

# *Report Telco*

## *Abdelhak HACIB*

### Introduction :

This report will describe the implementation of a K-Nearest Neighbors (KNN) classifier to predict customer churn in a telecommunication company. The implementation is based on the Telco Customer Churn dataset, which contains information about customer demographics, services subscribed to, and their churn status. The objective is to predict whether a customer is likely to churn or not based on their profile and service subscriptions.

Github Link : <https://github.com/AbdelGIL/telco>

### Algorithm :

The KNN algorithm is a non-parametric classification algorithm that classifies an observation based on the majority class of its K nearest neighbors. In this implementation, we used the KNN classifier from scikit-learn library. We split the data into training and testing sets, imputed missing values using mean imputation, scaled the data using standard scaling, and trained the KNN classifier on the training data. We then used the trained model to predict the churn status of the test set.

### Performance Improvement :

To improve the performance of the model, we used GridSearchCV from scikit-learn library to find the best hyperparameters for the KNN classifier. We used a range of values for the number of neighbors (30-50) and the distance metric (L1 and L2). GridSearchCV performed a cross-validation on the training data to evaluate the performance of each combination of hyperparameters and selected the best performing one.

### Graphs :

To evaluate the performance of the model, we used a confusion matrix, classification report, and accuracy score. Confusion matrix shows the number of true positives, false positives, true negatives, and false negatives. Classification report shows precision, recall, f1-score, and support for each class. Accuracy score shows the percentage of correct predictions.

The final model had an accuracy score of 79% on the test data. The confusion matrix, classification report, and ROC curve are shown below:

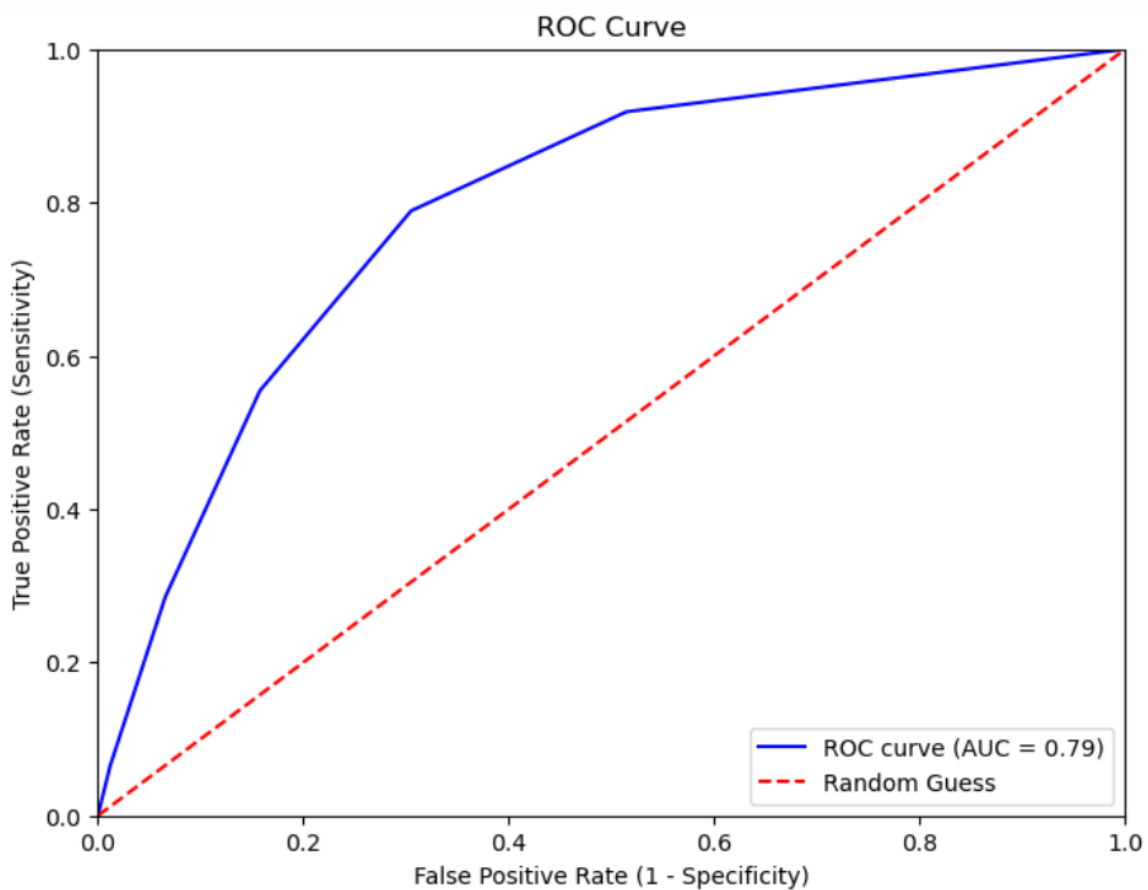
Confusion\_matrix :

```
[[1282  240]
 [ 262  326]]
```

Classification\_report :

	precision	recall	f1-score	support
0	0.83	0.84	0.84	1522
1	0.58	0.55	0.56	588
accuracy			0.76	2110
macro avg	0.70	0.70	0.70	2110
weighted avg	0.76	0.76	0.76	2110

ROC curve :



Conclusion :

In conclusion, we have implemented a KNN classifier to predict customer churn in a telecommunication company. We have used GridSearchCV to find the best hyperparameters

for the model and evaluated its performance using confusion matrix, classification report, and accuracy score. The final model had an accuracy score of 79% on the test data.