# Telecom Churn Predictor

Churn Detection Model for a Telecommunications Company

// Technical Presentation //

# The goal

The main objective of this project is to develop a predictive model to identify the probability of churn or customer attrition rate in a telecommunications company.

This model will enable the company to implement proactive strategies to retain customers at risk of churn, thereby optimizing customer relationship management and improving profitability.

| Churn | |
|---|---|
| 0 | 5163 |
| 1 | 1869 |

# Data

The starting point is a Kaggle dataset containing data from an American internet service company.
The dataset was in fairly good condition, with very few NaNs or empty fields. However, there was a slight imbalance in the target classes.

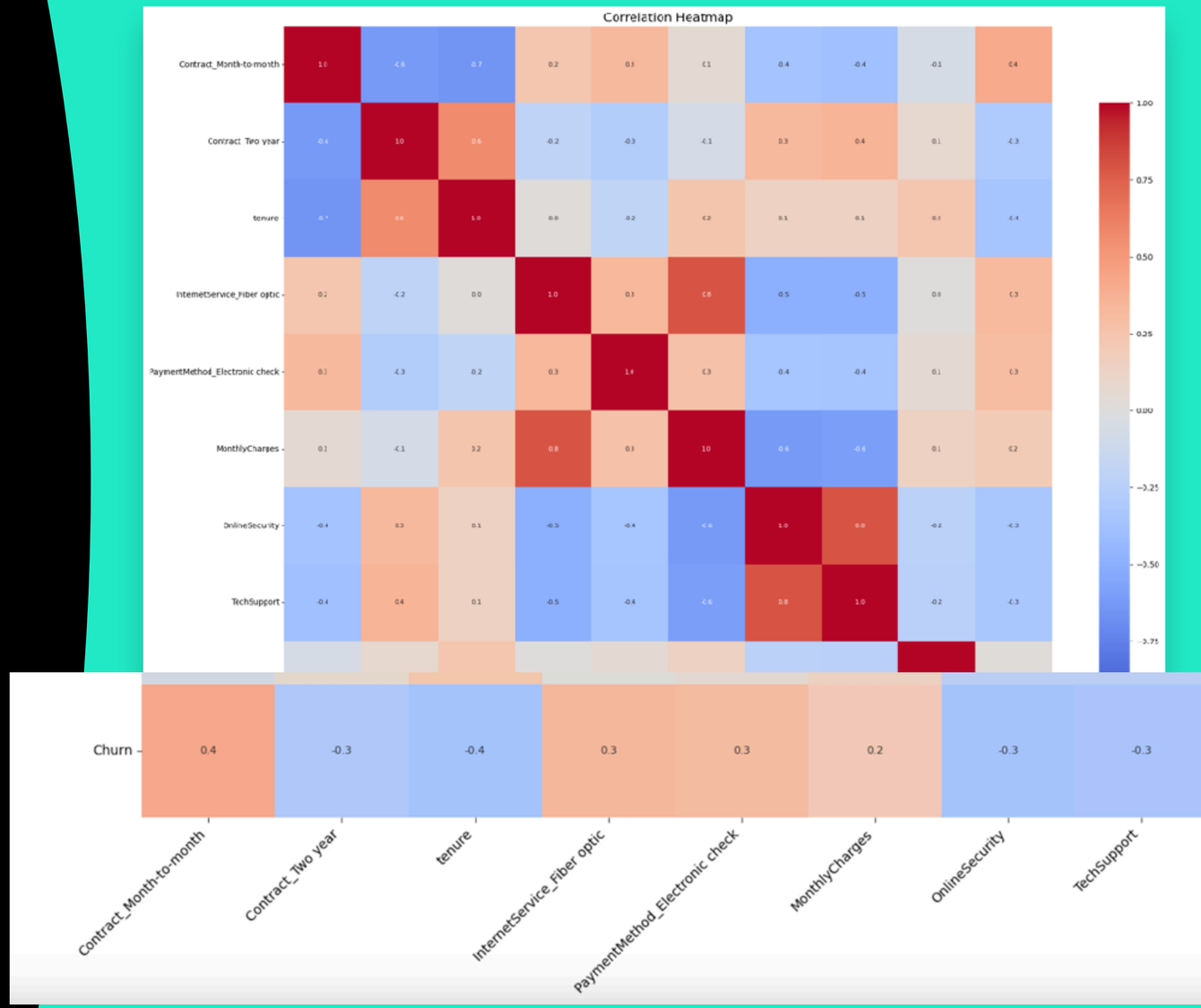
With 21 original variables, after converting categorical and binary variables to numerical format, the total number of variables expanded to 27.

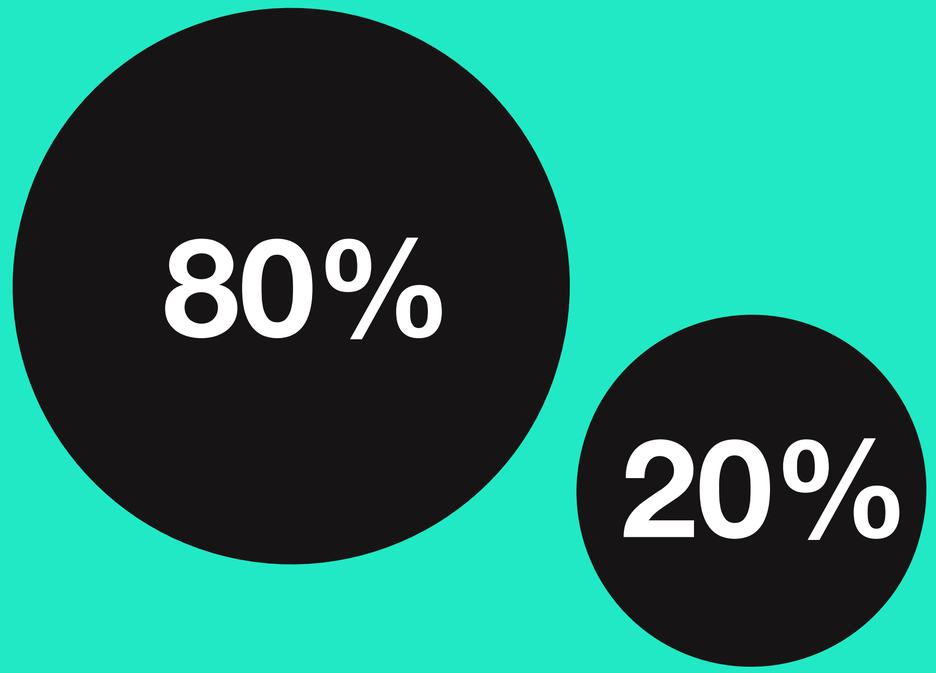
# Analyse and feature engineering

Using a correlation map to analyse the relationships between the different variables, I decided which variables would be included in the model testing.

Additionally, after detecting a possible risk of multicollinearity among some variables such as 'tenure,' 'TotalCharges,' and 'MonthlyCharges,' I decided to eliminate 'TotalCharges' due to its high correlation with the others.

I conducted tests with several different sets of variables and ultimately selected the set that showed the best performance.






# Data Split for train and test

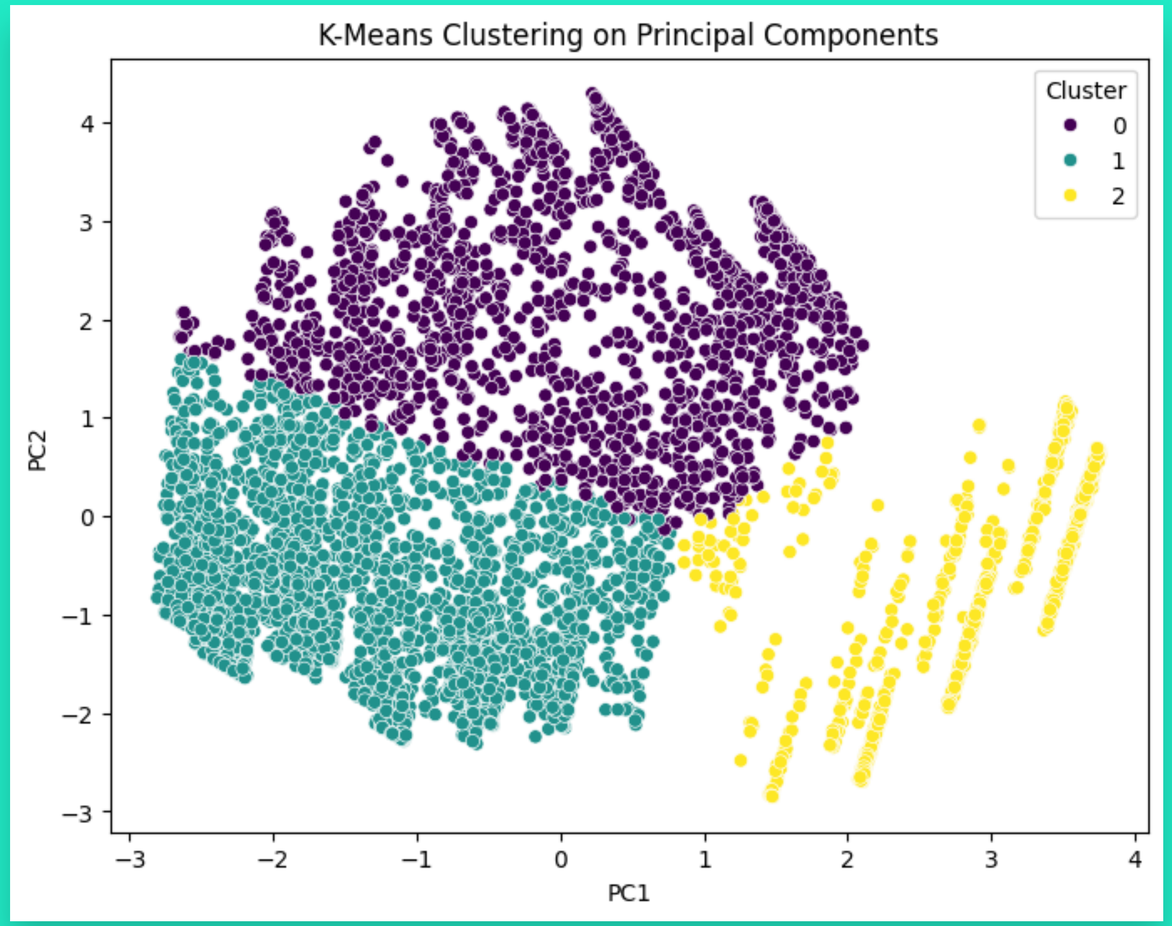
For the development and validation of the model, I split the data into a training set and a test set as usual.

This way, I ensure that the model is trained and evaluated in a manner that allows for measuring its generalisation ability, avoiding overfitting issues, and providing more accurate performance metrics.

# Unsupervised model, K-Means

During the development of this project, I considered using an unsupervised model like K-Means to identify patterns and segment customers into different groups.

However, after evaluating the results and the usefulness of the clusters for direct churn prediction, I saw that K-Means did not provide the necessary predictive capability, at least not without dedicating significantly more time, which I did not have.



# Tests with supervised models

Due to the need for clear and usable predictions, I decided to focus on supervised models.

These models allowed for a direct evaluation of churn using specific metrics that measured the probabilities of churn for each customer, focusing on achieving the best possible recall to maximise the model's ability to identify positive cases.

Additionally, I used GridSearch to tune the hyperparameters in some of these models to make them more efficient.

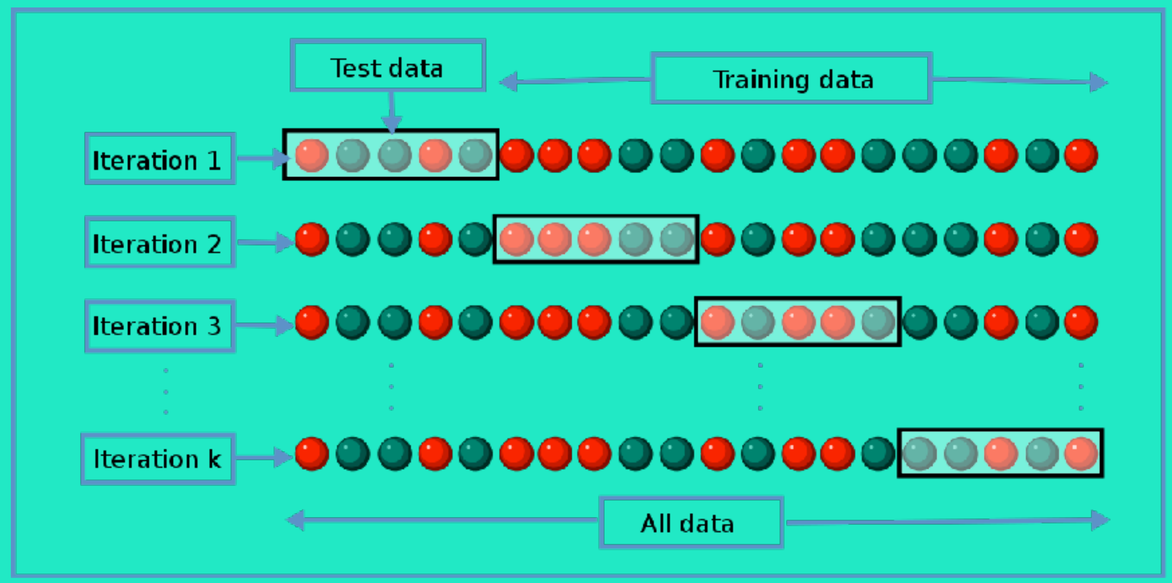
Furthermore, I used SMOTE for class balancing and StandardScaler for feature normalization.

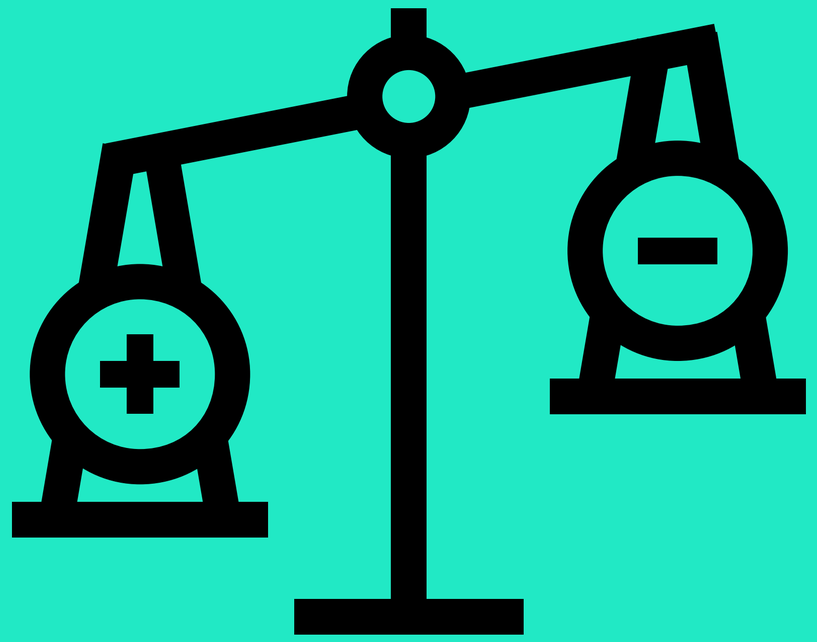
# Cross-Validation

To ensure that the model's results were robust and generalizable, I used cross-validation to evaluate its ability to control overfitting.

Instead of splitting the data into a single training set and a single test set, cross-validation allows for dividing the data into multiple subsets (folds).

This way, the model is trained and evaluated multiple times, using different combinations of these subsets. This approach helps to avoid overfitting, ensuring that the model generalises well to new data.





# The Importance of Detecting Positives

There are sectors such as healthcare, where failing to detect a positive case can have serious consequences.

Or security in fraud and intrusion detection systems, where it is crucial to detect all possible threats.

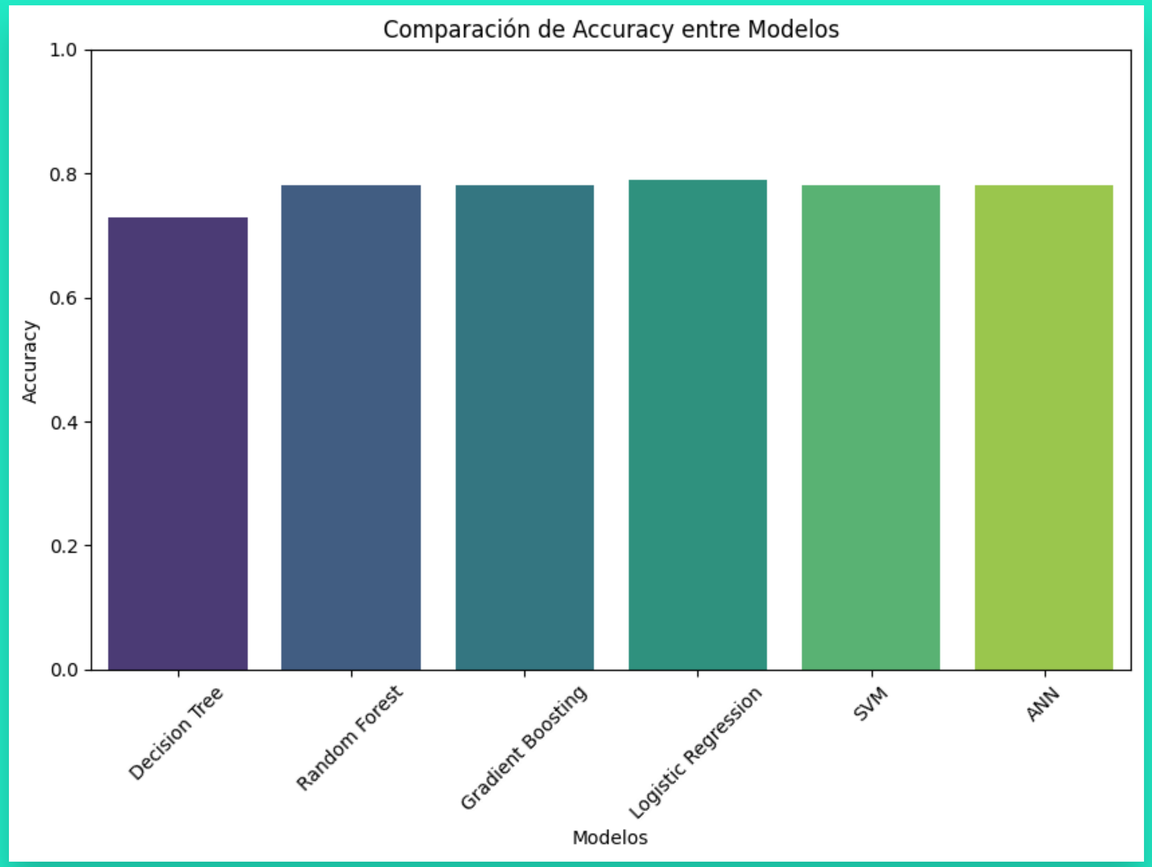
However, there is also marketing, specifically customer retention (churn) campaigns, where it is crucial to identify all customers who may leave the service to intervene.

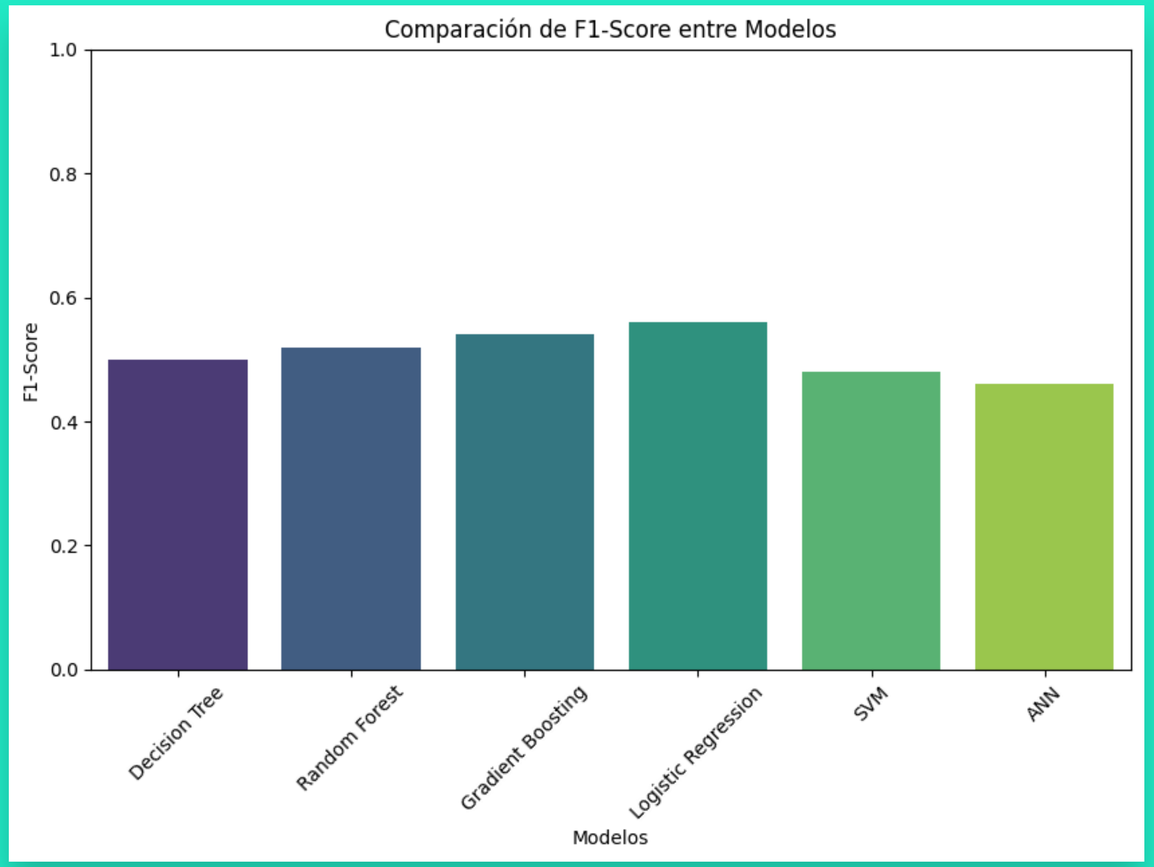
This is the reason why I decided to focus on Recall.

# Accuracy

Although the most important metric in this case is Recall, Accuracy in logistic regression shows a precision similar to the other models, close to 0.80.

This means that, despite being a linear model, logistic regression is performing well on this problem.






# F1-Score

Logistic regression has the highest F1-Score among the evaluated models, close to 0.60.

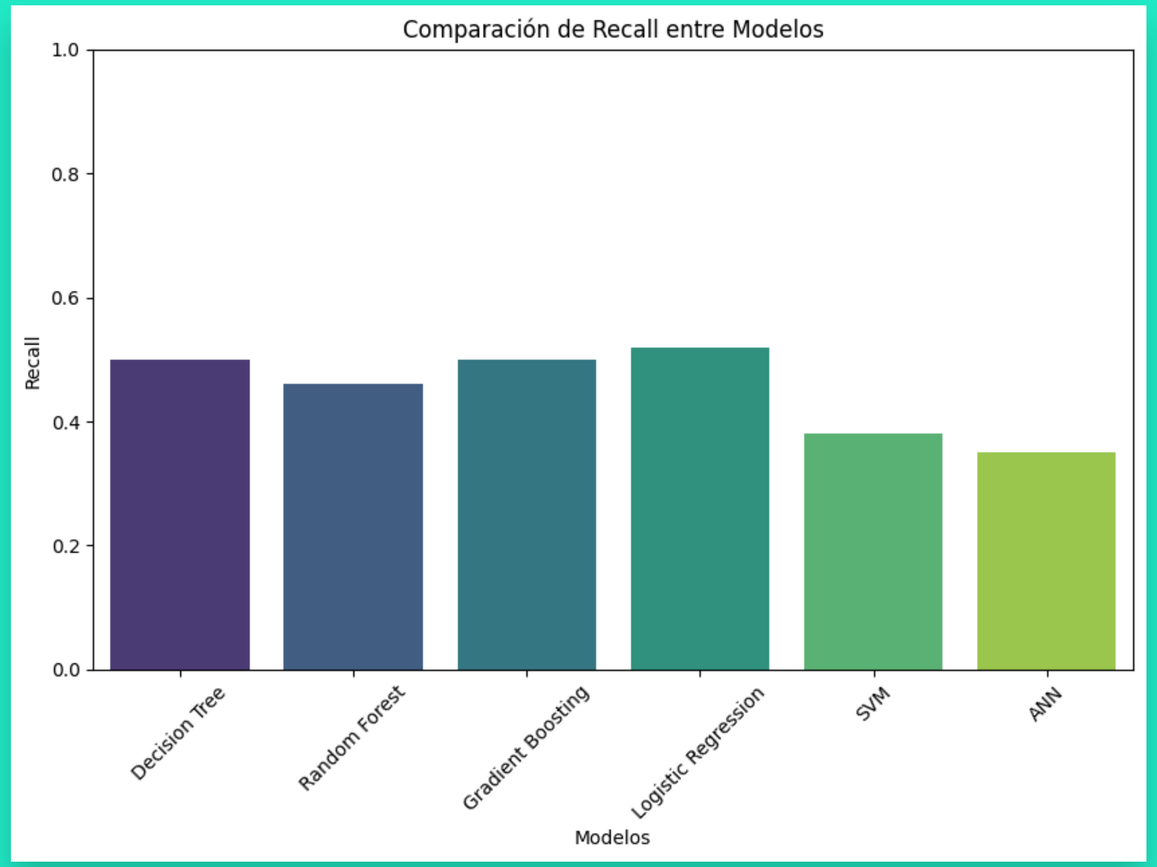
This suggests that this model also handles the balance between precision and recall well for this problem.

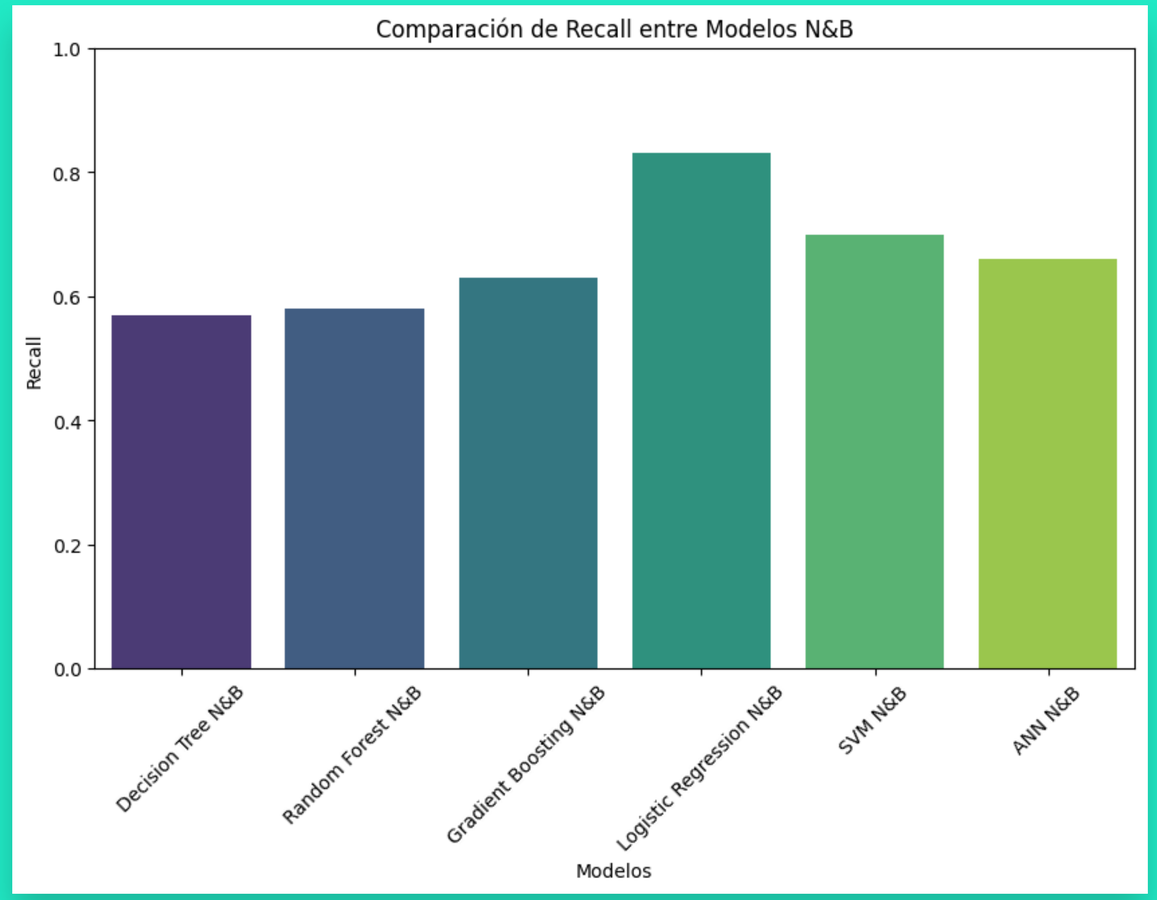
# Recall before balancing, normalisation, and hyperparameter tuning

In this graph, Recall is compared across different models before applying balancing, normalisation, and hyperparameter tuning.

It can be observed that the Logistic Regression model shows slightly better performance compared to other models such as Decision Tree, Random Forest, Gradient Boosting, SVM, and ANN.

Although the difference is not extremely marked, it is clear that logistic regression is already starting to stand out in its ability to correctly identify positive cases.






# Recall after balancing, normalisation, and hyperparameter tuning

In this second graph, after applying balancing techniques, normalisation, and hyperparameter tuning, we can see that the Logistic Regression model not only continues to be the best-performing one but also significantly increases its advantage.

The performance difference between the Logistic Regression model and the other models becomes more notable, highlighting the effectiveness of the applied techniques in improving its ability to detect customers at risk of churn.

# Confusion Matrix: Final Model Evaluation

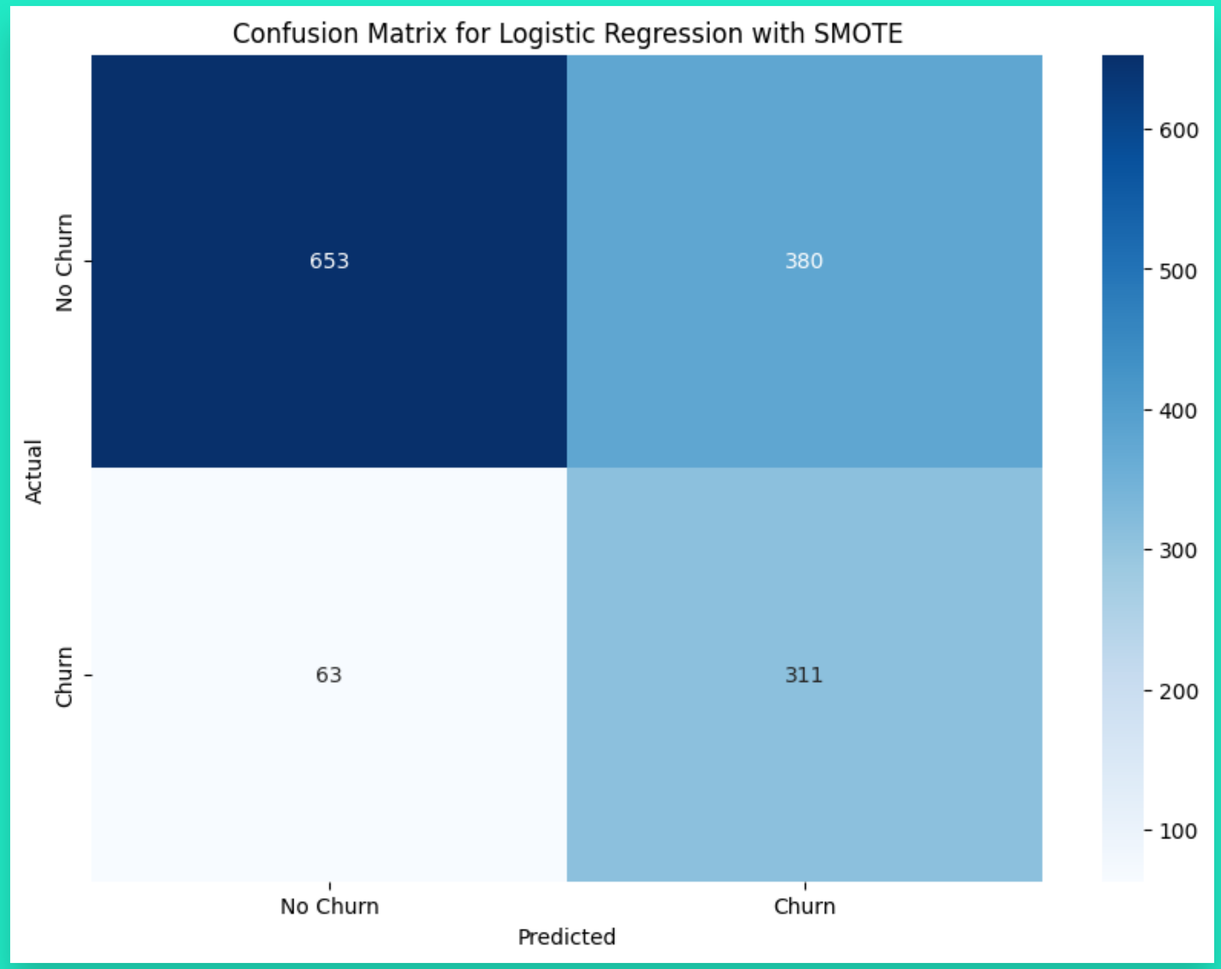
With this model, the recall is 0.83, which means that the majority of customers who will churn are correctly identified.

True Negatives (TN = 653): Customers that the model correctly identified as not likely to churn.

False Positives (FP = 380): Customers that the model predicted would churn, but actually did not.

False Negatives (FN = 63): Customers that the model predicted would not churn, but actually did.

True Positives (TP = 311): Customers that the model correctly identified as likely to churn.



# Conclusion, next steps

The recall of 0.83 reflects a high level of success in identifying customers at risk of churn, which is crucial for implementing effective retention strategies.

However, it is essential to continue optimising the model, monitoring its performance in production once deployed, and refining customer retention strategies with personalised actions based on the predictions.

# Thanks for watching!

**GitHub:** https://github.com/nacjacds/Telecom_Churn_Predictor

**Web:** https://nachojacquot.com/en/.

**LinkedIn:** https://www.linkedin.com/in/jacquot/

**Test the model:**
https://telechurnpredictor.streamlit.app/