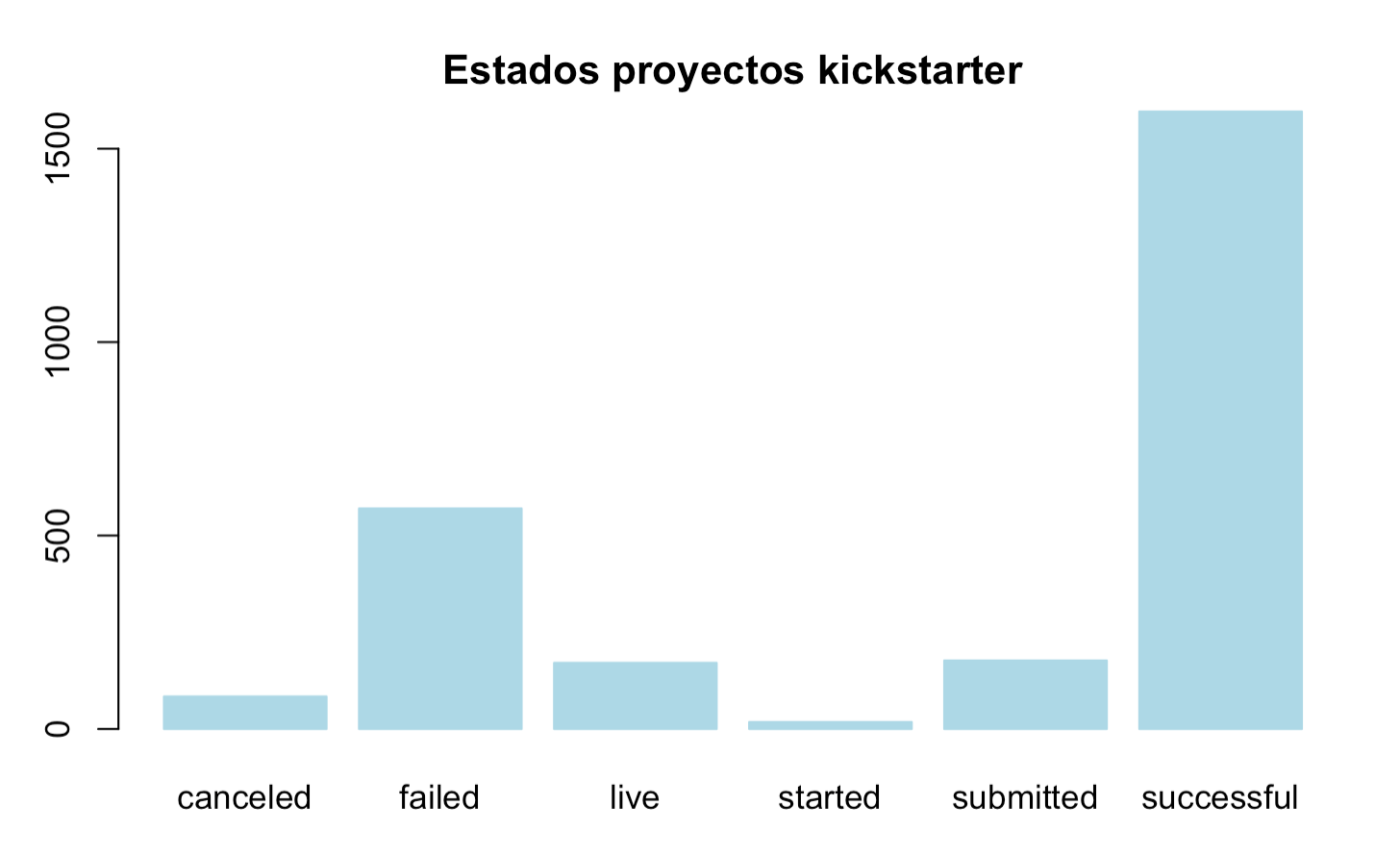
To ensure that our analysis was focused on relevant data, we filtered the database to include only Kickstarter projects launched in the US, as they comprised almost the total data. We also eliminated variables with missing data, as the number of missing individuals was small and would not significantly affect predictive models or web scraping.

To gain a deeper understanding of the data, we cross-referenced the BEA data with our database. Specifically, we selected the three sponsors that made the largest investment in each project and extracted information from the BEA based on their location and category. We obtained the value of investment per person made in a given category in a specific state, which will be used in the analysis later.

In addition to filtering and cross-referencing, we used data visualization tools to summarize and better understand the database. We used ggplot, plotly, and lubridate libraries to create visualizations that allowed us to observe the relationships between variables. Some examples of visualizations we created include scatter plots to show the relationship between variables, and histograms to summarize the distribution of variables.

Overall, the data preparation process allowed us to focus on relevant data, cross-reference with external data sources, and gain a better understanding of the relationships between variables. This will be valuable in our subsequent analysis of the factors that contribute to the success of Kickstarter projects launched in the US.





# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# Materialising minable view

The main task of this project is to know if a Kickstarter project will be successful or not. This entails knowing what characteristics make a project successful or not (understood as successful, a project that gets sponsored, in such a way that the amount of money raised reaches or even exceeds the figure that the creator of the project will have as a goal).

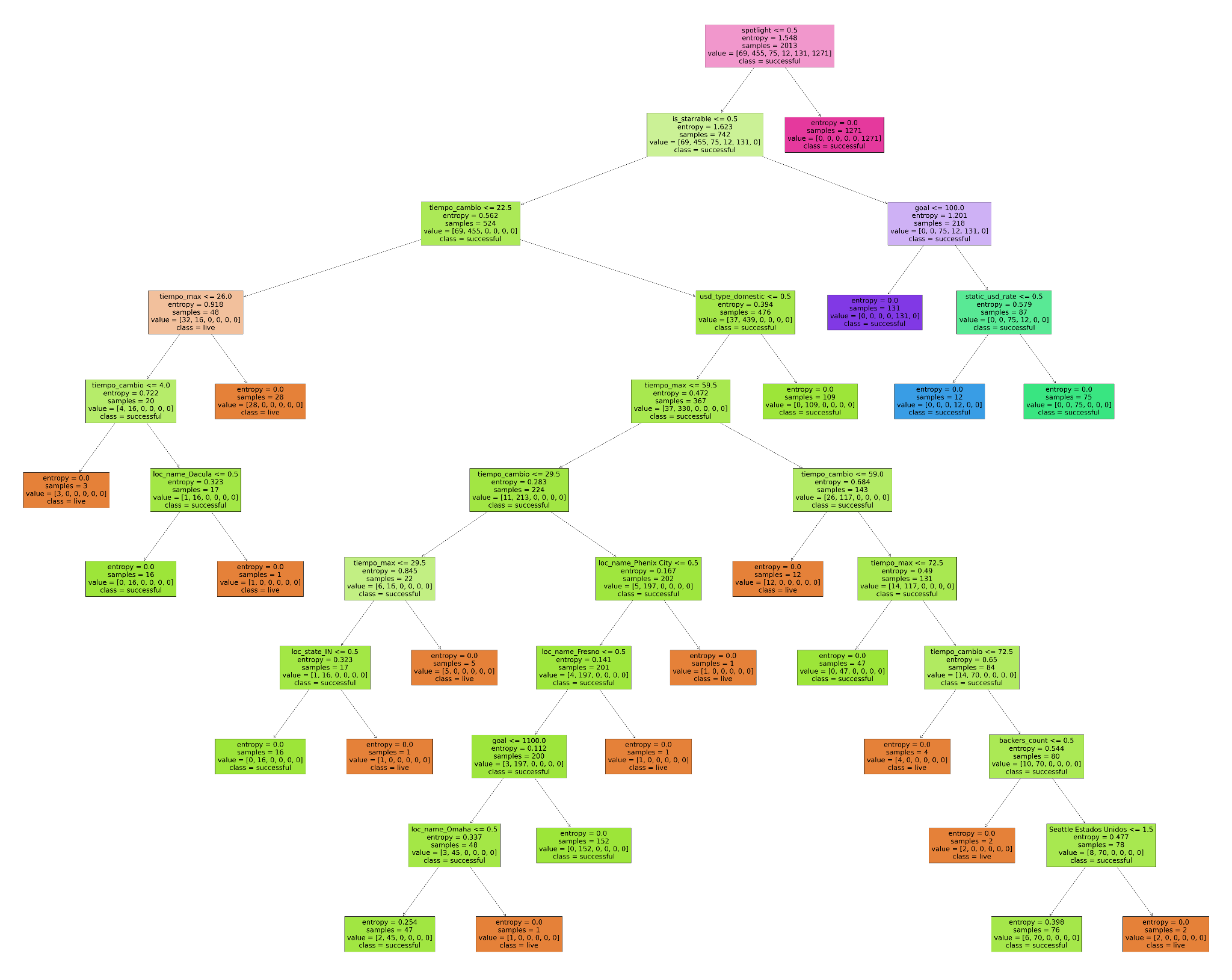
Based on the above information, the task to be performed is one of classification. The model that can be used to carry out this task is a classification tree. This choice is due to the fact that the classification trees are not parametric, since many of our data do not follow a normal distribution. In addition, they are easy to interpret and visualize, which allows a better understanding of how classification decisions are made and in this way identify the most important characteristics for classification, since it can also work with numerous variables, such as that of our base of data. Specifically, we will use the CART algorithm, for this we will use one-hot coding, where a binary variable is created for each possible value of the categorical variable.

As we have mentioned before, the objective is to know the final state of the project on Kickstarter, the corresponding variable is ‘*state*’ and, therefore, the output of the model. While the input features are all the variables that have been collected in the Kickstarter dataset. These collect demographics of the project creator and backers, project details (category ‘*main\_category*’, subcategory, whether the project has been favorited by the Kickstarter staff *'staff\_pick'*, whether the project has been featured on the Kickstarter homepage *'spotlight'*, etc.), funding details (amount of money for the realization of the project 'goal', number of backers *'backers\_count'*, campaign duration, etc.).

**Model building**

The selected model is Classification and Regression Trees (CART). The reason for choosing this model is that it is simple and therefore interpretable, making it a good option for a first approach to understanding our data. In order to fit the model, the database has been transformed, and many of the variables that were previously categorical have been converted to binary. Additionally, it was observed that some variables had perfect correlation (pledged, usd\_pledged, converted\_pledged\_amount), so some were deleted as they provided the same information and could cause multicollinearity issues. Ultimately, the database has a total of 857 variables.

This database will be used to feed the model, which predicts the variable 'state' based on these 857 features. For the selection of hyperparameters, results are evaluated using different measures of impurity (gini and entropy) and different levels of tree depth. It should be noted that the best performance of the model is obtained using entropy as the measure of impurity and a maximum tree depth of 10 layers.

 Illustration 2. CART algorithm

# Evaluation

To create the decision tree, we first instantiate a decision tree classifier object with a random state of 42. Next, we define a set of parameters to explore various combinations of hyperparameters, such as the maximum depth of the decision tree (max\_depth) and the criterion, which is a function that measures the quality of a split. We then create a grid search object (GridSearchCV) that uses the decision tree classifier and the parameter set to identify the combination of hyperparameters that achieves the highest accuracy.

Finally, we create a new decision tree classifier object using the best hyperparameters that were found during the grid search (max\_depth=10, criterion='entropy', random\_state=42). The entropy criterion measures the impurity based on the information gain.

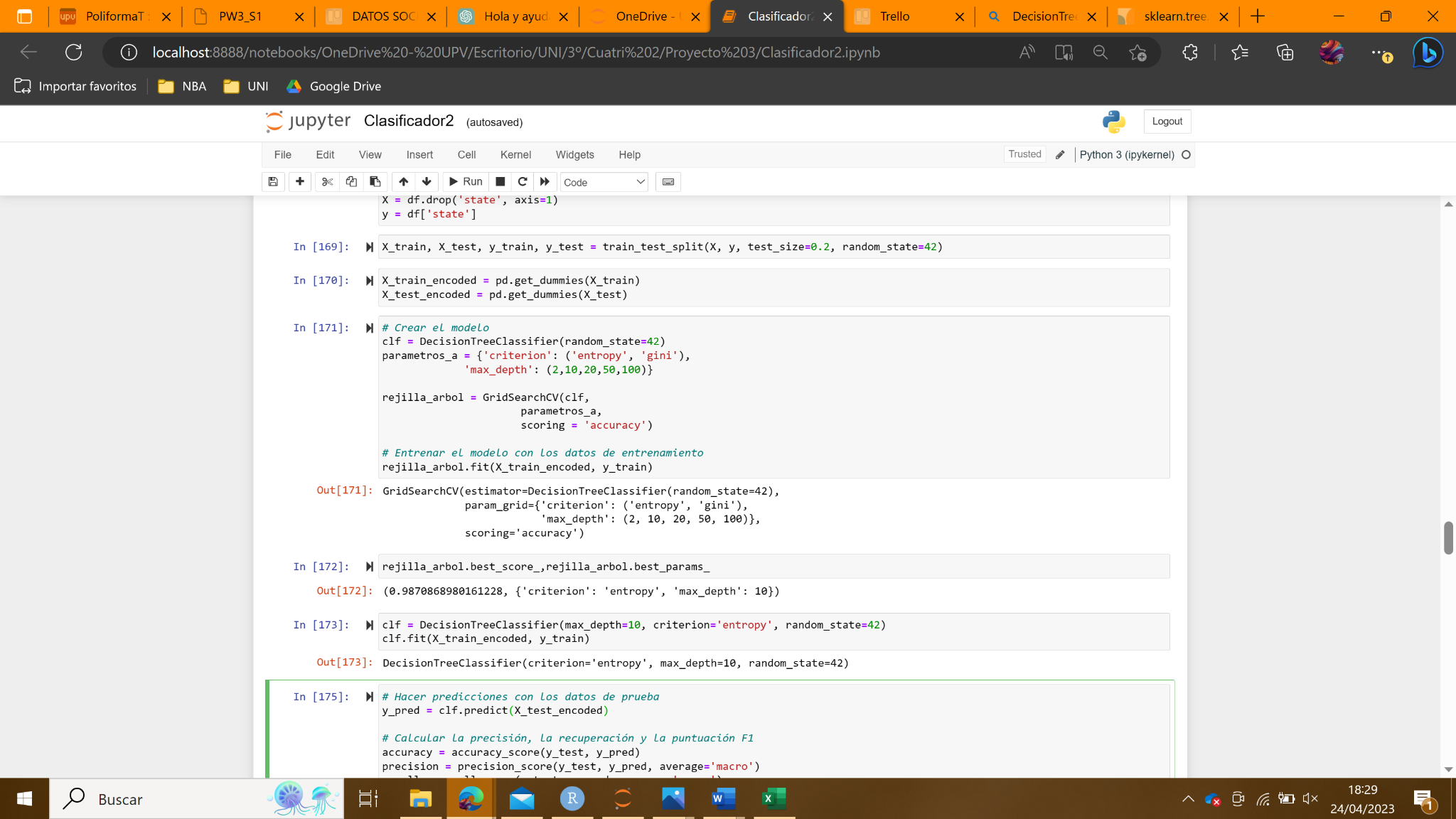
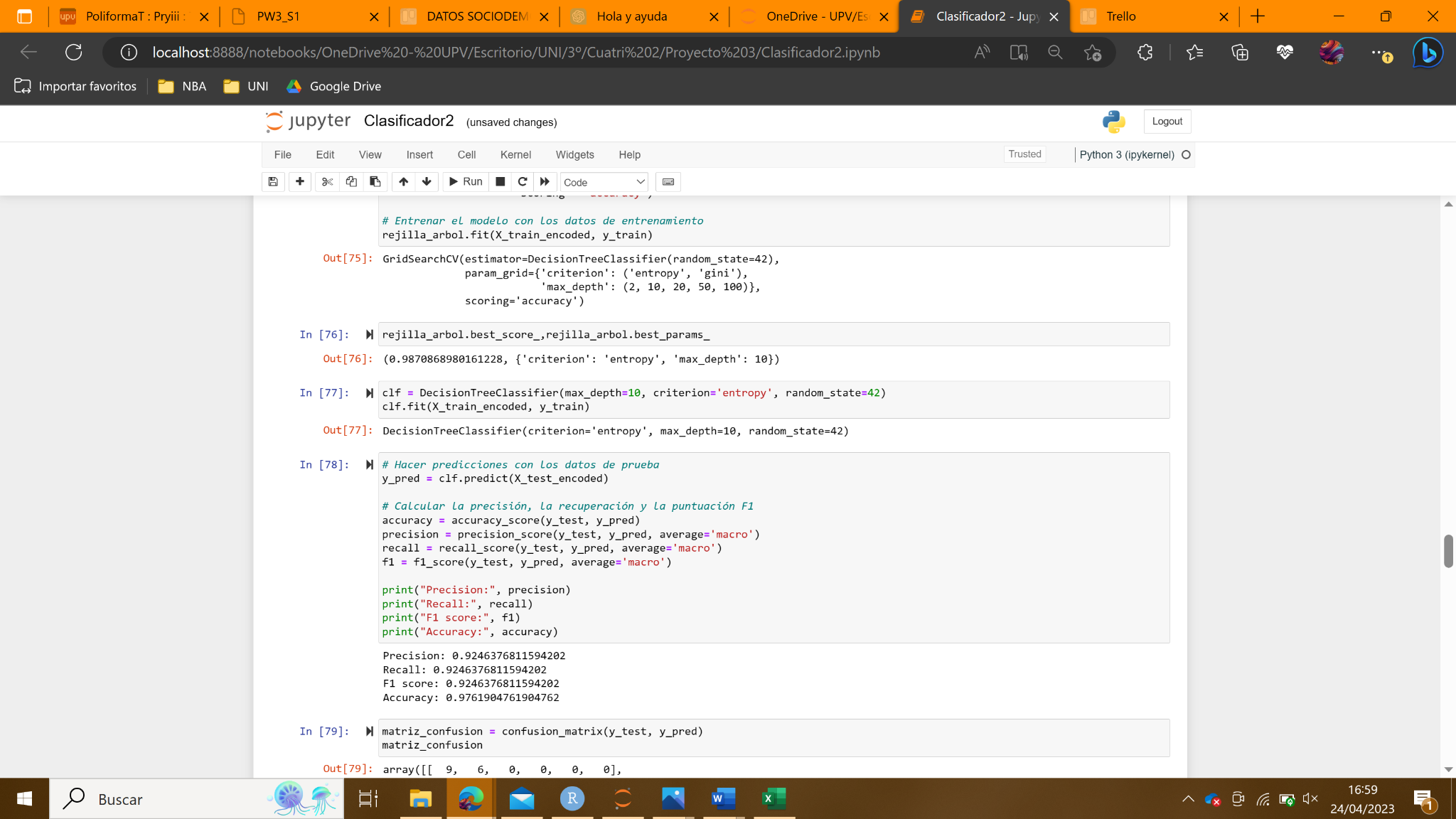


Illustration 3. Model evaluation

After constructing the model, it is crucial to evaluate its effectiveness in the classification task. Evaluating a predictive model helps to understand its ability to accurately classify samples that were not used in the training process. Various metrics, such as precision, recall, F1 score, and accuracy, are used to quantify the model's performance in the classification task. The evaluation of our model yielded the following results:



Based on the evaluation metrics obtained for the multiclass classification model, it appears that the model is performing strongly. The precision score, which measures the model's ability to accurately predict a class, is 0.9246. This indicates that the model has a high level of accuracy and a low chance of false positives, allowing it to correctly identify the class in the majority of cases.

Based on the obtained evaluation metrics for the multiclass classification model, it appears that the model is performing well. The precision score of 0.9246 indicates that the model has a high level of accuracy in predicting a class, with a low chance of false positives. Additionally, the recall score of 0.9246 suggests that the model can correctly identify a high proportion of samples belonging to a specific class, with a low false negative rate.

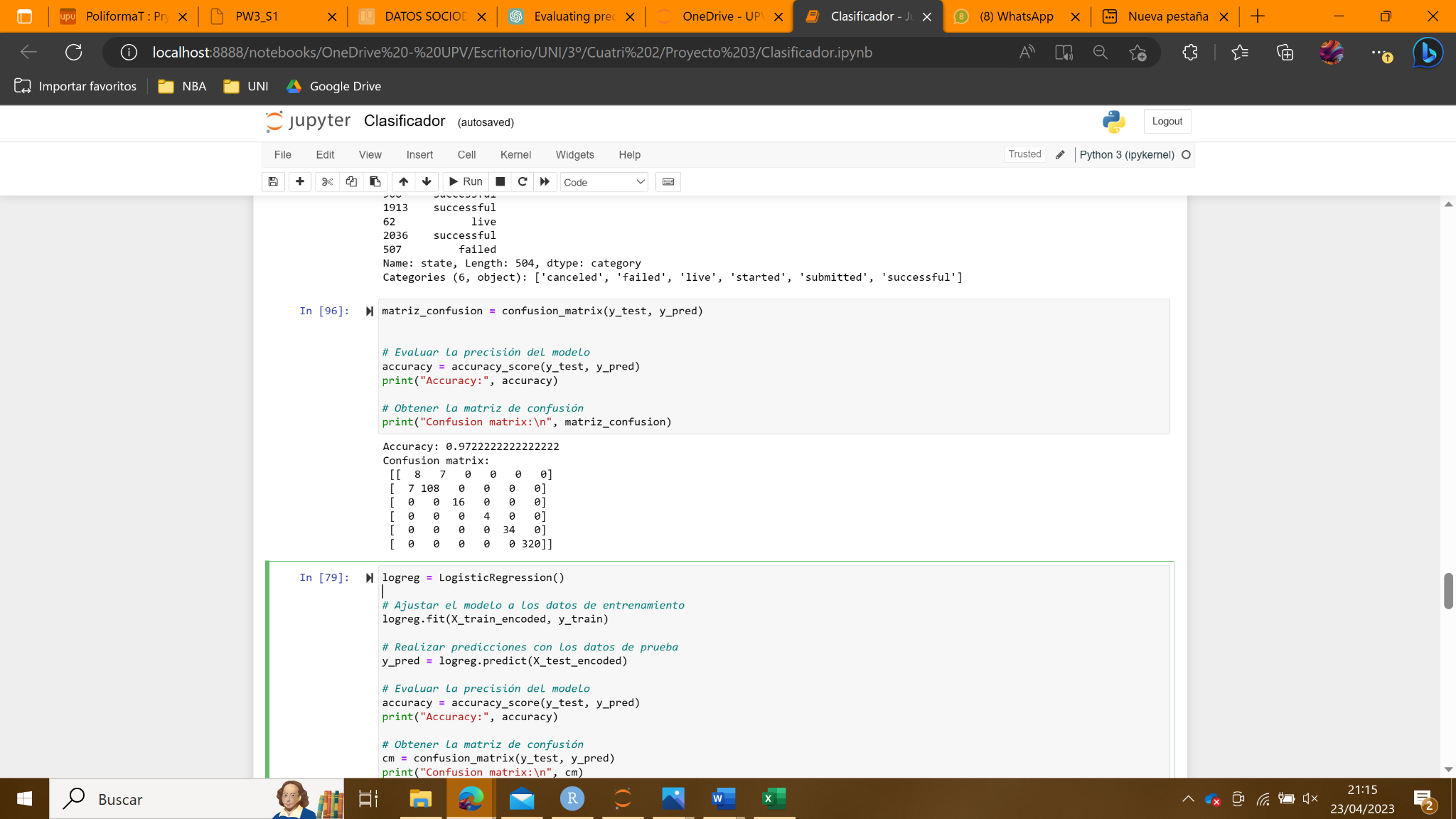
Moreover, the F1 score of 0.9246 indicates that the model has a good balance between precision and recall, which is important when both metrics are equally crucial. These evaluation metrics were calculated individually for each class and then averaged, which helped identify any imbalances in the dataset. The results indicate that the model performs well across all classes.

The high accuracy score of 0.978 suggests that the model can correctly predict the majority of samples in the dataset, with only a small percentage of misclassifications. This indicates that the model is robust and can generalize well to new data.

It is important to note that after the initial evaluation, other metrics should be used to assess the model's performance. One commonly used metric is the confusion matrix, which shows the number of correctly and incorrectly classified samples for each class.

For our multiclass classification model, the confusion matrix displays the results for the six classes: Canceled, Failed, Live, Started, Submitted, and Successful.

:

[ 9 6 0 0 0 0 ] ← Canceled

[ 6 109 0 0 0 0 ] ← Failed

[ 0 0 16 0 0 0 ] ← Live

[ 0 0 0 4 0 0 ] ← Started

[ 0 0 0 0 34 0 ] ← Submitted

[ 0 0 0 0 0 320 ] ← Successful

The main diagonal of the matrix represents correct predictions, while the other entries of the matrix represent incorrect predictions. In this case, each row and column of the matrix represents a different class.

The first row represents the "Canceled" class, and we can see that the model has correctly predicted that 9 samples belong to this class and has incorrectly classified 6 samples of this class into the second class "Failed". The second row represents the "Failed" class, and the model has correctly predicted that 109 samples belong to this class and has incorrectly classified 6 samples of this class into the first class "Canceled".

The other rows represent the "Live", "Started", "Submitted" and "Successful" classes, and we can see that the model has correctly predicted all samples in these classes.

These results suggest that the model performs well in predicting the class of new data instances, with high accuracy in most of the classes. However, we can notice the misclassification of samples in the “Canceled” class. It is not surprising that the model struggled to predict this class accurately, as there could be several reasons why a startup may be canceled that are not captured in the available data.

The majority class is "Successful” with 320 samples, and all of them were correctly classified. This is important because correctly classifying the “Successful” class is key to achieving the objective of determining whether a startup will be successful or not. Therefore, the fact that all samples in this class were correctly classified is a positive indication of the model's ability to achieve this goal.

In conclusion, the evaluation results suggest that the model is highly accurate and performs well in classifying samples into their respective classes.

# Deployment mockup

# 

# 

# Use of technology

As data scientists, we have found ourselves in the unique position to leverage cutting-edge technology in order to tackle complex problems. Through our extensive experience and training, we have developed a deep understanding of technical tools and packages, enabling us to work with ease and precision.

Throughout the data science degree program at UPV, we have been given the opportunity to develop a broad range of technical skills, which we have eagerly put into practice. For the programming languages, we have mostly used Python and R. Here are some packages and tools the team has decided to highlight due to their importance in the project realization:

1. Scikit-learn
2. ROPLS
3. ggplot2, lubridate , plotly (simple but understandable and professional visualizations)

In addition, we have demonstrated our resourcefulness and willingness to learn by utilizing online technical forums such as Stack Overflow, GitHub and ChatGPT to troubleshoot technical issues and find related information. These forums and websites have been invaluable in helping us tackle challenging problems that have appeared in the process.

Moreover, we have demonstrated autonomy and resourcefulness in our technical work by actively seeking out external experts for help when necessary. For example, we found ourselves in a situation where we could not extract Kickstarter webpage information due to web scraping restrictions in common Python methods. We tried contacting the Kickstarter company for assistance with web scraping tasks, but found no helpful support. The metrics we were looking for were not available in the API documentation (*https://status.kickstarter.com/api*) or StatsPage (*https://www.kickstarter.com/help/stats*) either, so we were forced to find an alternative web scraping method. Finally, we reached out to Cèsar Ferri, our model deployment teacher, to find information about the automation tool *UIPath* he had taught us in class. With Cèsar’s assistance, we were able to use this tool to access and automate the web scraping process, and extract the data required for the project.

Overall, our training in data science has equipped us with a proficient use of technology and problem-solving skills necessary to deploy a reliable and accurate predictive model for Kickstarter project success. We are confident that our innovative approach, coupled with our resourcefulness and determination, will enable us to achieve our project objectives and deliver meaningful insights. The team has not only mastered several technical tools and packages, but also shown autonomy by finding solutions in technical forums and seeking external help when needed.

# Seminar use