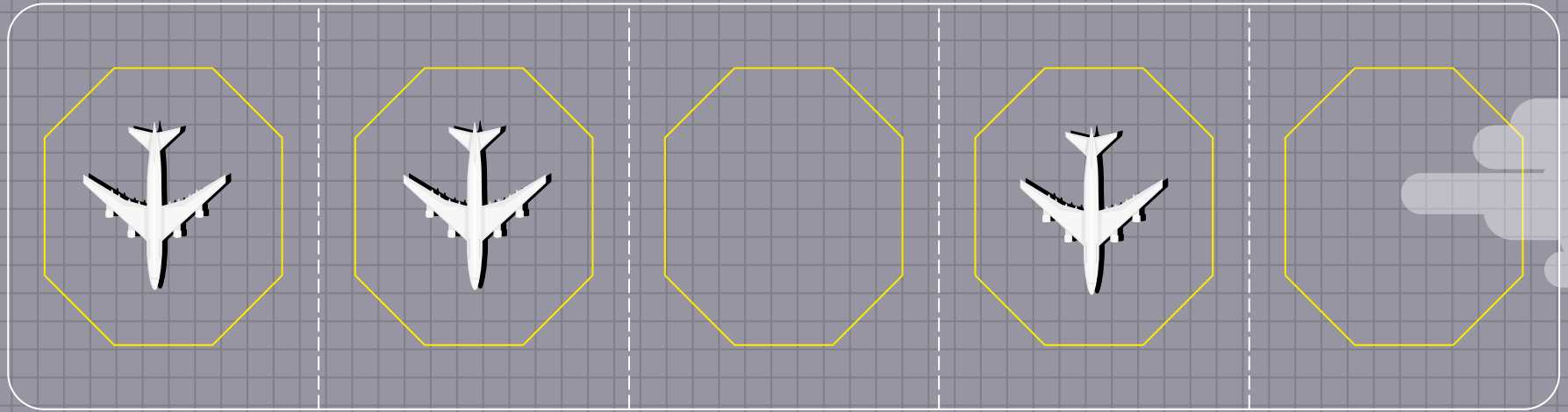


Airline Passenger Satisfaction

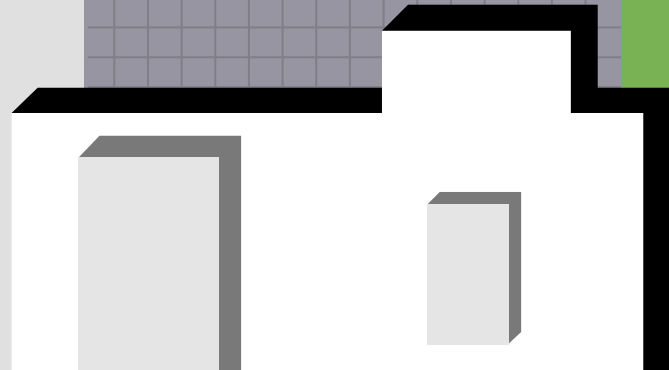
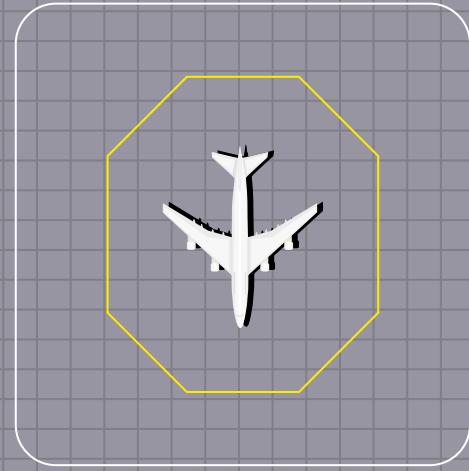
Big Data Computing, a.y. 2021/22, Michele Spina



01

A/D

What are the starting points
and goals of this project?





2.987.240



More than 100.000 rows and 23 columns



What do we know?



Customer information

Age and gender of the passenger and reason of the travel



Flight data

Flight distance, class and delay



Satisfaction levels by category

Satisfaction level (0 to 5) of flight services, like food or cleanliness





Arrival point

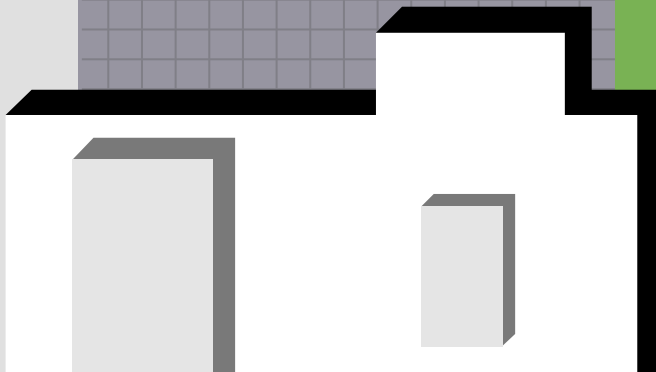
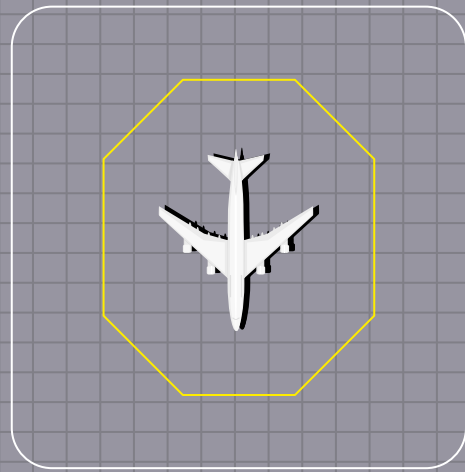
The goal of the project is to predict if a passenger it's **satisfied or not**



02

Pre- processing

Analyze the distribution and correlation of the data, and apply feature engineering



Pre-processing

A

Analyze dataset

Analyze data distribution and correlation

B

Drop columns

Drop useless or redundant columns

Data encoding

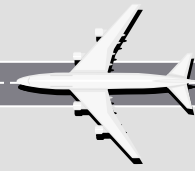
Convert categorical values to numerical values

C

Normalize

Normalize the data distribution

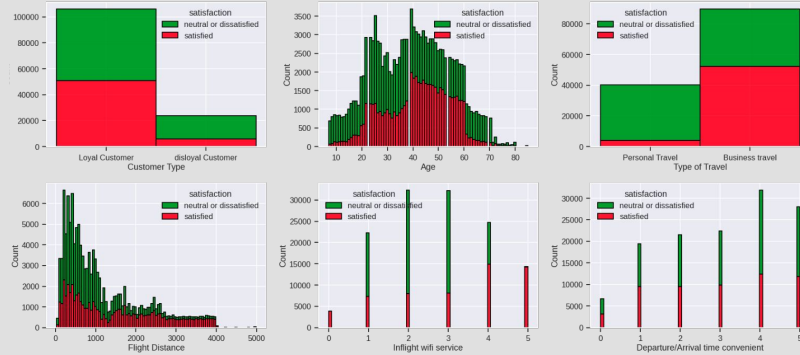
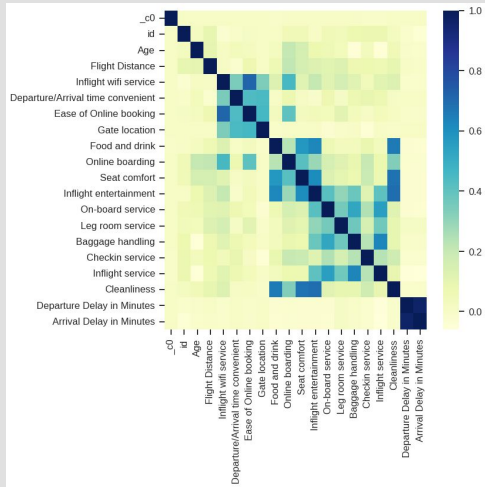
D



Analyze dataset

Data distribution

Data are not normally distributed



Correlation matrix

High correlation
arrival-departure delay

Data encoding pipeline



StringIndexer

it encodes string column of labels to a column of label indices

OneHotEncoder

it maps a categorical feature to a binary vector indicating the presence of a specific feature value

VectorAssembler

It combines a list of columns into a single vector column;

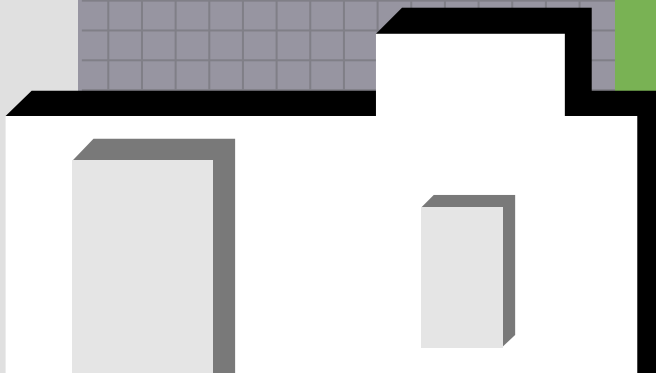
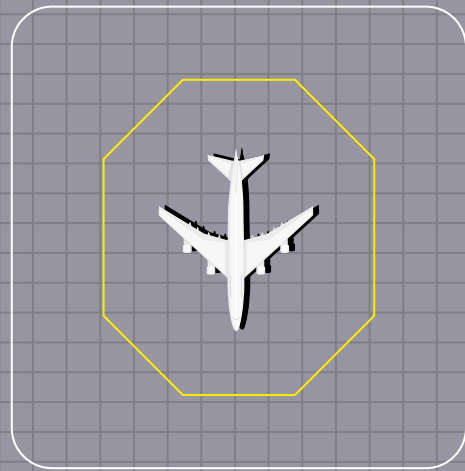
StandardScaler

it normalizes each feature to have unit standard deviation and/or zero mean;

03

Classification models

Logistic Regression, Random
Forest and Decision Tree



Classification **models**

**Logistic
Regression**

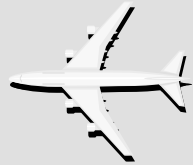
LR

**Random
Forest**

RF

**Decision
Tree**

DT



LR results

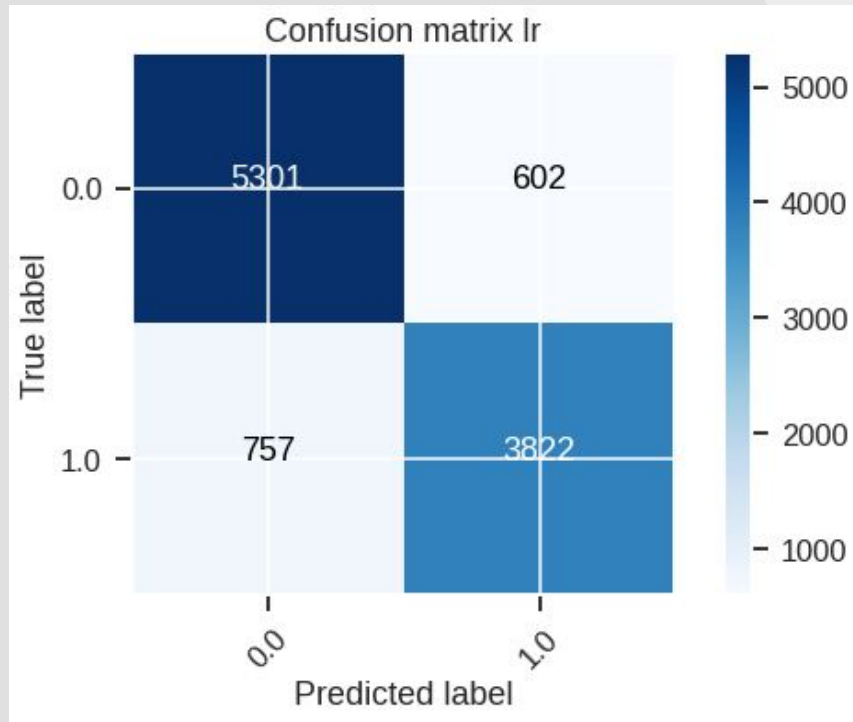
Precision metric: 0.869

Recall Metric: 0.866

Accuracy Metric: 0.870

F1-score Metric: 0.868

AUROC: 0.925



RF results

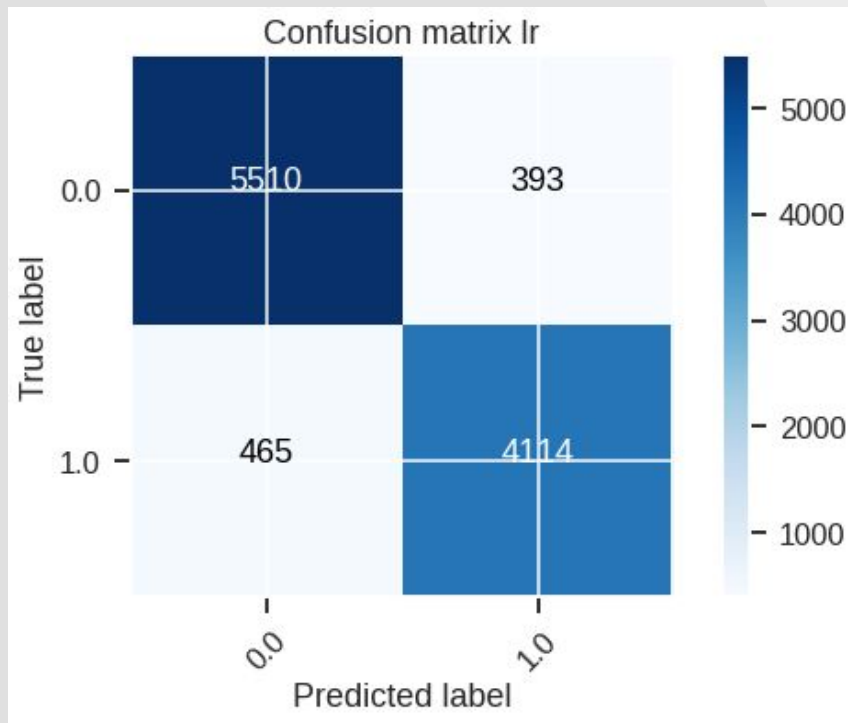
Precision metric: 0.917

Recall Metric: 0.916

Accuracy Metric: 0.918

F1-score Metric: 0.917

AUROC: 0.972



DT results

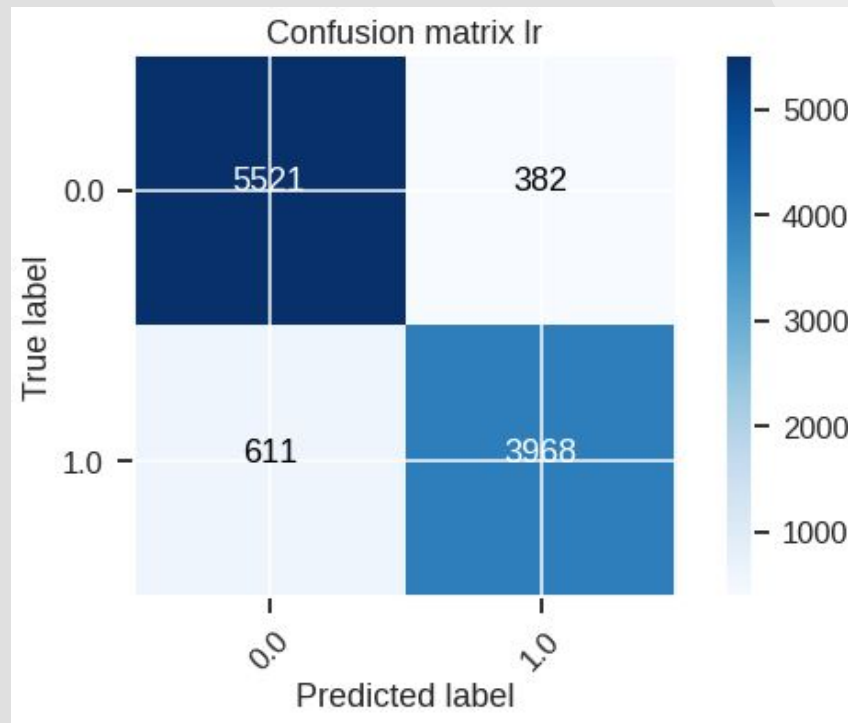
Precision metric: 0.906

Recall Metric: 0.901

Accuracy Metric: 0.905

F1-score Metric: 0.904

AUROC: 0.752



Models evaluation

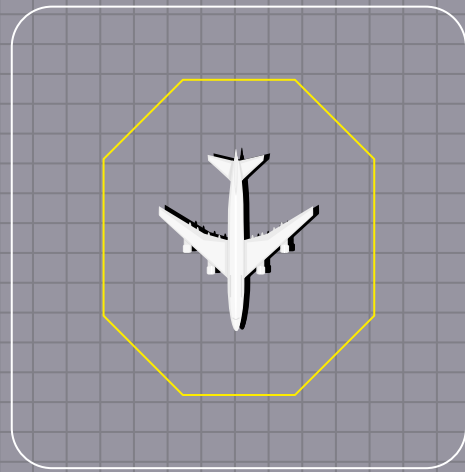
	AUROC	Precision
Logistic Regression	0.925	0.869
Random Forest	0.972	0.917
Decision Tree	0.752	0.906



05

Tuning & K-fold

Tuning hyperparameter and
apply k-fold validation



Tuning hyper- Parameter (AUROC)

Logistic Regression

lambda= 0 alfa=0
maxiter= 1000

0.927

Random Forest

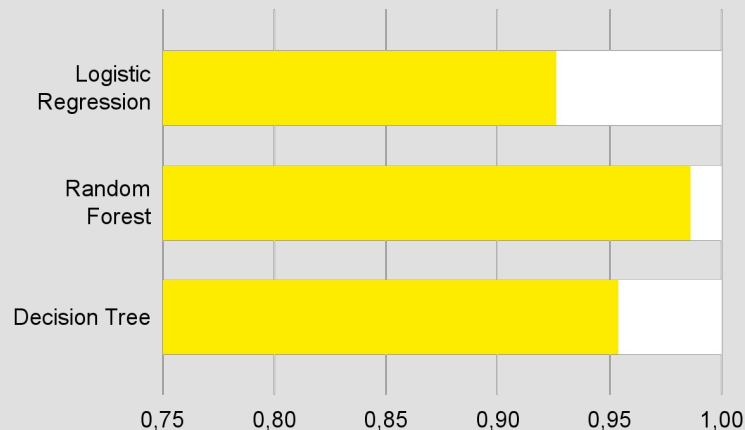
maxDepth = 8
numTrees = 100

0.986

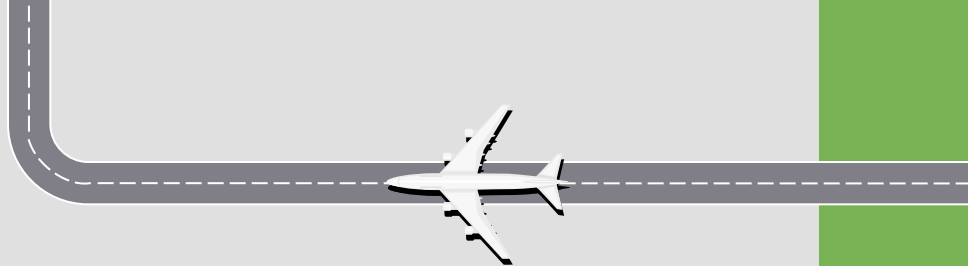
Decision Tree

maxDepth= 8
impurity= entropy

0.945



Precision evaluation



Logistic Regression

lambda= 0 alfa=0
maxiter= 1000

0.868

Random Forest

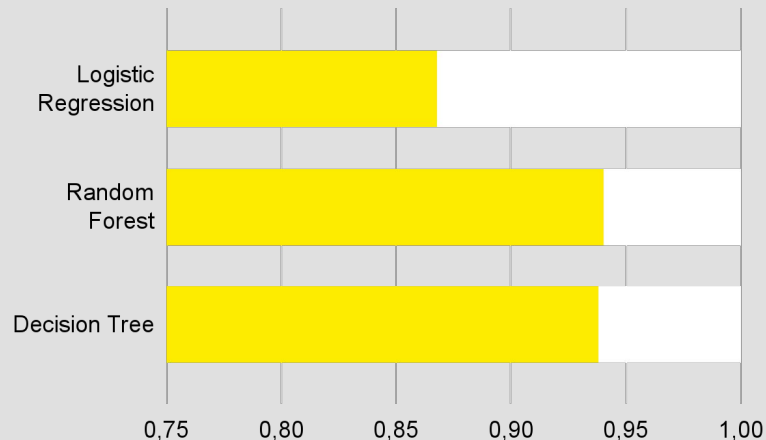
maxDepth = 8
numTrees = 100

0.940

Decision Tree

maxDepth= 8
impurity= entropy

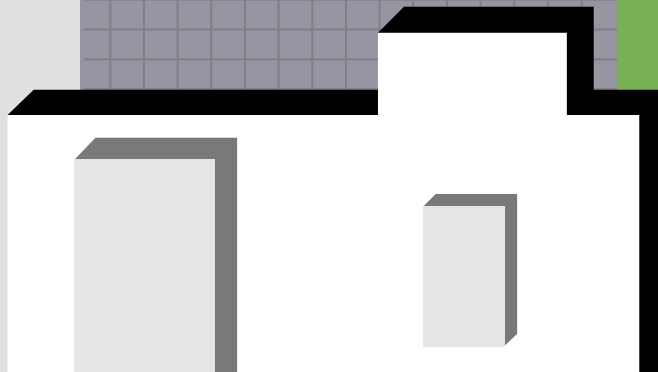
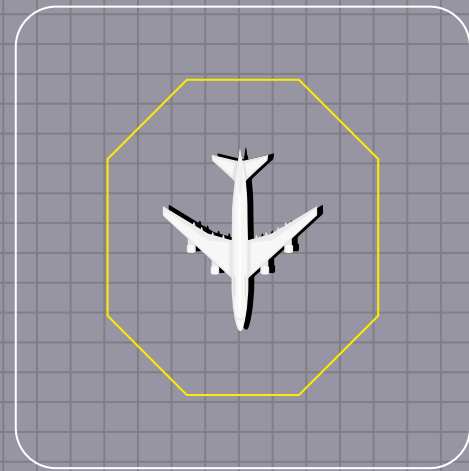
0.938



06

Conclusion

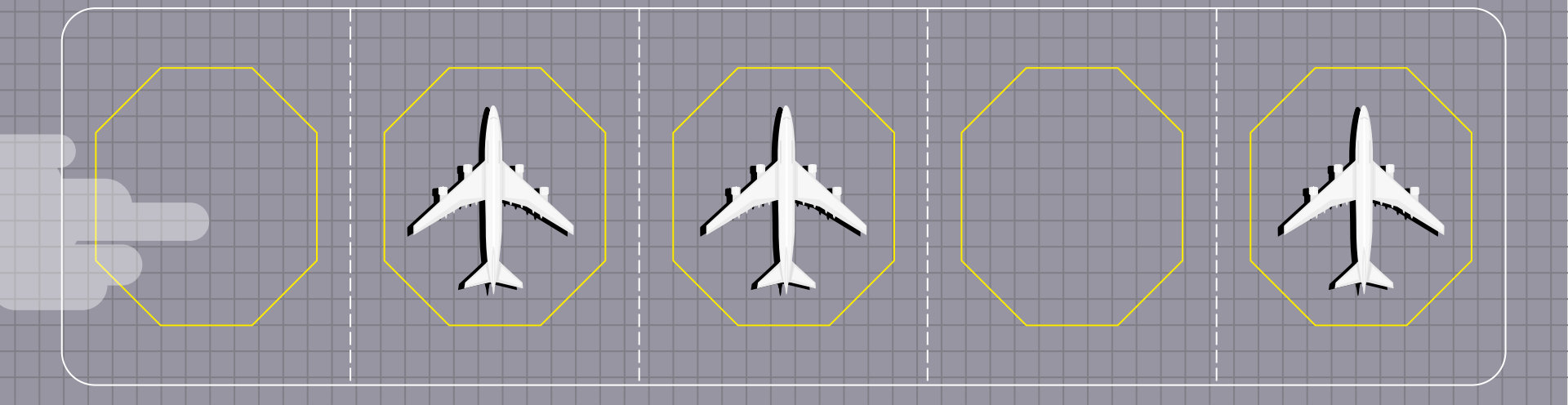
Conclusion and future works





Conclusions

- I trained 3 different models and Random Forest result the best one, and decision tree is the most improved by tuning
- The best results occur for high number of iterations
It's probably I initially occurs in underfitting
- Overcoming the limits of Google Colab it would be possible to perform the tuning with more interaction or trees or use more dense intervals to tune the parameters



Thanks

