

PREDICTION DES PRIX DES VOLS

Présenté par :



AARBAT FARAH

AZZIZ CHAIMAA

GUIGUI SALMA

Encadré par :



PR. FKIHI

PR. HDIOUD

PR. HAMLAOUI

SOMMAIRE

- | | | | |
|----------|------------------|----------|------------------------------------|
| 1 | Contexte général | 5 | Méthodologie MLOps |
| 2 | Jeu de données | 6 | Developpement de l'interface |
| 3 | Prédiction | 7 | Containerisation et Automatisation |
| 4 | Tableau de bord | 8 | Outils et technologies |

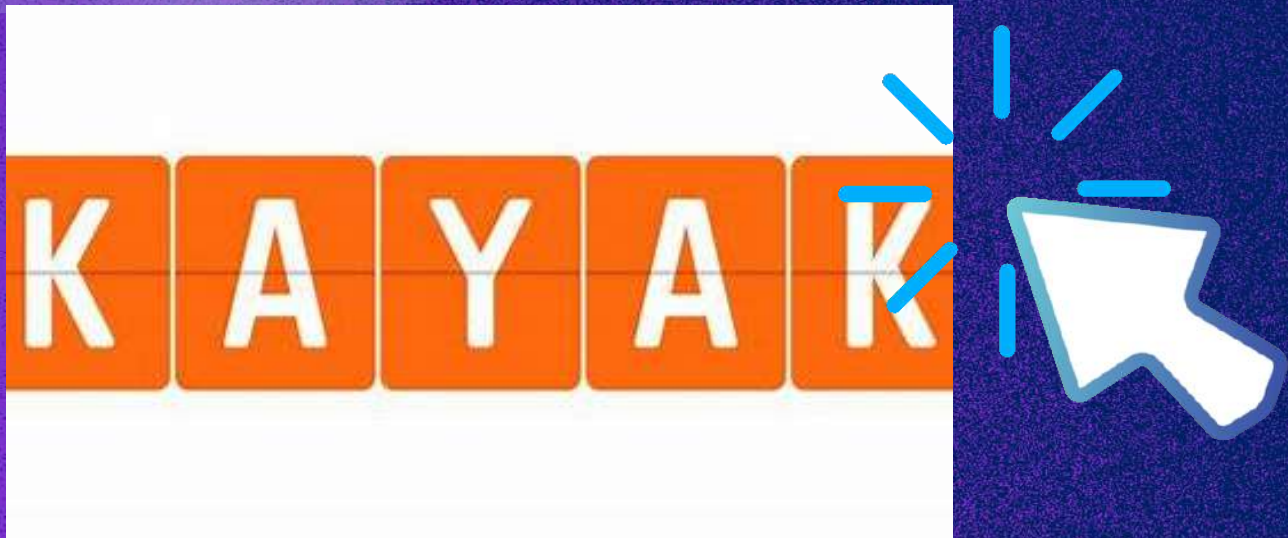


1

CONTEXTE GENERAL

2

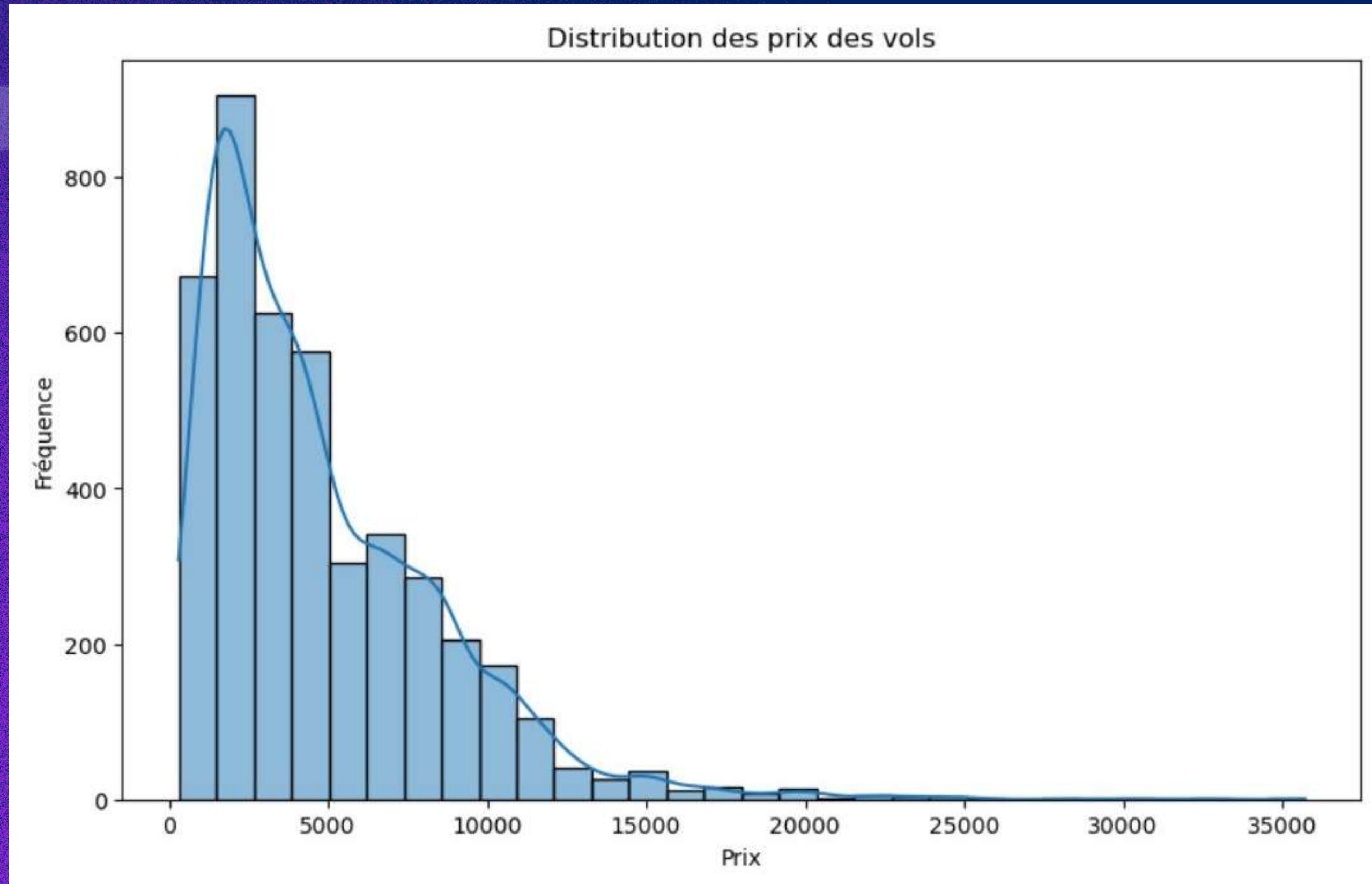
JEU DE DONNEES



	Airline	Source	Destination	Duration	stops	class	depature time	arrival time	Price	Date
0	TUI Fly Belgium	CMN	PAR	3h 30m	direct	economy	07:15	10:45	AED 242	2023-12-09
1	Transavia France	CMN	PAR	3h 15m	direct	economy	19:25	22:40	AED 588	2023-12-09
2	Royal Air Maroc	CMN	PAR	2h 55m	direct	economy	18:30	21:25	AED 686	2023-12-09
3	Royal Air Maroc	CMN	PAR	3h 00m	direct	economy	13:25	16:25	AED 704	2023-12-09
4	Royal Air Maroc	CMN	PAR	2h 55m	direct	economy	16:15	19:10	AED 977	2023-12-09

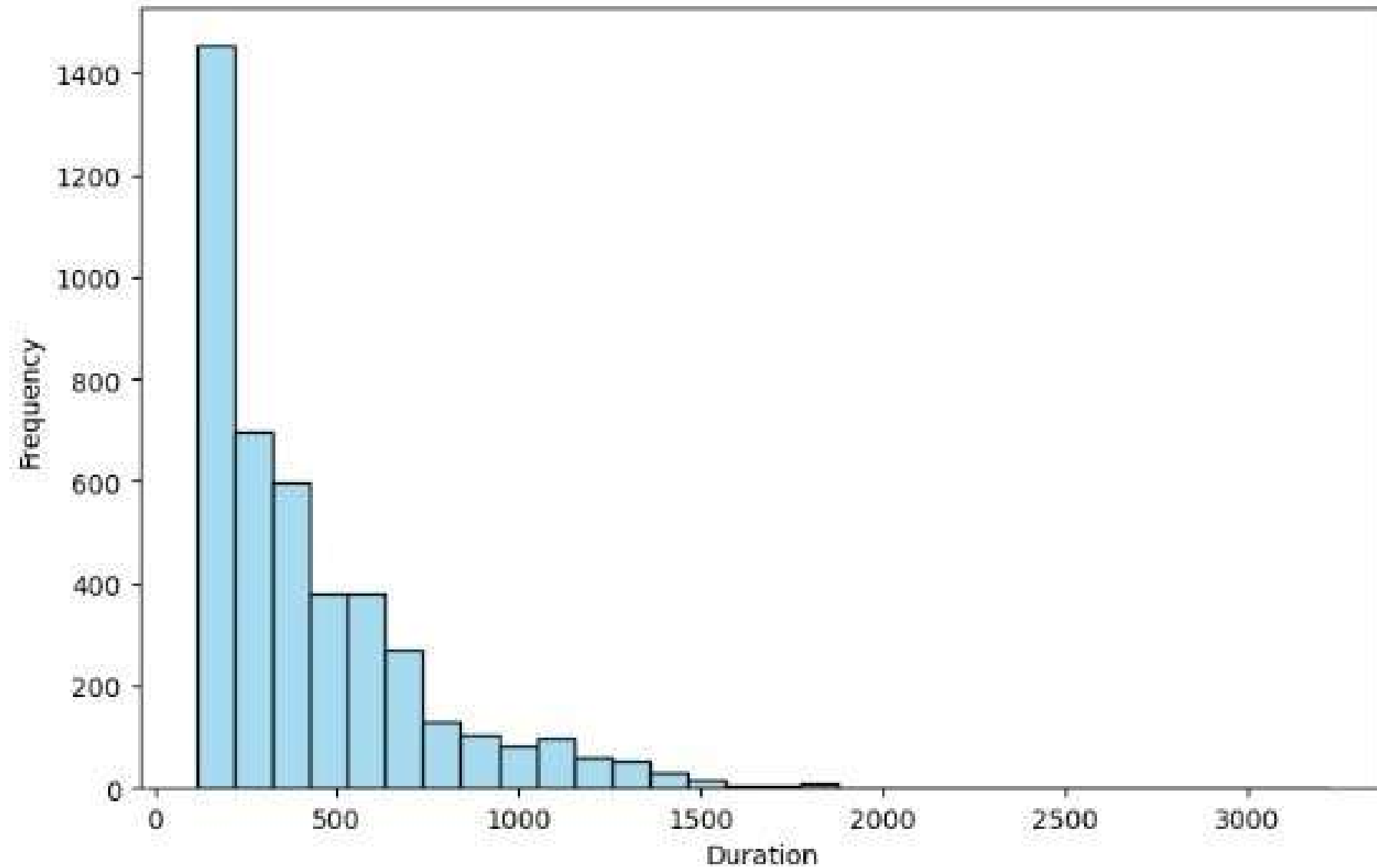
VISUALISATION DES DONNEES

DISTRIBUTION DU PRIX DE VOL

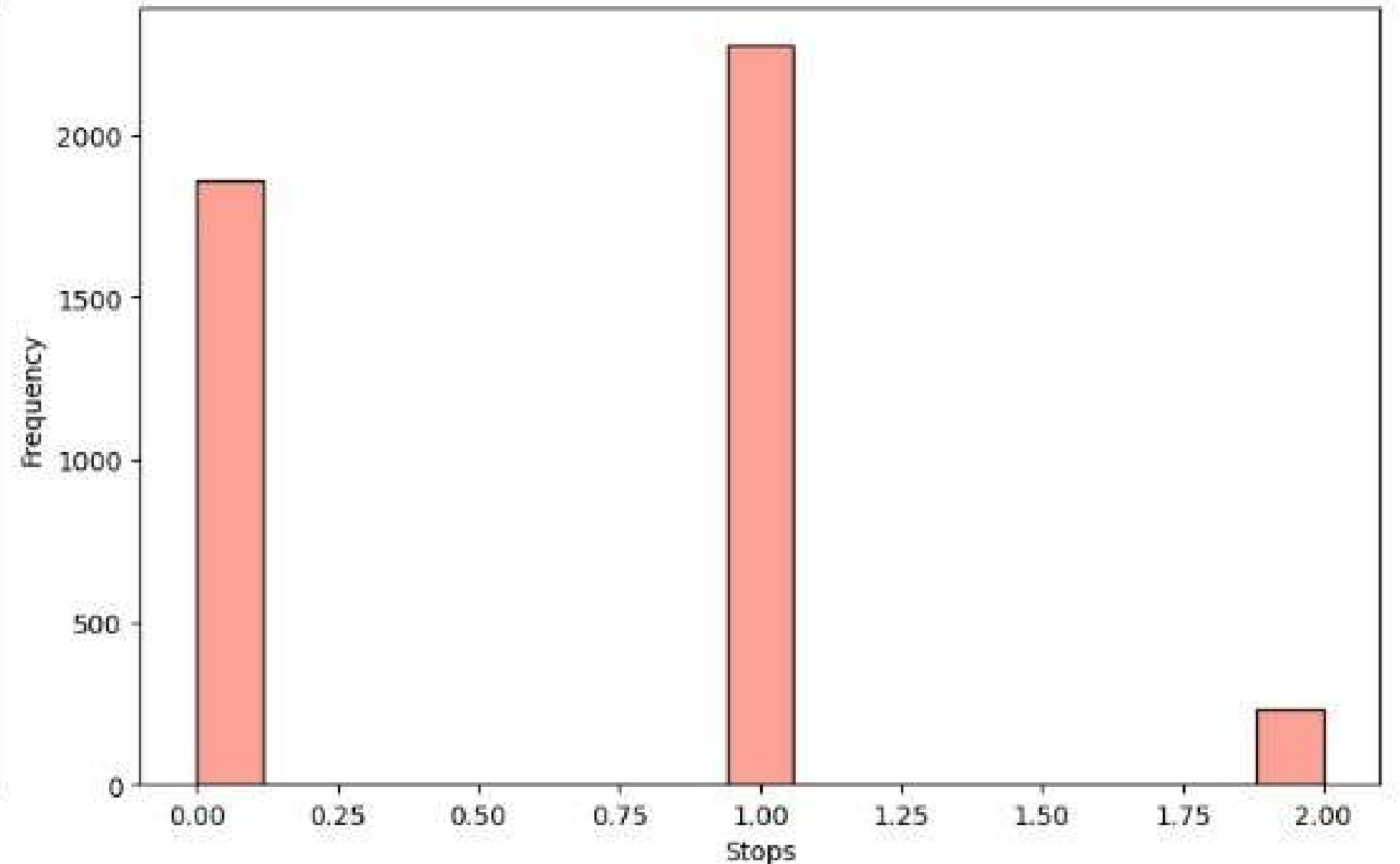


CARACTERISTIQUES NUMERIQUES

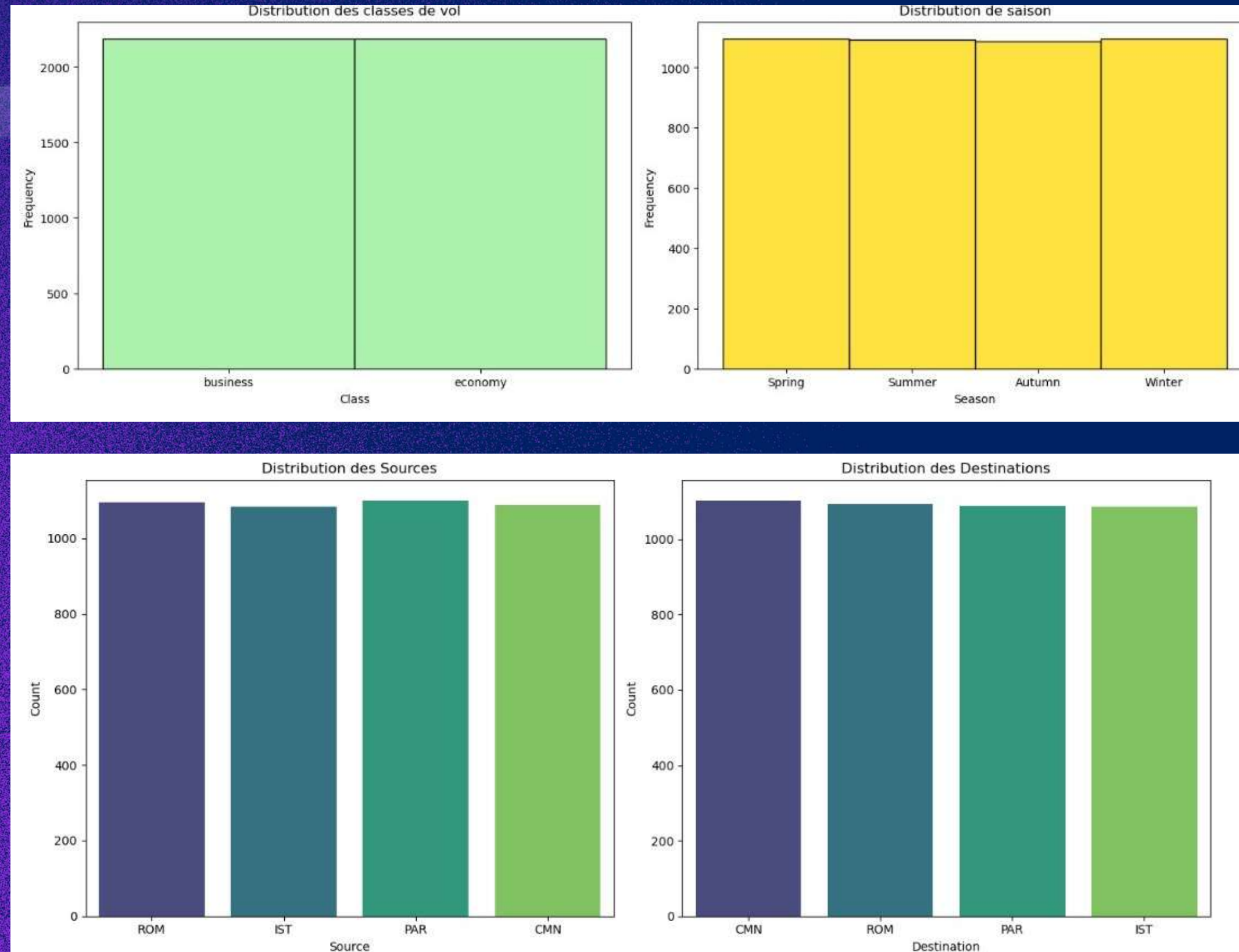
Distribution des durées



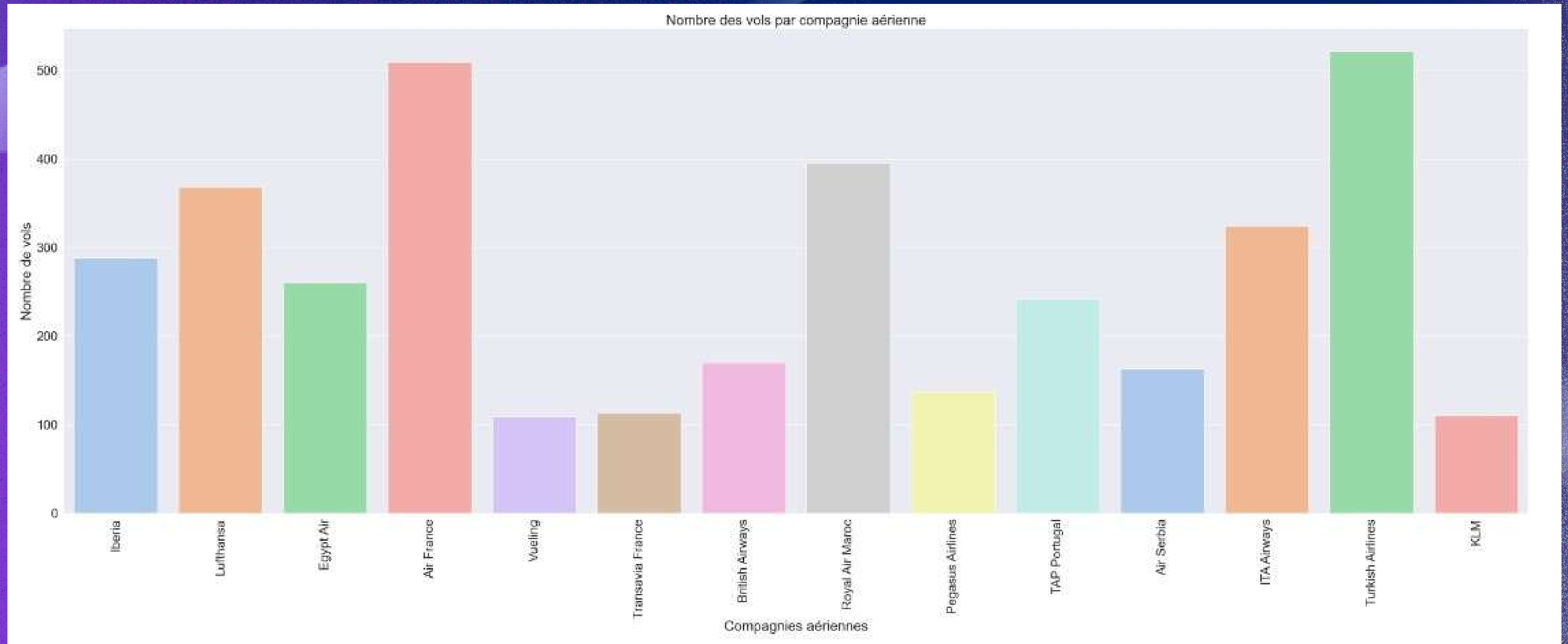
Distribution des escales



CARACTERISTIQUES CATEGORIELLES



CARACTERISTIQUES CATEGORIELLES

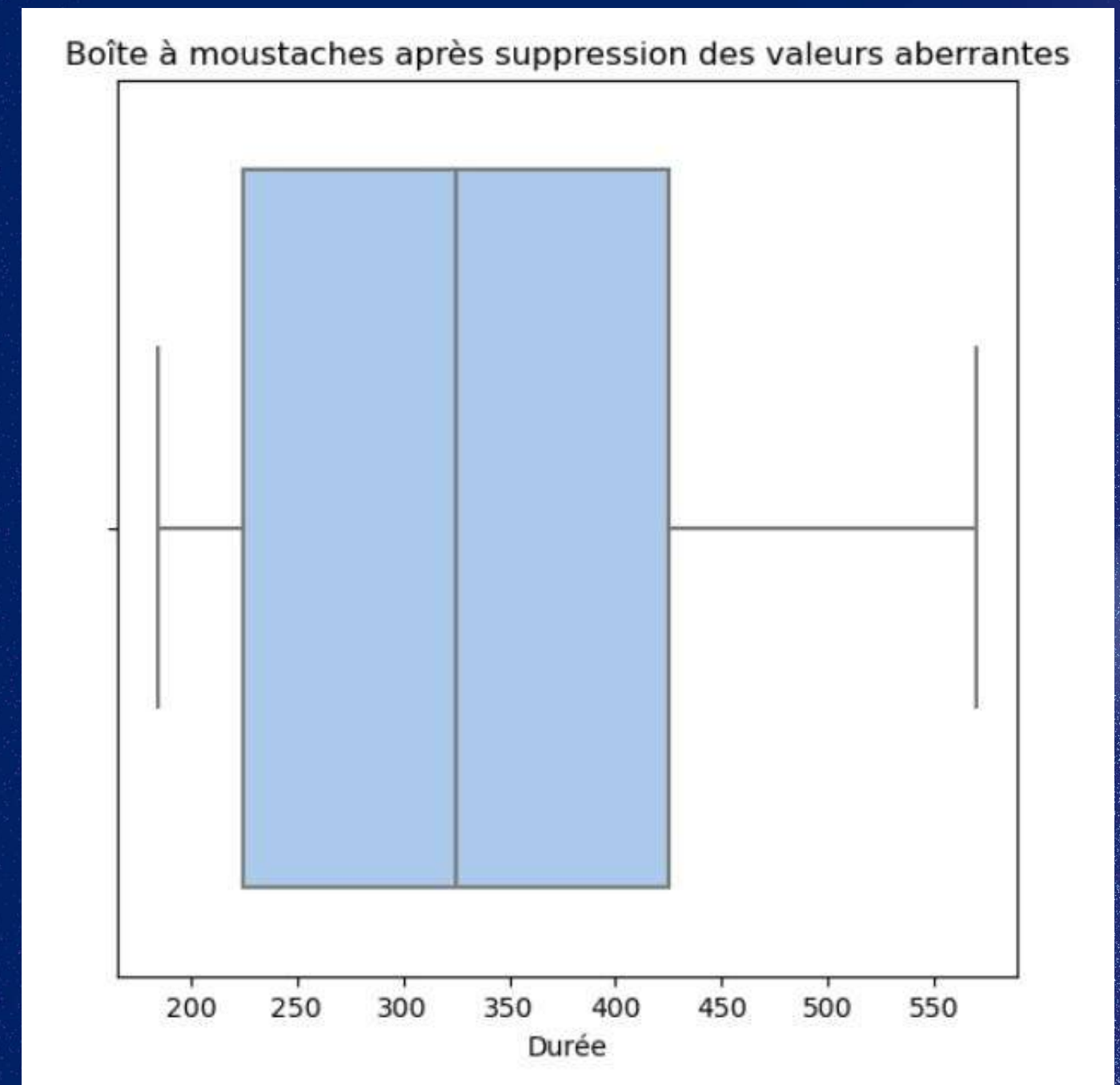
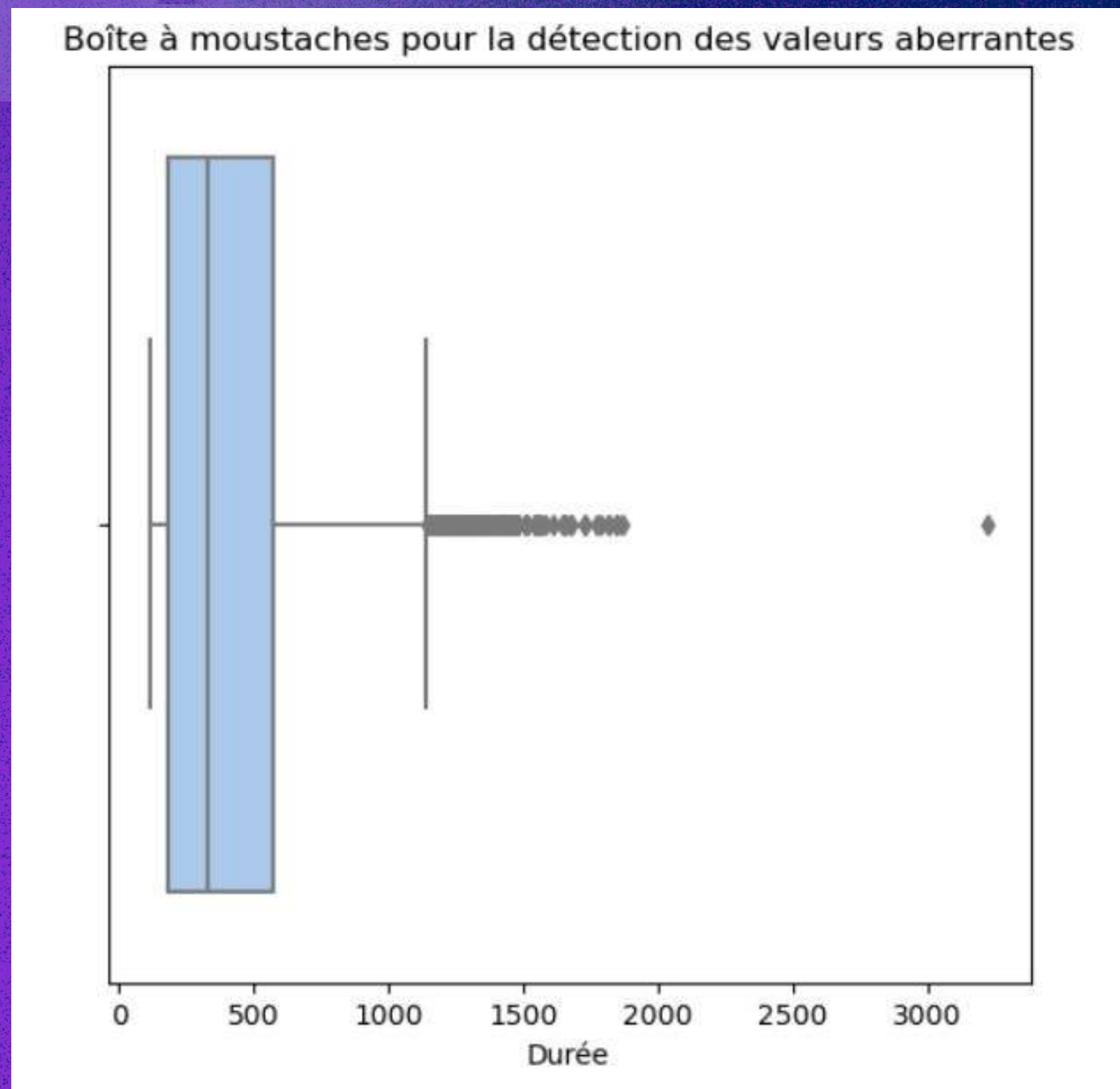


NETTOYAGES DE DONNEES

VALEURS MANQUANTES

	Airline	Source	Destination	Duration	stops	class	depature time	arrival time	Price	Date	Season
1											
4368											

ELIMINATION DES VALEURS ABERRANTES



INGENIERIE DES CARACTERISTIQUES



The diagram illustrates feature relationships. Three blue curved arrows point from the 'Season' column to the 'departure time' and 'arrival time' columns. Additionally, three purple circles are drawn around the 'departure time', 'arrival time', and 'Season' columns, indicating they are the target features for the engineering process.

Airline	Source	Destination	Duration	stops	class	departure time	arrival time	Price	Date	Season
Iberia	ROM	CMN	580	1	business	Morning	Night	4135.95	2024-03-17	Spring
Transavia	ROM	CMN	590	1	economy	Early Morning	Afternoon	2443.35	2024-06-01	Summer
Lufthansa	ROM	CMN	570	2	economy	Afternoon	Late Night	2156.70	2024-09-03	Autumn
Egypt Air	IST	ROM	640	1	business	Late Night	Afternoon	5561.01	2024-02-10	Winter
TAROM	IST	PAR	350	1	business	Morning	Afternoon	8097.18	2024-04-18	Spring
...
Pegasus Airlines	IST	PAR	230	0	economy	Afternoon	Afternoon	3559.92	2024-09-03	Autumn
Iberia	ROM	CMN	505	2	economy	Early Morning	Afternoon	2429.70	2024-01-12	Winter
Turkish Airlines	IST	ROM	160	0	economy	Afternoon	Afternoon	3628.17	2024-08-16	Summer
ITA Airways	ROM	IST	150	0	economy	Morning	Afternoon	1911.00	2024-10-19	Autumn
ITA Airways	ROM	IST	160	0	economy	Early Morning	Morning	1722.63	2024-02-10	Winter

ENCODAGE DES VARIABLES

Airline	Duration	stops	class	depature time	arrival time	Price	Season	Source_CMN	Source_IST	Source_PAR	Source_ROM	Destination_CMN	Destination_IST	Dest
19	570.0	2	1	0	3	2156.70	0	0	0	0	1	1	0	
24	350.0	1	0	4	0	8097.18	1	0	1	0	0	0	0	
19	505.0	1	1	2	3	3194.10	2	0	1	0	0	1	0	
4	525.0	1	1	0	5	1719.90	3	0	0	0	1	1	0	
19	505.0	1	0	2	3	5733.00	2	0	1	0	0	1	0	
...	
4	475.0	1	0	4	2	6060.60	2	1	0	0	0	0	0	
2	550.0	1	1	1	2	2511.60	3	1	0	0	0	0	1	
10	510.0	1	0	4	2	6734.91	2	0	0	1	0	0	0	
20	230.0	0	1	0	0	3559.92	0	0	1	0	0	0	0	
16	505.0	2	1	1	0	2429.70	3	0	0	0	1	1	0	

3

PREDICTION

ALGORITHMES DE PREDICTION

**Decision
Tree**

**XG
Boost**

**Linear
Regression**

**Random
Forest**

Run Name	Created 	mae	mse	r2_score	rmse
 Decision Tree Model	 17 hours ago	925.5325873...	2622911.432...	0.769842296...	30.42256707...
 XGBoost Regressor Model	 17 hours ago	2194.437164...	7795023.645...	0.315994920...	2791.956956...
 Random Forest Regressor Model	 17 hours ago	808.0492776...	1999520.105...	0.824544225...	1414.043883...
 Linear regression model	 17 hours ago	1578.791375...	4626322.074...	0.594045131...	39.73400779...

OPTIMISATION DES HYPERPARAMETRES DU RANDOM FOREST




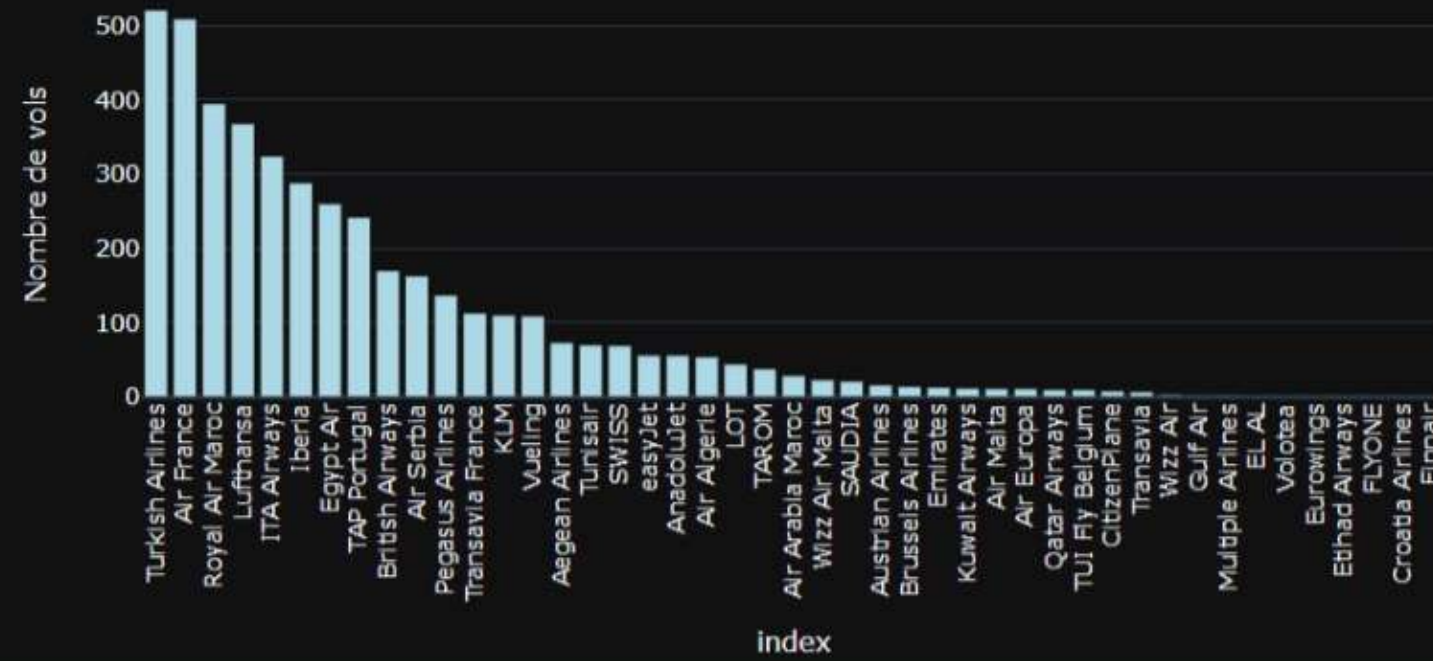
Run Name	Created 	mae	mse	r2_score
 Random Forest Regressor Model	 55 seconds ago	881.4982904...	2182983.044...	0.808445546...
 Random Forest Regressor Model	 3 minutes ago	811.1131232...	2027929.866...	0.822051299...



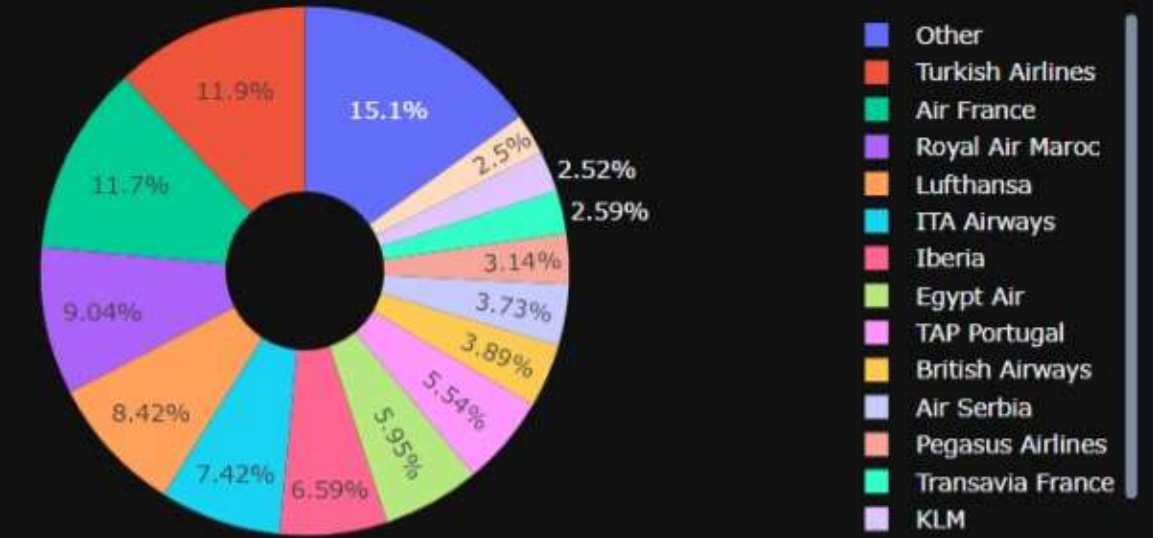
TABLEAU DE BORD

DASHBOARD DE VUE D'ENSEMBLE DES COMPAGNIES AERIENNES

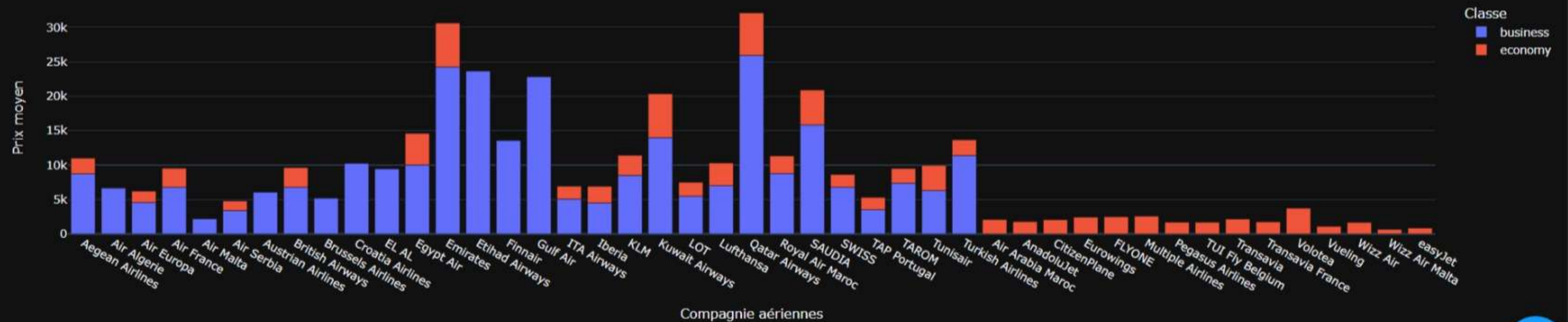
Nombre des vols par compagnie aérienne



Distribution des compagnies aériennes dans le jeux de données

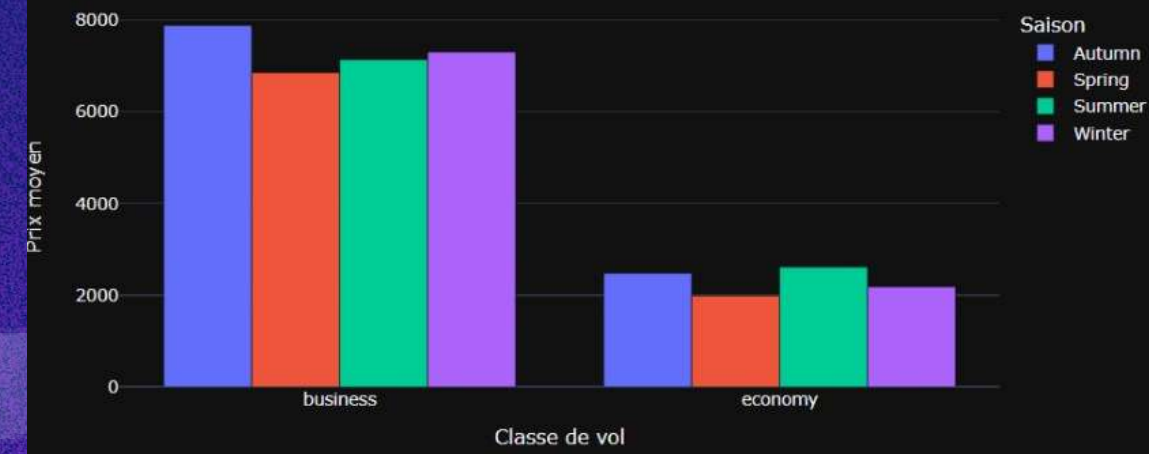


Tarifs moyens des billets pour différentes compagnies aériennes selon la classe

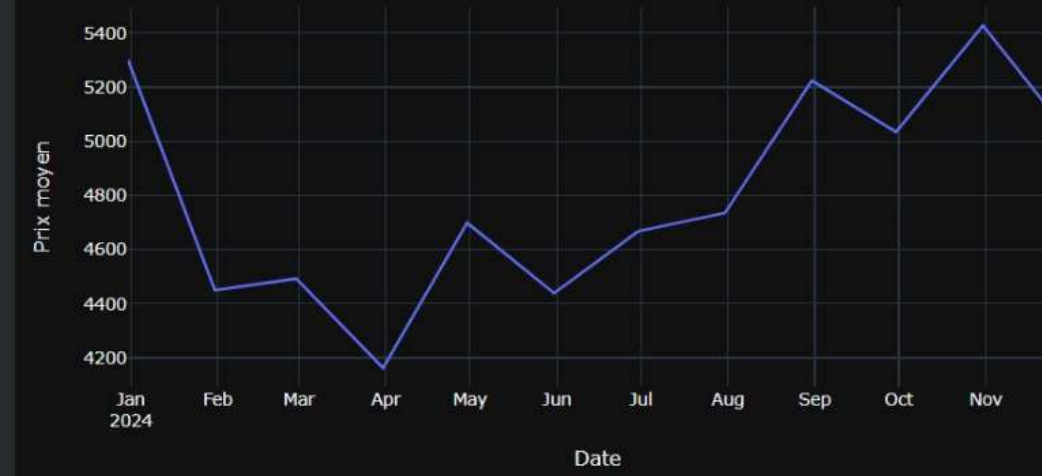


DASHBOARD DES TARIFS EN FONCTION DE DIFFERENTS FACTEURS

Prix moyen par classe de vol et saison



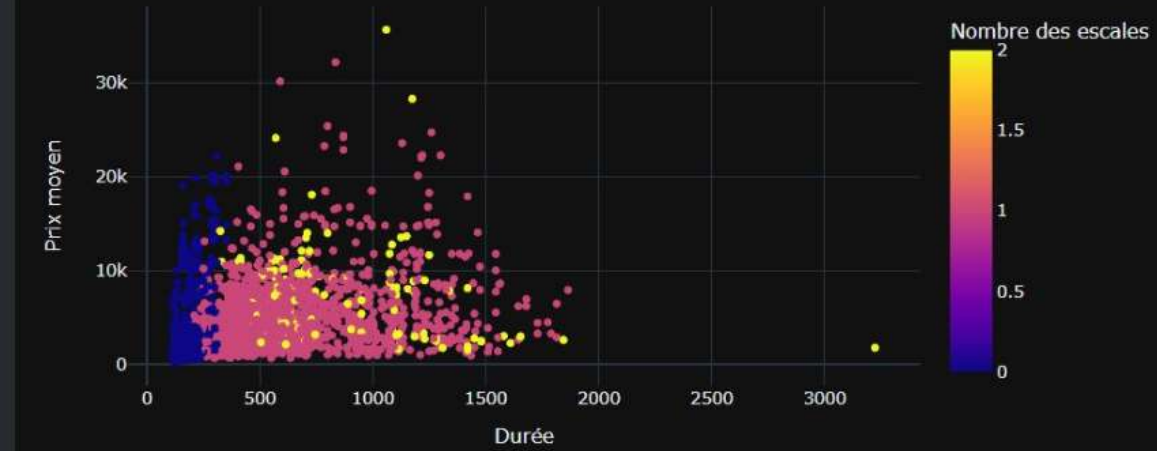
Prix moyen tout au long de l'année



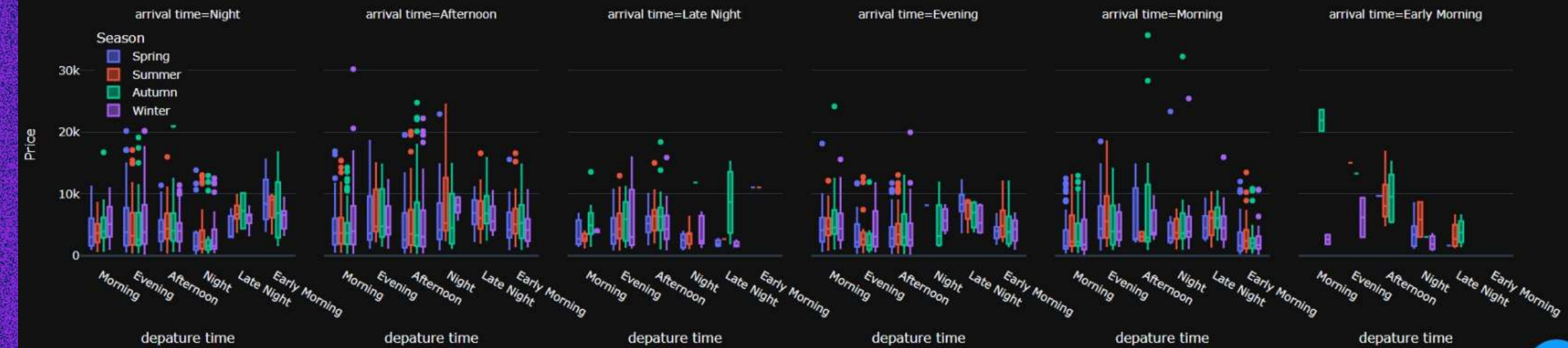
Comparer les prix des vols pour différentes classes



Average Price Based on Duration and Stops



Répartition des prix par heure de départ, heure d'arrivée et saison





METHODOLOGIE

MLOPS

The MLFLOW logo is centered on a dark blue background with abstract, layered geometric shapes in shades of blue and purple. The text "MLFLOW" is rendered in a bold, white, sans-serif typeface. The letters are closely spaced, with the "M" and "L" being particularly prominent. The overall aesthetic is modern and tech-oriented.

MLFLOW

EXPERIMENT TRACKING

mlflow 2.9.2

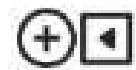
Experiments

Models

GitHub

Docs

Experiments



Search Experiments

☐ Default



☒ Flight price prediction tracki...



Flight price prediction tracking



Provide Feedback

Share

Experiment ID: 514094643223700237

Artifact Location: mlflow-artifacts/514094643223700237

Description Edit

metrics.rmse < 1 and params.model = "tree"



Time created

State: Active

Sort: Created



+ New run

Columns

Table

Chart

Evaluation

Experimental

					Metrics				Parameter
<input type="checkbox"/>		Run Name	Created		mae	mse	r2_score	rmse	alpha
<input type="checkbox"/>		Random Forest Regressor Model	55 seconds ago		881.4982904...	2182983.044...	0.808445546...	1477.492147...	-
<input type="checkbox"/>		Random Forest Regressor Model	3 minutes ago		811.1131232...	2027929.866...	0.822051299...	1424.054025...	-
<input type="checkbox"/>		Decision Tree Model	4 minutes ago		921.6233624...	2655056.560...	0.767021595...	30.35825031...	-
<input type="checkbox"/>		XGBoost Regressor Model	4 minutes ago		2194.437164...	7795023.645...	0.315994920...	2791.956956...	10
<input type="checkbox"/>		Linear regression model	4 minutes ago		1578.791375...	4626322.074...	0.594045131...	39.73400779...	-
<input type="checkbox"/>		Decision Tree Model	17 hours ago		925.5325873...	2622911.432...	0.769842296...	30.42256707...	-
<input type="checkbox"/>		XGBoost Regressor Model	17 hours ago		2194.437164...	7795023.645...	0.315994920...	2791.956956...	10
<input type="checkbox"/>		Random Forest Regressor Model	17 hours ago		808.0492776...	1999520.105...	0.824544225...	1414.043883...	-
<input type="checkbox"/>		Linear regression model	17 hours ago		1578.791375...	4626322.074...	0.594045131...	39.73400779...	-

10 matching runs

MODEL PACKAGING

Artifacts

Random Forest Regressor Model

- MLmodel
- conda.yaml
- model.pkl
- python_env.yaml
- requirements.txt

Full Path: mlflow-artifacts/385954286924491928/a14bd7fae8354a01821bd29535102286/artifacts/Random Forest Regressor Model

Register Model

MLflow Model

The code snippets below demonstrate how to make predictions using the logged model. You can also [register it to the model registry](#) to version control

Model schema

Input and output schema for your model. [Learn more](#)

Name	Type
No schema. See MLflow docs for how to include input and output schema with your model.	

Make Predictions

Predict on a Spark DataFrame:

```
import mlflow
from pyspark.sql.functions import struct, col
logged_model = 'runs:/a14bd7fae8354a01821bd29535102286/Random Forest Regressor Model'

# Load model as a Spark UDF. Override result_type if the model does not return double values.
loaded_model = mlflow.pyfunc.spark_udf(spark, model_uri=logged_model, result_type='double')

# Predict on a Spark DataFrame.
df.withColumn('predictions', loaded_model(struct(*map(col, df.columns))))
```

Predict on a Pandas DataFrame:

```
import mlflow
logged_model = 'runs:/a14bd7fae8354a01821bd29535102286/Random Forest Regressor Model'

# Load model as a PyFuncModel.
loaded_model = mlflow.pyfunc.load_model(logged_model)

# Predict on a Pandas DataFrame.
import pandas as pd
loaded_model.predict(pd.DataFrame(data))
```




DEVELOPPEMENT DE L'INTERFACE

CREATION DE L'API

▼ Random Forest Regressor Model

MLmodel

conda.yaml

model.pkl

python_env.yaml

requirements.txt

Full Path:mlflow-artifacts:/385954286924491928/a14bd7fae8354a01821bd29535102286/artifacts/Random Forest Regressor Model

Register Model

MLflow Model

The code snippets below demonstrate how to make predictions using the logged model. You can also [register it to the model registry](#) to version control

← ↻ ⓘ localhost:8000/docs#/

FastAPI

0.1.0 OAS 3.1

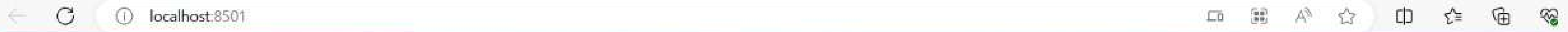
/openapi.json

default

GET / Read Root

POST /predict Predict

INTERFACE WEB



Deploy



Flight Price Prediction

INTERFACE WEB

Deploy 

Departure Date

2024/01/01

Departure Time

19:53



Arrival Date

2024/01/02

Arrival Time

00:15



Number of Stops



0



1



2



3

Airline

Royal Air Maroc



Class

Business



INTERFACE WEB

Deploy



19:59



Number of Stops

☐ 0

☐ 1

☒ 2

☐ 3

Airline

Royal Air Maroc



Class

Business



Source

Casablanca



Destination

Paris



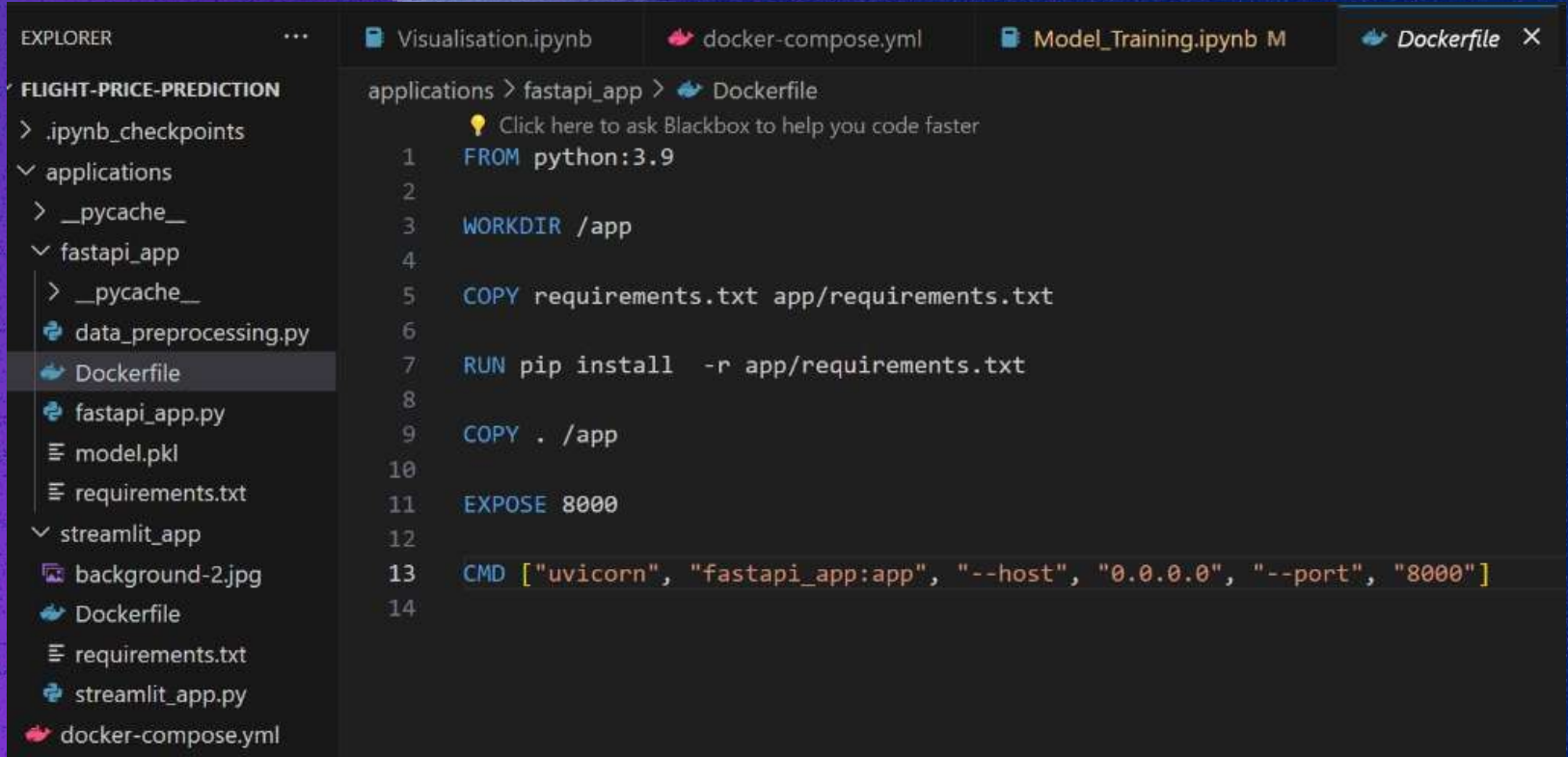
Predict

Predicted Flight Price: 10825.75 MAD

7

CONTAINERISATION

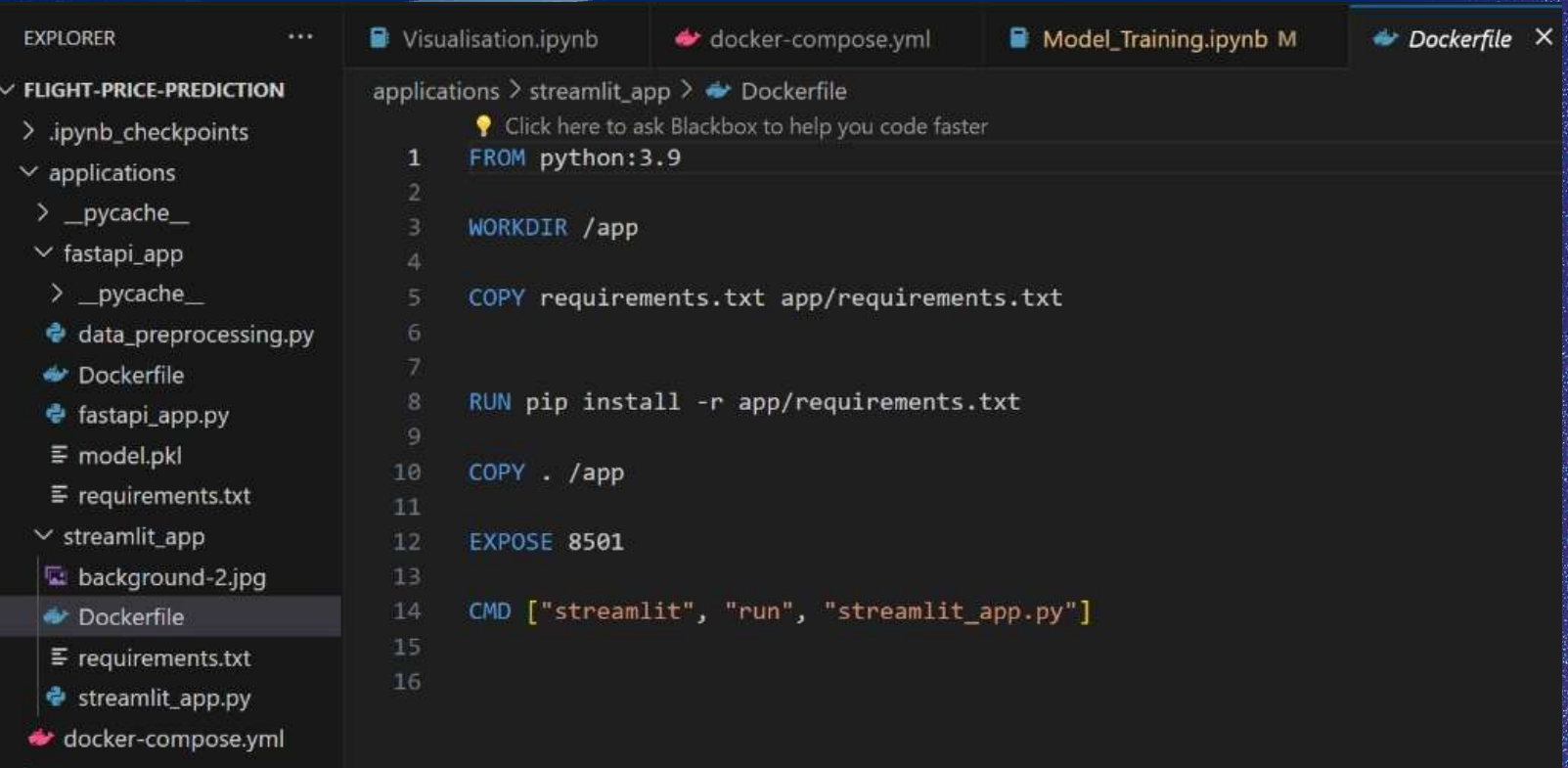
CREATION DES IMAGES



This screenshot shows the VS Code interface with the Explorer sidebar on the left displaying the file structure of the 'FLIGHT-PRICE-PREDICTION' project. The 'Dockerfile' file is selected and its content is shown in the main editor. The Dockerfile defines a container for the FastAPI application, starting from a Python 3.9 base image, setting the working directory to /app, installing requirements, and running the application using uvicorn on port 8000.

```
1 FROM python:3.9
2
3 WORKDIR /app
4
5 COPY requirements.txt app/requirements.txt
6
7 RUN pip install -r app/requirements.txt
8
9 COPY . /app
10
11 EXPOSE 8000
12
13 CMD ["uvicorn", "fastapi_app:app", "--host", "0.0.0.0", "--port", "8000"]
14
```

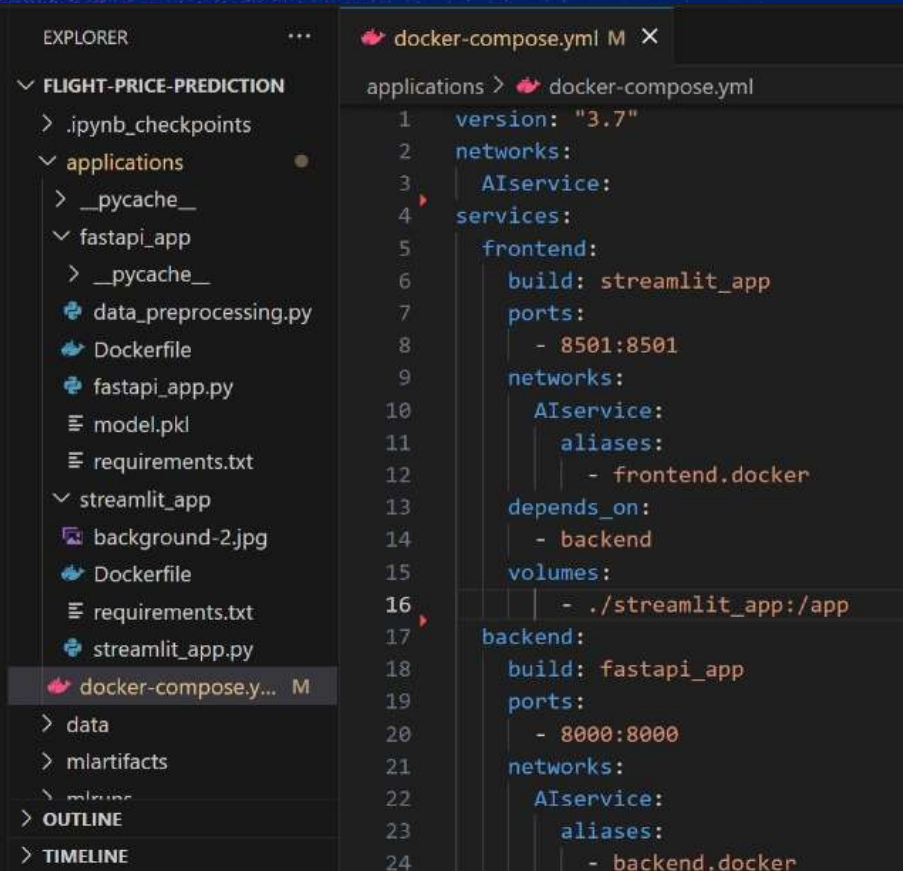
**FASTAPI
Docker File**



This screenshot shows the VS Code interface with the Explorer sidebar on the left displaying the file structure of the 'FLIGHT-PRICE-PREDICTION' project. The 'Dockerfile' file is selected and its content is shown in the main editor. The Dockerfile defines a container for the Streamlit application, starting from a Python 3.9 base image, setting the working directory to /app, installing requirements, and running the application using streamlit on port 8501.

```
1 FROM python:3.9
2
3 WORKDIR /app
4
5 COPY requirements.txt app/requirements.txt
6
7 RUN pip install -r app/requirements.txt
8
9 COPY . /app
10
11 EXPOSE 8501
12
13 CMD ["streamlit", "run", "streamlit_app.py"]
14
15
16
```

**Streamlit App
Docker File**

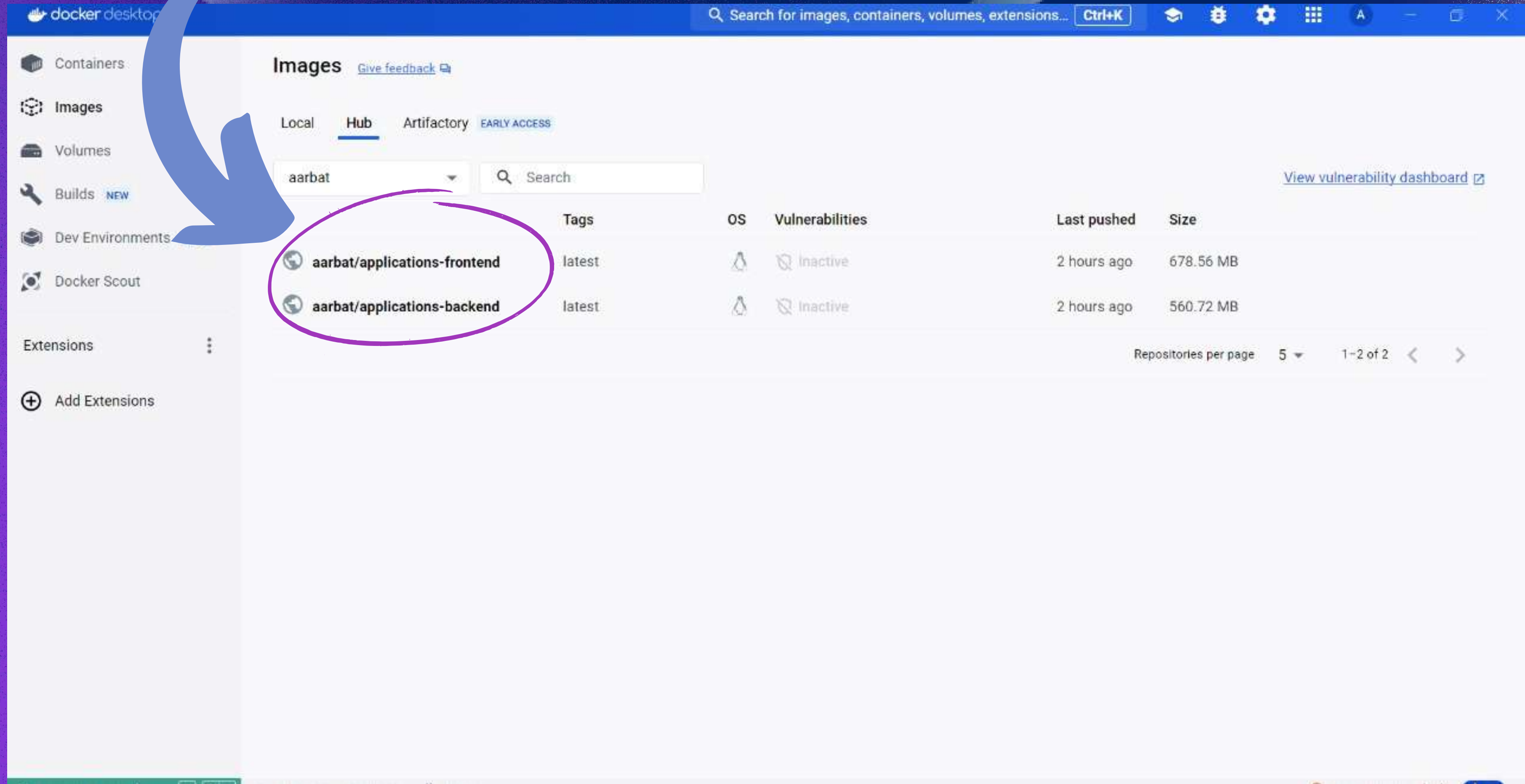


This screenshot shows the VS Code interface with the Explorer sidebar on the left displaying the file structure of the 'FLIGHT-PRICE-PREDICTION' project. The 'docker-compose.yml' file is selected and its content is shown in the main editor. The docker-compose.yml file defines two services: 'frontend' (Streamlit app) and 'backend' (FastAPI app), which are connected to a common network 'AIService' and share a volume 'app'.

```
1 version: "3.7"
2 networks:
3   AIService:
4 services:
5   frontend:
6     build: streamlit_app
7     ports:
8       - 8501:8501
9     networks:
10       AIService:
11         aliases:
12           - frontend.docker
13     depends_on:
14       - backend
15     volumes:
16       - ./streamlit_app:/app
17   backend:
18     build: fastapi_app
19     ports:
20       - 8000:8000
21     networks:
22       AIService:
23         aliases:
24           - backend.docker
```

**Docker
Compose**

PUSH SUR DOCKER HUB









The screenshot shows the Docker Desktop interface with the 'Images' section selected. The 'Hub' tab is active, displaying a list of images from the 'aarbat' repository. A blue arrow points from the title 'PUSH SUR DOCKER HUB' to the 'Hub' tab. A purple circle highlights the two repository entries: 'aarbat/applications-frontend' and 'aarbat/applications-backend'.

Images [Give feedback](#)

Local **Hub** Artifactory EARLY ACCESS

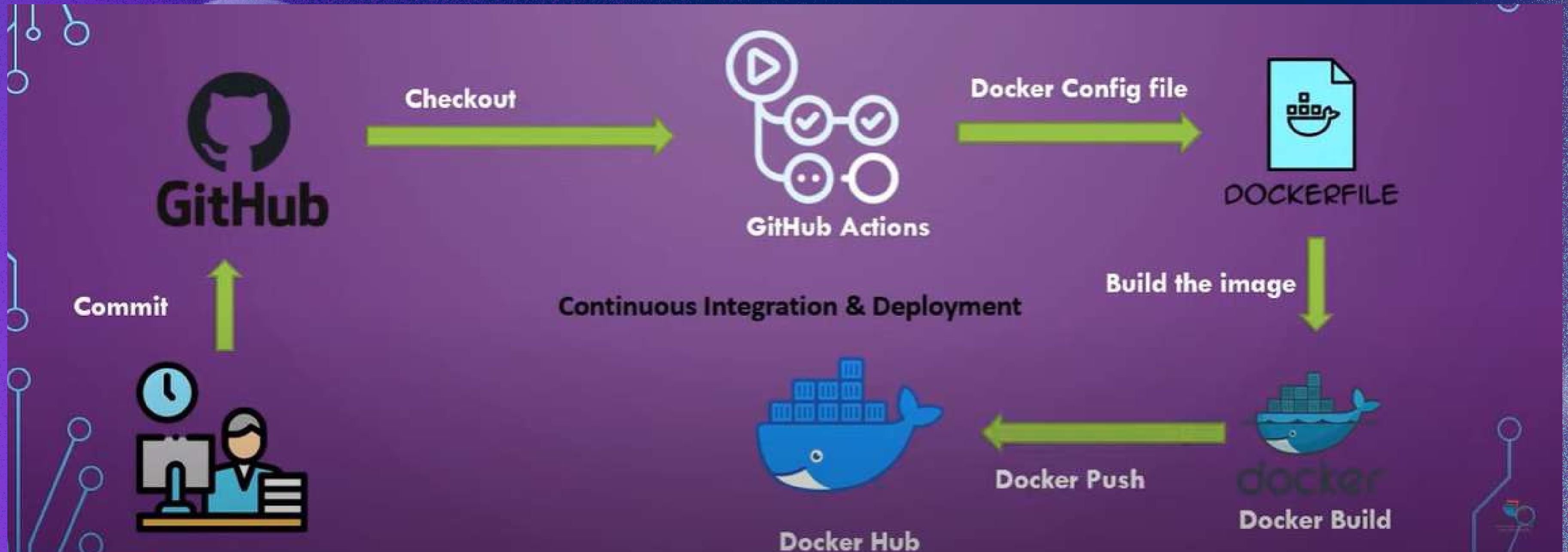
aarbat

[View vulnerability dashboard](#)

	Tags	OS	Vulnerabilities	Last pushed	Size
 aarbat/applications-frontend	latest		 Inactive	2 hours ago	678.56 MB
 aarbat/applications-backend	latest		 Inactive	2 hours ago	560.72 MB

Repositories per page 5 1-2 of 2 < >

IMPLEMENTATION D'UN CI/CD PIPELINE



DECLenchement automatique du pipeline apres un push

aarbatfarah1 / flight-price-prediction-test2

Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Search

Type to search

>

+

Actions

New workflow

All workflows

CI/CD Pipeline

Management

Caches

Runners

All workflows

Showing runs from all workflows

2 workflow runs

Event Status Branch Actor

Merge branch 'main' of https://github.com/aarbatfarah1/flight-price-p...

CI/CD Pipeline #2: Commit [ff07950](#) pushed by aarbatfarah1

main

3 minutes ago

In progress

...

Create main.yml

CI/CD Pipeline #1: Commit [99777d4](#) pushed by aarbatfarah1

main

24 minutes ago

14m 4s

...

RESULTAT D'EXECUTION

<> Code Issues Pull requests **Actions** Projects Wiki Security Insights Settings

← CI/CD Pipeline

✓ Merge branch 'main' of https://github.com/aarbatfarah1/flight-price-p... #2

Re-run all jobs

...

Summary

Jobs

✓ build-and-push-docker-images

✓ deploy-docker-compose

Run details

Usage

Workflow file

build-and-push-docker-images

succeeded 2 minutes ago in 9m 51s

Search logs



> ✓ Set up job	1s
> ✓ Checkout Repository	4s
> ✓ Set up Docker Buildx	8s
> ✓ Login to Docker Hub	0s
> ✓ Build and Push Backend Docker Image	4m 11s
> ✓ Build and Push Frontend Docker Image	5m 25s
> ✓ Post Set up Docker Buildx	0s
> ✓ Post Checkout Repository	0s
> ✓ Complete job	0s

8

OUTILS ET TECHNOLOGIES



Github



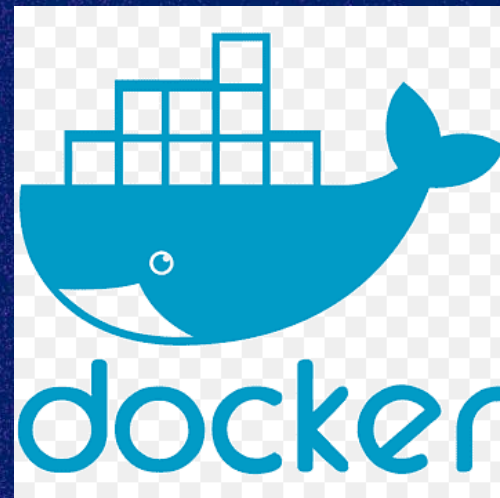
Scikit learn



Python



VS-Code



mlflow™



**MERCI DE
VOTRE
ATTENTION!**

