



Licence MIASHS première année

## Rapport de projet informatique

# Prédiction des prix des actions du S&P 500 à l'aide de Machine Learning

Projet réalisé du 10/2025 au 12/2025

### Membres du groupe

**BOUDJADJA Anis (45000199)** – [https://github.com/Anis017/SP500-Prediction-Aziz-Amor \(44020039\)](https://github.com/Anis017/SP500-Prediction-Aziz-Amor)  
Membre Membre Membre

## **Remerciements**

Nous tenons à remercier notre encadrant pour ses conseils précieux, ainsi que les outils open-source qui ont facilité le développement de ce projet.

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Environnement de travail</b>	<b>4</b>
<b>3</b>	<b>Description du projet et objectifs</b>	<b>4</b>
3.1	Problématique . . . . .	4
3.2	Objectifs . . . . .	4
<b>4</b>	<b>Bibliothèques, Outils et technologies</b>	<b>4</b>
<b>5</b>	<b>Travail réalisé</b>	<b>5</b>
<b>6</b>	<b>Aide apportée par l'intelligence artificielle</b>	<b>5</b>
<b>7</b>	<b>Difficultés rencontrées</b>	<b>5</b>
<b>8</b>	<b>Démonstration du projet</b>	<b>6</b>
8.1	Page d'accueil . . . . .	6
8.2	Sélection et visualisation des prédictions . . . . .	6
8.3	Export et résultats . . . . .	7
<b>9</b>	<b>Bilan</b>	<b>7</b>
9.1	Conclusion . . . . .	7
9.2	Perspectives . . . . .	7
<b>10</b>	<b>Bibliographie</b>	<b>8</b>
<b>11</b>	<b>Webographie</b>	<b>9</b>
<b>12</b>	<b>Annexes</b>	<b>10</b>
<b>A</b>	<b>Cahier des charges</b>	<b>10</b>
<b>B</b>	<b>Exemple d'exécution du projet</b>	<b>10</b>
<b>C</b>	<b>Manuel utilisateur</b>	<b>10</b>

# 1 Introduction

Ce projet consiste en la réalisation d'une application web permettant de prédire les prix futurs des actions composant l'indice S&P 500 à l'aide de techniques de Machine Learning, plus précisément un modèle de séries temporelles autorégressif (AutoReg).

L'objectif principal est de fournir aux utilisateurs un outil intuitif pour visualiser les données historiques et obtenir des prévisions à court et moyen terme (5 à 180 jours), tout en soulignant les limites inhérentes à la prédiction boursière.

Le projet s'inscrit dans le cadre d'un travail académique en informatique, en mettant l'accent sur l'utilisation de bibliothèques Python modernes pour le traitement de données financières.

## 2 Environnement de travail

Le développement a été réalisé en Python 3, avec un environnement virtuel pour gérer les dépendances. L'application est déployée localement via Streamlit, qui permet une interface web interactive sans serveur complexe.

Les outils principaux utilisés :

- Éditeur : VS Code ou PyCharm.
- Gestion de versions : Git et GitHub (dépôt public : <https://github.com/Anis017/SP500-Prediction->)
- Exécution : Streamlit pour l'interface utilisateur.

## 3 Description du projet et objectifs

### 3.1 Problématique

La prédiction des prix des actions est un défi majeur en raison de la nature volatile et influencée par de nombreux facteurs externes des marchés financiers. L'indice S&P 500 regroupe 500 grandes entreprises américaines, et ses composants sont souvent analysés pour des décisions d'investissement.

### 3.2 Objectifs

- Récupérer des données historiques en temps réel via l'API Yahoo Finance.
- Appliquer un modèle AutoReg pour prévoir les prix futurs.
- Fournir une interface utilisateur conviviale avec graphiques interactifs et export CSV.
- Limiter les prévisions à des horizons raisonnables (jusqu'à 180 jours).

## 4 Bibliothèques, Outils et technologies

Le projet repose sur les bibliothèques suivantes :

- **Streamlit** : Interface web interactive.
- **yfinance** : Récupération des données boursières.
- **statsmodels** : Implémentation du modèle AutoReg.
- **Plotly** : Graphiques interactifs.

- **Pandas** : Manipulation des données.

La structure du projet est la suivante :

```
SP500-PREDICTION-
assets/
  data/
    sp500_tickers.csv
streamlit_app/
  modules/
    helper.py
  pages/
    01__StockPredictor.py
    00_Info.py
LICENSE
README.md
requirements.txt
```

## 5 Travail réalisé

Les fonctionnalités prévues et réalisées :

- Sélection d'un ticker parmi les entreprises du S&P 500 : réalisée.
  - Récupération des données historiques (2 ans) : réalisée.
  - Entraînement du modèle AutoReg et prévision (5 à 180 jours) : réalisée.
  - Affichage interactif avec Plotly (historique + prévision) : réalisée.
  - Téléchargement des prévisions en CSV : réalisée.
  - Interface multilingue/informations : réalisée via pages Streamlit.
- Aucune fonctionnalité majeure n'a été abandonnée.

### Répartition du travail :

- Amor : Développement de l'interface Streamlit et intégration Plotly.
- Anis : Implémentation du modèle AutoReg, gestion des données yfinance et tests.

## 6 Aide apportée par l'intelligence artificielle

L'intelligence artificielle (notamment Grok de xAI et/ou ChatGPT) a été utilisée à plusieurs étapes :

- Génération et optimisation de code (ex. : traitement des données, débogage yfinance).
- Aide conceptuelle pour le choix et le paramétrage du modèle AutoReg.
- Rédaction et structuration de parties du rapport.
- Suggestions pour l'amélioration de l'interface Streamlit.

Cela a accéléré le développement tout en approfondissant la compréhension des concepts.

## 7 Difficultés rencontrées

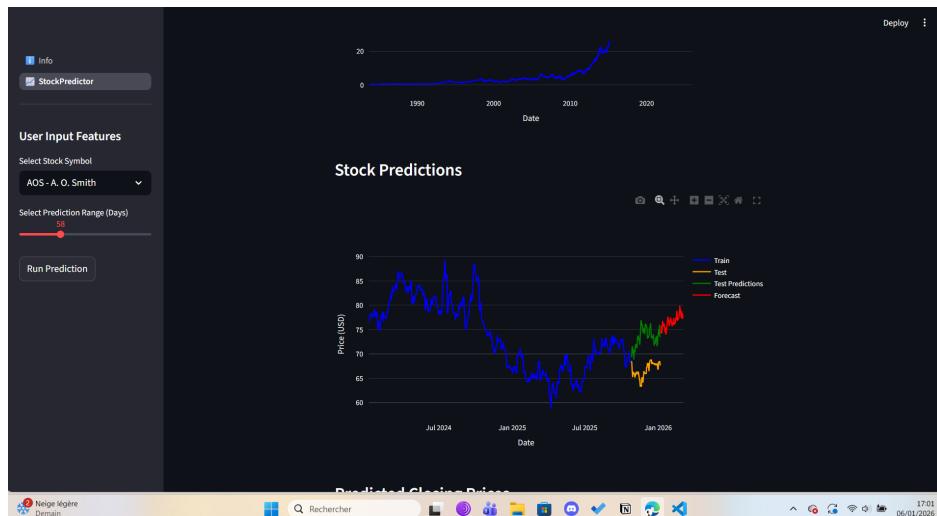
- Volatilité des marchés rendant les prévisions imprécises à long terme.
- Choix optimal des lags dans le modèle AutoReg.

- Gestion des données manquantes (jours fériés).
- Performances variables selon la liquidité des tickers.

## 8 Démonstration du projet

Cette section présente les captures d'écran de l'application en action, montrant les différentes fonctionnalités et l'interface utilisateur.

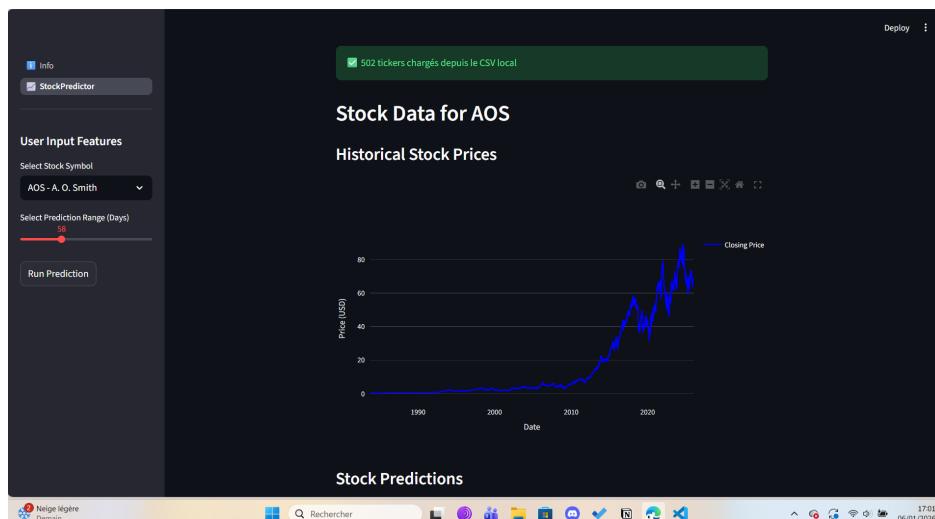
### 8.1 Page d'accueil



figurePage

d'accueil de l'application avec les informations du projet

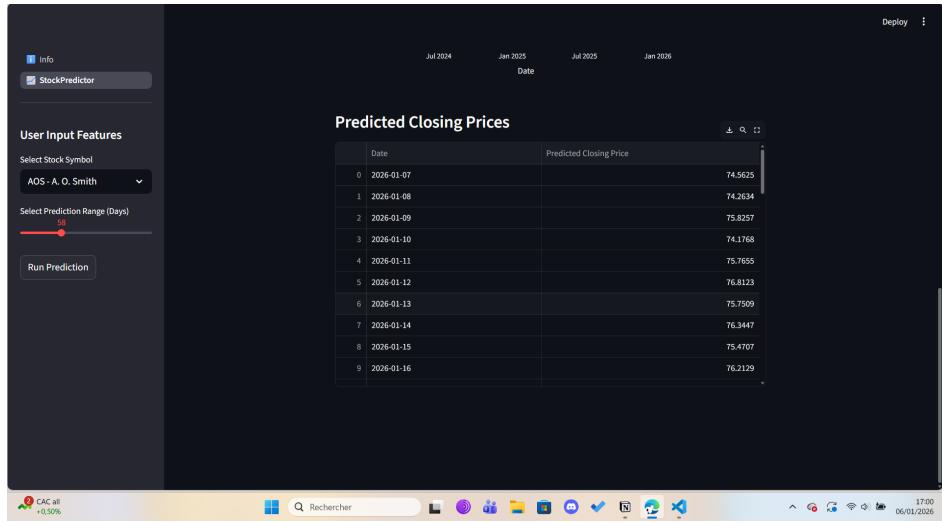
### 8.2 Sélection et visualisation des prédictions



figureInterface

de sélection du ticker et graphique de prédition

## 8.3 Export et résultats



figureAffichage

des résultats et options d'export en CSV

## 9 Bilan

### 9.1 Conclusion

Le projet a permis de développer une application fonctionnelle démontrant l'application du Machine Learning aux données financières. Tous les objectifs principaux ont été atteints.

### 9.2 Perspectives

- Intégrer des modèles plus avancés (LSTM, Prophet).
- Ajouter des indicateurs techniques (RSI, MACD).
- Déploiement en ligne.
- Analyse de sentiment à partir de news.

## 10 Bibliographie

### Références

- [1] Yahoo Finance API documentation, <https://pypi.org/project/yfinance/>
- [2] Statsmodels documentation, AutoReg model, [https://www.statsmodels.org/stable/generated/statsmodels.tsa.ar\\_model.AutoReg.html](https://www.statsmodels.org/stable/generated/statsmodels.tsa.ar_model.AutoReg.html)
- [3] Streamlit documentation, <https://docs.streamlit.io/>

## **11 Webographie**

### **Références**

- [1] Liste des tickers S&P 500, Wikipedia.
- [2] Plotly Python, <https://plotly.com/python/>

## 12 Annexes

### Annexe A : Cahier des charges

Description détaillée des fonctionnalités prévues (voir section “Travail réalisé”).

### Annexe B : Exemple d'exécution du projet

L'application est exécutée localement via la commande :

```
streamlit run streamlit_app/00_Info.py
```

L'utilisateur peut alors :

1. Accéder à la page d'information.
2. Sélectionner un ticker S&P 500 dans la liste déroulante.
3. Choisir l'horizon de prévision (5 à 180 jours).
4. Visualiser le graphique interactif montrant les données historiques et les prévisions.
5. Télécharger les résultats sous forme de fichier CSV.

### Annexe C : Manuel utilisateur

1. Cloner le dépôt GitHub.
2. Créer et activer un environnement virtuel : `python -m venv .venv`
3. Installer les dépendances : `pip install -r requirements.txt`
4. Se placer dans le dossier `streamlit_app`
5. Lancer l'application : `streamlit run 00_Info.py`
6. L'application s'ouvre dans le navigateur à l'adresse `http://localhost:8501`