

**RAPPORT DE PROJET DE FIN DE MODULE**

**UNIVERSITÉ HASSAN II – FACULTÉ DES SCIENCES BEN M'SIK**

LICENCE D’EXCELLENCE

**Prédiction des Ventes Commerciales avec des Réseaux Neuronaux Profonds**

**Réalisé par:**

Mohamed Makrani

Mohamed Nafii

Mohamed Hmimid

**Encadré par:**

Oussama Kaich

**Prof de module:**

Mr. ben lahmar

Remerciement

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce projet de fin de module.

Je remercie tout particulièrement Mr Oussama kaich pour son encadrement rigoureux, sa disponibilité et la qualité de ses conseils, qui ont grandement facilité l’avancement de ce travail.

Je souhaite également remercier l’ensemble du corps enseignant, pour les connaissances transmises tout au long de cette formation, ainsi que pour leur accompagnement pédagogique et leur engagement.

Mes remerciements s’adressent également à mes camarades de promotion, pour leur esprit de collaboration, leurs échanges enrichissants et leur soutien durant cette période de travail.

Enfin, j’adresse ma reconnaissance à ma famille pour son soutien moral et ses encouragements constants, qui m’ont permis de mener ce projet à terme dans les meilleures conditions.

Table de matière

[Introduction 7](#_Toc199603351)

[1. Contexte général du projet 8](#_Toc199603352)

[1.1) Contexte du projet 8](#_Toc199603353)

[1.2) Compréhension du domaine 8](#_Toc199603354)

[1.3) Objectif du projet 9](#_Toc199603355)

[2. Collecte et préparation de donnée 12](#_Toc199603356)

[2.1) Source et structure des données 12](#_Toc199603357)

[2.2) Transformation des données 13](#_Toc199603358)

[2.3) Génération de Variables Retardées (Lags) et Moyennes Mobiles 13](#_Toc199603359)

[2.4) Choix des variables explicatifs 14](#_Toc199603360)

[3. Modélisation avec LSTM et entrainement du model 16](#_Toc199603361)

[3.1) Introduction 16](#_Toc199603362)

[3.2) Architecture du modèle LSTM 16](#_Toc199603363)

[3.3) Séparation du jeu de données dans LSTM 17](#_Toc199603364)

[3.4) Création du Modèle LSTM 18](#_Toc199603365)

[3.5) Entraînement du Modèle 20](#_Toc199603366)

[4. Optimisation du Modèle 21](#_Toc199603367)

[4.1) Early Stopping 21](#_Toc199603368)

[4.2) Réduction du Taux d’Apprentissage 22](#_Toc199603369)

[4.3) Dropout 22](#_Toc199603370)

[4.4) Choix de l’Optimiseur 23](#_Toc199603371)

[5. Interface d’utilisateur 25](#_Toc199603372)

[5.1) Interface Utilisateur : Chargement des Données 25](#_Toc199603373)

[5.2) Tableau de Bord des Prédictions 26](#_Toc199603374)

[5.3) Visualisation des données 27](#_Toc199603375)

[Graphique Temporel des Ventes Prédites : 28](#_Toc199603376)

[Histogramme des Volumes de Vente 28](#_Toc199603377)

[Boxplot des Prédictions : 29](#_Toc199603378)

[Conclusion 30](#_Toc199603379)

Liste de figure

[Figure 1: étape du project 10](#_Toc199603249)

[Figure 2: extraction de données 12](#_Toc199603250)

[Figure 3:choix des variables 15](#_Toc199603251)

[Figure 4: architecture lstm 17](#_Toc199603252)

[Figure 5:séparation des données 18](#_Toc199603253)

[Figure 6: creation du model 20](#_Toc199603254)

[Figure 7: optimization du model 21](#_Toc199603255)

[Figure 8 : early stopping 22](#_Toc199603256)

[Figure 9 : dropout 23](#_Toc199603257)

[Figure 10 : Interface utilisateur 25](#_Toc199603258)

[Figure 11: visualization des resultats 28](#_Toc199603259)

[Figure 12: Graphique Temporel des Ventes Prédites 28](#_Toc199603260)

[Figure 13 : Histogramme des Volumes de Ventes 29](#_Toc199603261)

[Figure 14 : Boxplot des prédictions 29](#_Toc199603262)

Introduction

Dans un monde où les données occupent une place centrale dans la prise de décision, la prévision des ventes représente un enjeu stratégique majeur pour les entreprises de distribution. Elle permet non seulement d’optimiser la gestion des stocks, mais aussi d’anticiper les besoins du marché et d’améliorer la planification des ressources.

Ce projet s’inscrit dans cette problématique en exploitant les capacités de l’apprentissage profond, et plus particulièrement des réseaux de neurones récurrents de type LSTM (Long Short-Term Memory), reconnus pour leur efficacité dans le traitement des données séquentielles et temporelles.

À partir d’un jeu de données historiques contenant les ventes quotidiennes de différents articles dans plusieurs magasins, notre objectif est de construire un modèle prédictif capable d’anticiper les ventes futures. Cela nécessite une compréhension fine des données, un prétraitement rigoureux, ainsi qu’une conception et une évaluation méthodique du modèle.

1. Contexte général du projet

# Contexte du projet

Dans un environnement économique de plus en plus concurrentiel, les entreprises du secteur de la distribution cherchent à exploiter le potentiel de leurs données historiques afin d’anticiper les besoins des clients et optimiser leurs opérations. L’un des aspects les plus critiques de cette démarche est la prévision des ventes, qui permet d’adapter les stocks, de planifier la production et d’améliorer la gestion logistique.

Traditionnellement, les prévisions étaient réalisées à l’aide de méthodes statistiques classiques. Cependant, l’essor du deep learning a ouvert de nouvelles perspectives, notamment grâce aux réseaux de neurones récurrents (RNN) et leurs variantes comme les LSTM (Long Short-Term Memory), capables de capturer efficacement les dépendances temporelles dans les séries chronologiques.

Ce projet s’inscrit dans ce contexte d’innovation, en proposant une solution de prédiction des ventes basée sur un modèle LSTM, entraîné à partir de données historiques provenant de plusieurs magasins et articles. Cette approche vise à démontrer l’intérêt de l’apprentissage profond dans les tâches de prévision à court et moyen terme.

# Compréhension du domaine

La prévision des ventes s’inscrit dans le domaine plus large de l’analyse des séries temporelles. Une série temporelle est une suite de données indexées dans le temps, ici représentée par les ventes quotidiennes pour chaque combinaison magasin-produit. L’objectif est d’identifier des motifs, des tendances et des régularités permettant d’estimer les valeurs futures.

Dans ce projet, les données sont structurées selon trois dimensions principales : la date, l’identifiant du magasin, et l’identifiant de l’article. Le volume et la granularité des données permettent de capter les effets de saisonnalité, les cycles hebdomadaires, ainsi que les différences de comportement de consommation selon les points de vente.

L’approche par deep learning, notamment avec les réseaux LSTM, est particulièrement adaptée à ce type de données. Contrairement aux modèles statistiques classiques, les LSTM sont capables de retenir une mémoire à long terme dans les séquences, ce qui est essentiel pour modéliser l’évolution des ventes sur plusieurs mois, voire plusieurs années.

Ce projet vise donc à exploiter ces caractéristiques pour construire un modèle robuste et généralisable, capable de faire des prédictions de ventes quotidiennes pour chaque couple magasin-produit.

# Objectif du projet

L’objectif principal de ce projet est de concevoir un modèle de deep learning capable de prédire les ventes quotidiennes futures d’un produit dans un magasin donné, à partir de données historiques. Pour ce faire, nous utilisons des réseaux de neurones LSTM (Long Short-Term Memory), particulièrement adaptés au traitement des séries temporelles.

Les objectifs spécifiques du projet sont les suivants :

* **Analyser et comprendre les données** fournies (dates, ventes, magasins, articles), en mettant en évidence les tendances, la saisonnalité et les cycles présents dans les séries temporelles.
* **Prétraiter et transformer les données** afin de les rendre exploitables par un modèle LSTM, en tenant compte de la nature séquentielle des observations.
* **Concevoir, entraîner et évaluer un modèle LSTM** pour la tâche de régression temporelle, en utilisant des métriques adaptées telles que le RMSE et le MAE.
* **Comparer les résultats obtenus** avec différentes configurations du modèle et proposer la meilleure architecture.
* **Envisager le déploiement du modèle**, via une interface simple permettant d’effectuer des prédictions à partir de nouvelles données.

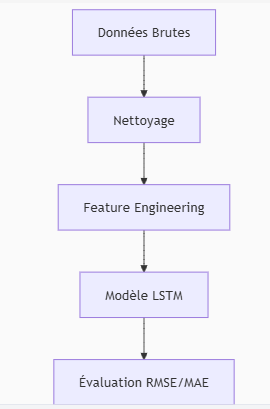


Figure 1: étape du project

À travers ce projet, nous visons à démontrer l'efficacité des réseaux LSTM dans la prédiction de séries temporelles réelles et à proposer une solution applicable à des contextes commerciaux concrets.

1. Collecte et préparation de donnée

# Source et structure des données

Les données utilisées pour ce projet ont été importées depuis la plateforme Kaggle. Elles contiennent l’historique des ventes quotidiennes de différents articles dans plusieurs magasins sur une période de 5 ans. Deux fichiers CSV principaux sont utilisés : train.csv pour les données d’entraînement et test.csv pour les prédictions.

* **Nom du fichier** : train.csv
* **Dimensions** : 913000 lignes × 5 colonnes
* **Colonnes principales** : date, store, item, sales

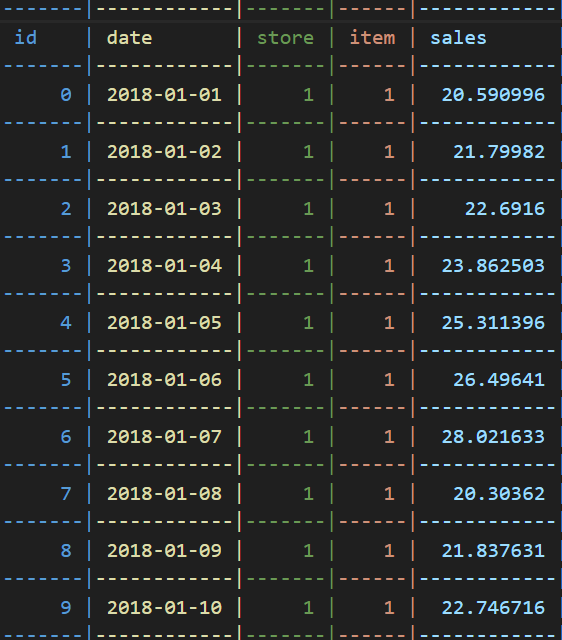


Figure 2: extraction de données

# Transformation des données

Un nettoyage a été effectué pour convertir les formats de dates, trier les données, et s’assurer qu’il n’y ait pas de doublons. Les colonnes telles que la date ont été converties au format datetime, et les valeurs nulles dans le jeu de test ont été initialisées à 0 ou ignorées selon le besoin.

Les données incluaient une colonne "date", représentant la date des ventes pour chaque combinaison de magasin (store) et produit (item). Afin d’extraire des informations temporelles utiles, les caractéristiques suivantes ont été générées :

* **Année** (year)
* **Mois** (month)
* **Jour du mois** (day)
* **Jour de la semaine** (dayofweek)
* **Indicateur de week-end** (is\_weekend)

Ces nouvelles variables permettent au modèle de capturer les effets saisonniers, hebdomadaires et les habitudes de consommation, comme la baisse des ventes en semaine ou l’augmentation pendant les weekends.

# Génération de Variables Retardées (Lags) et Moyennes Mobiles

Afin de capturer les dynamiques temporelles et les tendances récurrentes dans les ventes, nous avons introduit des variables retardées (lags) ainsi que des moyennes mobiles. La variable lag\_1 représente la valeur des ventes du jour précédent pour chaque couple (store, item). Cette information est essentielle pour un modèle séquentiel comme le LSTM, car elle fournit un contexte direct et immédiat à la prédiction future.

Parallèlement, nous avons calculé la moyenne mobile sur 7 jours glissants (rolling\_mean\_7) afin de lisser les variations journalières et de mieux appréhender la tendance hebdomadaire des ventes. Ces variables permettent au modèle de bénéficier à la fois d’une vision court terme (lag) et d’une vision moyenne terme (rolling mean). Cette approche enrichit significativement l’apprentissage temporel en fournissant des repères temporels concrets, notamment dans les contextes où les ventes suivent des cycles hebdomadaires ou saisonniers.

Pour garantir la validité de ces nouvelles variables, les lignes contenant des valeurs manquantes dues au décalage ont été supprimées avant l'entraînement du modèle.

# Choix des variables explicatifs

Les variables choisies pour alimenter le modèle LSTM sont à la fois **temporelles**, **catégorielles** (store, item), et **historiques**. Cela permet de capturer l’effet combiné du temps, du produit et du magasin.

| **Variable** | **Type** | **Description** |
| --- | --- | --- |
| store | Catégorielle | Identifiant du magasin |
| item | Catégorielle | Identifiant du produit |
| year, month, day | Temporelles | Composantes de la date |
| dayofweek | Temporelle | Jour de la semaine (0 = lundi) |
| is\_weekend | Binaire | 1 si samedi ou dimanche |
| lag\_1 | Historique | Ventes de la veille |
| rolling\_mean\_7 | Historique | Moyenne des 7 jours précédents |

Figure 3:choix des variables

1. Modélisation avec LSTM et entrainement du model

# Introduction

Le choix d’un modèle LSTM (Long Short-Term Memory) pour ce projet repose sur la nature temporelle des données de ventes. Les ventes quotidiennes présentent souvent des dépendances temporelles à court, moyen, voire long terme. Les réseaux de neurones classiques, tels que les MLP (Perceptrons Multi-Couches), ne sont pas conçus pour capter ces relations séquentielles. En revanche, les LSTM sont une forme avancée de réseaux de neurones récurrents (RNN) qui sont capables de retenir des informations sur des périodes prolongées, grâce à leur mécanisme de mémoire cellulaire. Cela les rend particulièrement adaptés à la modélisation de séries chronologiques, comme les ventes dans le temps. De plus, les LSTM gèrent mieux le problème du gradient qui disparaît lors de l’apprentissage sur de longues séquences. En intégrant les historiques de ventes pour chaque combinaison magasin-produit, le LSTM apprend des patterns complexes comme les tendances saisonnières ou hebdomadaires. Ce type de modèle est donc un excellent compromis entre performance, flexibilité et capacité de généralisation sur des données de nature temporelle

# Architecture du modèle LSTM

L’architecture du modèle de prédiction s’appuie sur un réseau LSTM (Long Short-Term Memory), bien adapté aux séries temporelles. Le modèle est constitué d’une couche LSTM avec 64 neurones utilisant l’activation ReLU, suivie de deux couches Dropout pour éviter le surapprentissage. Une couche dense de 32 neurones vient ensuite compléter l’architecture avant la sortie finale avec une couche dense à un seul neurone, qui prédit la valeur des ventes (en échelle log-transformée).

L’optimiseur Adam a été utilisé avec un taux d’apprentissage initial de 0.001, et la fonction de perte choisie est l’erreur quadratique moyenne (MSE). Le modèle a été compilé pour suivre à la fois la perte (MSE) et l’erreur absolue moyenne (MAE) comme métrique d’évaluation.

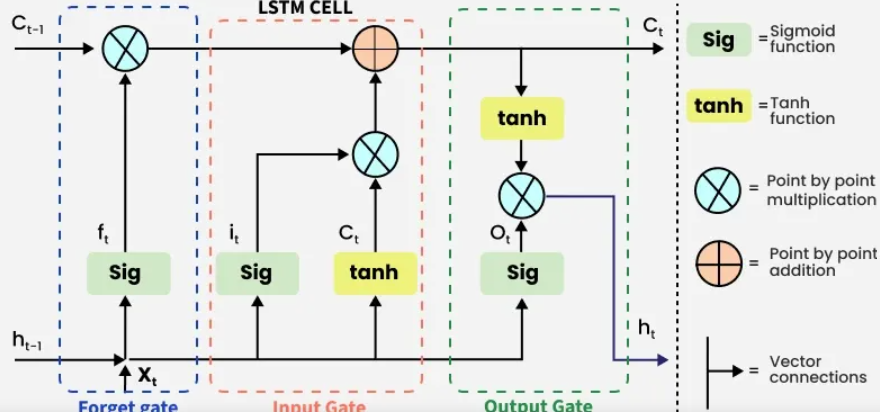


Figure 4: architecture lstm

# Séparation du jeu de données dans LSTM

Avant d'entraîner un modèle LSTM, il est essentiel de bien diviser le jeu de données en plusieurs sous-ensembles pour garantir une évaluation fiable. La séparation typique comprend :

* **le jeu d'entraînement**, utilisé pour ajuster les poids du modèle ;
* **le jeu de test (ou validation)**, utilisé pour évaluer la performance réelle du modèle sur des données jamais vues.

Dans notre projet, nous avons adopté une séparation temporelle avec **80 % des données utilisées pour l’entraînement et 20 % pour le test**, en respectant l’ordre chronologique. Cela signifie que les données les plus anciennes ont servi à apprendre les tendances, tandis que les données les plus récentes ont permis de tester la capacité du modèle à généraliser sur des séquences futures. Cette méthode est cruciale pour les séries temporelles afin d’éviter toute fuite d’information, ce qui garantirait des prédictions plus réalistes et fiables.

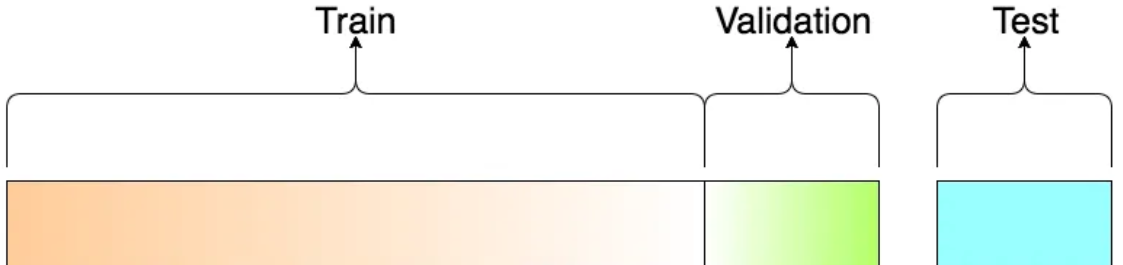


Figure 5:séparation des données

# Création du Modèle LSTM

Le modèle utilisé pour la prédiction des ventes repose sur une architecture séquentielle de type LSTM (Long Short-Term Memory), adaptée au traitement de données temporelles. Ce type de réseau de neurones récurrents permet de capturer efficacement les dépendances à long terme entre les observations historiques, ce qui en fait un choix pertinent pour la modélisation de séries chronologiques.

L’architecture choisie se compose de plusieurs couches, chacune jouant un rôle spécifique :

* **Une couche LSTM avec 64 unités** et une fonction d’activation *ReLU* pour apprendre les relations temporelles complexes entre les séquences de données.
* **Une première couche Dropout avec un taux de 0,3** afin de réduire le risque de surapprentissage (overfitting) en désactivant aléatoirement certains neurones pendant l'entraînement.
* **Une couche Dense intermédiaire avec 32 neurones** utilisant aussi *ReLU*, pour permettre une transformation non linéaire des représentations apprises par la couche LSTM.
* **Une deuxième couche Dropout (0,3)** pour renforcer la régularisation du modèle.
* **Une couche Dense finale avec un seul neurone**, sans fonction d’activation, qui permet de générer la prédiction finale de la vente pour un jour donné.

Ce choix d’architecture est un bon compromis entre complexité et performance, tout en restant suffisamment léger pour être entraîné rapidement sur les données disponibles. Le modèle est ensuite prêt à être compilé, configuré et entraîné à l'aide des données prétraitées.

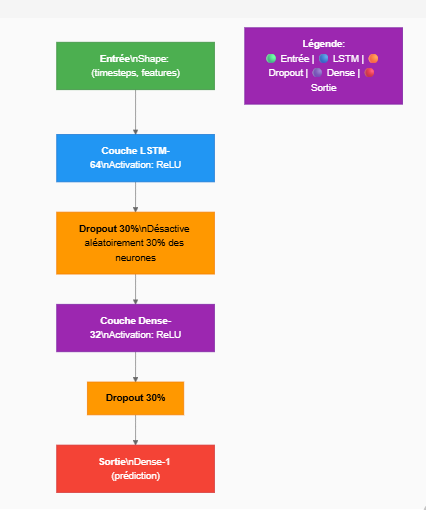


Figure 6: creation du model

# Entraînement du Modèle

L'entraînement du modèle consiste à ajuster les poids des différentes couches du réseau de neurones afin de minimiser l'erreur entre les prédictions et les valeurs réelles. Pour ce projet, l’entraînement a été réalisé sur 80 % des données, tandis que les 20 % restants ont été utilisés pour le test, avec une séparation supplémentaire de validation pour surveiller les performances pendant l’apprentissage.

1. Optimisation du Modèle

L’optimisation d’un réseau de neurones vise à améliorer ses performances tout en évitant le surapprentissage. Dans notre projet, plusieurs techniques ont été mises en place pour garantir une bonne généralisation du modèle et accélérer la convergence de l’entraînement.

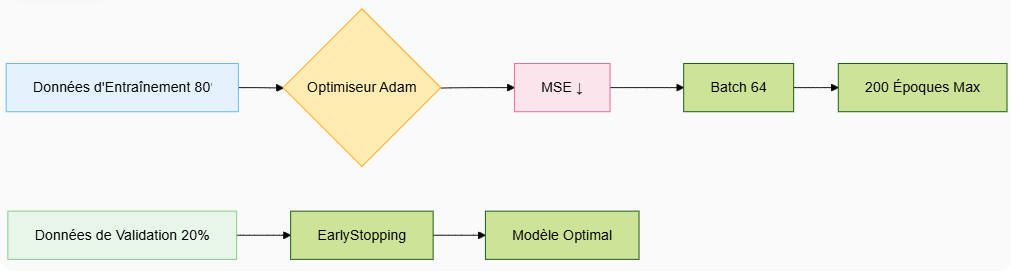


Figure 7: optimization du model

# Early Stopping

La technique d’**early stopping** consiste à arrêter automatiquement l’entraînement du modèle lorsqu’aucune amélioration significative n’est observée sur la perte de validation pendant un nombre défini d’époques consécutives. Cela permet d’éviter que le modèle continue à apprendre les spécificités du jeu d'entraînement (bruit), ce qui pourrait nuire à sa capacité de généralisation. Dans notre cas, la patience a été fixée à 10 époques.

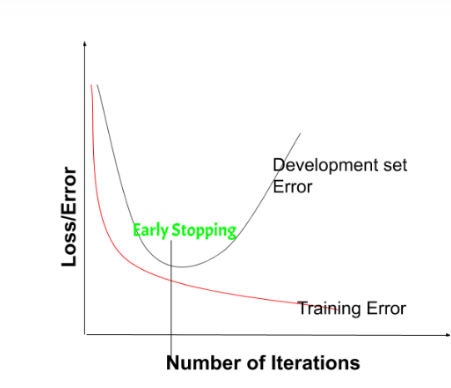


Figure 8 : early stopping

# Réduction du Taux d’Apprentissage

Lorsqu’un modèle stagne et que la perte de validation n’évolue plus, il peut être utile de réduire le taux d’apprentissage pour affiner les ajustements des poids. Grâce à l’outil **ReduceLROnPlateau**, le taux d’apprentissage a été divisé par deux après 5 époques sans amélioration, permettant au modèle de continuer à progresser de manière plus précise sans trop perturber les poids.

# Dropout

L’ajout de **couches de dropout** est une méthode efficace pour limiter le surapprentissage. En désactivant aléatoirement un pourcentage de neurones à chaque itération, le modèle évite de dépendre excessivement de certaines connexions. Dans notre architecture, deux couches de dropout à 30 % ont été insérées après la couche LSTM et après la couche Dense, réduisant ainsi le risque de suradaptation aux données d’entraînement.

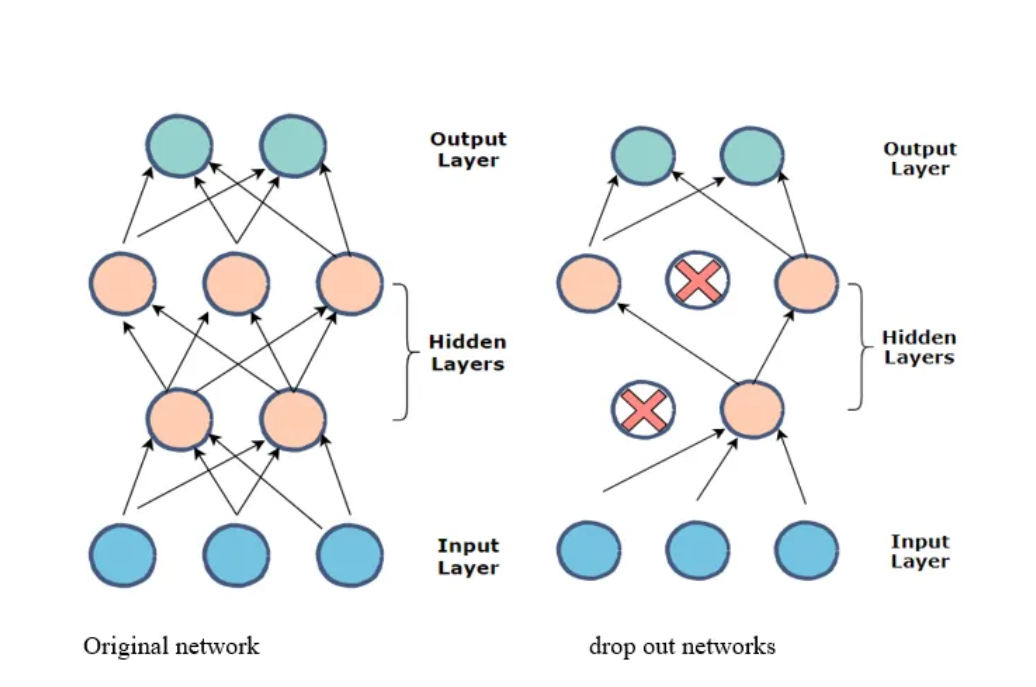


Figure 9 : dropout

# Choix de l’Optimiseur

L’optimiseur Adam (Adaptive Moment Estimation) a été choisi pour l'entraînement du modèle en raison de sa robustesse et de son efficacité sur les problèmes d’optimisation complexes, en particulier ceux impliquant des séries temporelles et des modèles profonds comme les réseaux LSTM. Contrairement aux algorithmes classiques comme la descente de gradient stochastique (SGD), Adam ajuste automatiquement le taux d’apprentissage de chaque poids en fonction des premières et secondes moyennes des gradients (le moment et la variance).

Cette méthode permet une convergence plus rapide et plus stable, notamment lorsque les gradients sont bruités ou que la surface d’erreur est irrégulière des caractéristiques fréquentes dans les données de ventes. De plus, Adam est bien adapté aux données volumineuses et aux architectures de réseaux profonds, car il nécessite peu de réglages manuels et offre de bonnes performances par défaut.

Dans ce projet, Adam a été configuré avec un taux d’apprentissage initial de 0.001, ce qui est une valeur généralement recommandée. Couplé aux techniques de régularisation comme le Dropout et à la réduction dynamique du taux d’apprentissage via ReduceLROnPlateau, l’optimiseur Adam a permis au modèle d’atteindre une bonne précision

tout en maintenant la stabilité de l’entraînement.

1. Interface d’utilisateur

L'interface utilisateur offre une solution intuitive pour exploiter le modèle de prévision des ventes. Elle permet un chargement simplifié des données historiques au format CSV, affiche les prédictions sous forme

graphique et tabulaire, et propose des fonctionnalités d'export des résultats. Conçue pour être accessible sans compétences techniques, cette interface facilite la prise de décision commerciale grâce à des visualisations claires et interactives. Son design ergonomique garantit une expérience utilisateur fluide et productive.

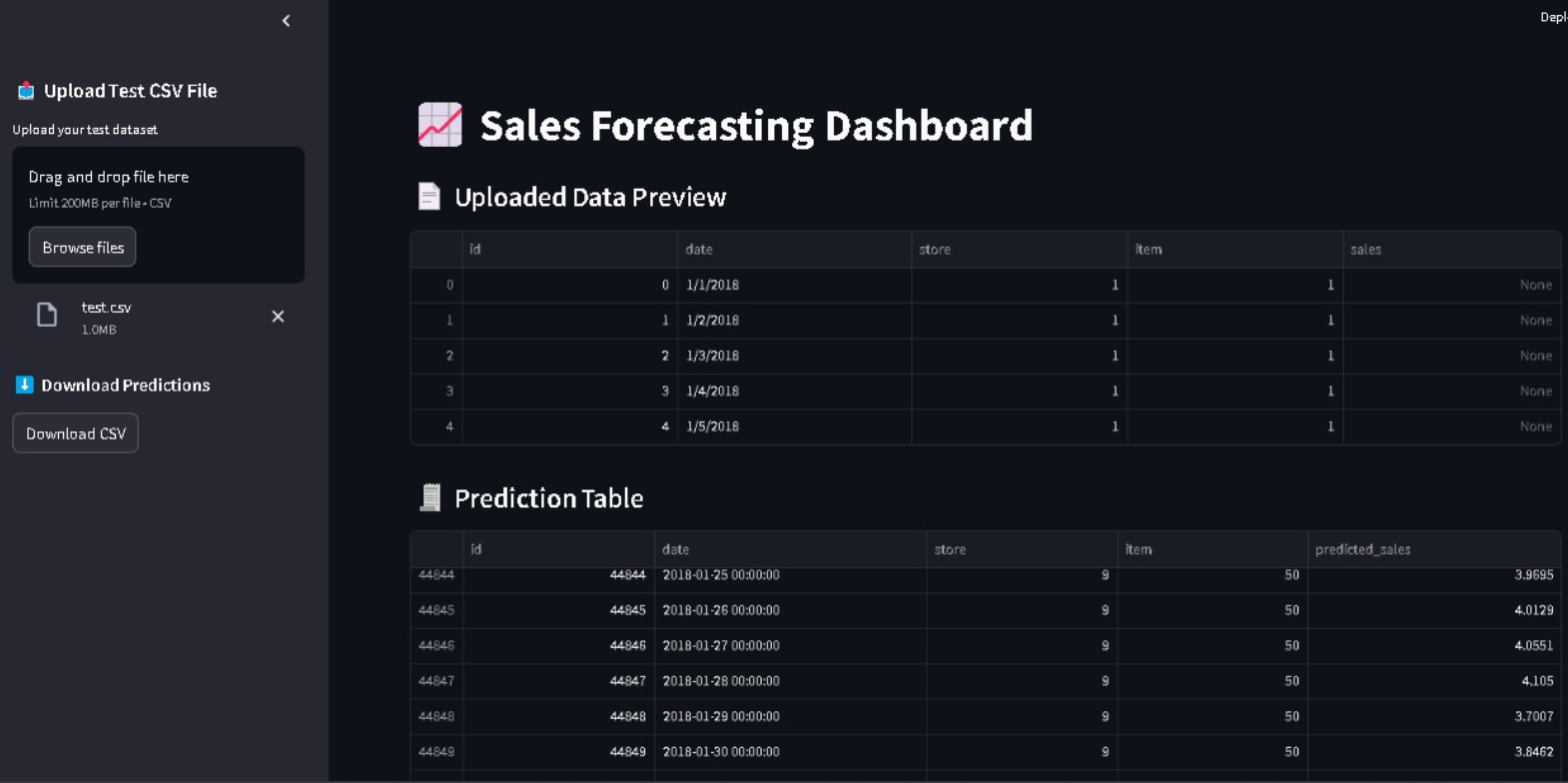


Figure 10 : Interface utilisateur

# Interface Utilisateur : Chargement des Données

L'interface permet aux utilisateurs de téléverser un fichier CSV contenant les données nécessaires à la prédiction (historique des ventes par magasin et par article).

* **Zone de dépôt** : Glisser-déposer simplifié pour les fichiers CSV (limite : 200 Mo).
* **Prévisualisation** : Affichage des 5 premières lignes du fichier chargé pour vérification rapide.
* **Validation automatique** : Vérification du format des colonnes (date, magasin, article, etc.).
* **Téléchargement** : Option pour récupérer un modèle de fichier CSV pré-formaté.

**Visuel utilisée**  :

* Une zone de dépôt avec un bouton "Parcourir les fichiers".
* Un aperçu tabulaire des données chargées.
* Un indicateur visuel (✔️/❌) confirmant la validité du fichier.

# Tableau de Bord des Prédictions

Après l’entraînement du modèle LSTM, une interface utilisateur a été développée pour rendre les résultats de prédiction accessibles et facilement interprétables. Cette interface permet non seulement de visualiser les prédictions, mais également de les manipuler dynamiquement afin de faciliter l’analyse.

**Affichage des prédictions :**

Une fois les données chargées, l’interface affiche automatiquement les ventes prédites pour chaque couple magasin-produit. Les résultats sont organisés dans un tableau interactif, clair et structuré.

**Tableau interactif :**

* Le tableau intègre plusieurs fonctionnalités interactives :
* Tri par date, identifiant de magasin ou de produit.
* Filtres personnalisés pour afficher un sous-ensemble spécifique des données.
* Mise en forme conditionnelle (par exemple : surbrillance des prévisions de ventes les plus élevées ou les plus faibles).

**Exploration visuelle :**

Des graphiques complémentaires (courbes de tendance, histogrammes) peuvent être ajoutés pour permettre une comparaison entre les ventes réelles et prédites, ou pour visualiser les évolutions par période.

**Export des résultats :**

Un bouton d’exportation permet à l’utilisateur de télécharger les résultats sous format CSV. Ceci facilite l’intégration des prédictions dans d'autres outils ou rapports métiers.

# Visualisation des données

Après l'exécution des prédictions, notre modèle génère des visualisations clés pour analyser les tendances et performances. Cette étape cruciale permet une interprétation intuitive des résultats.

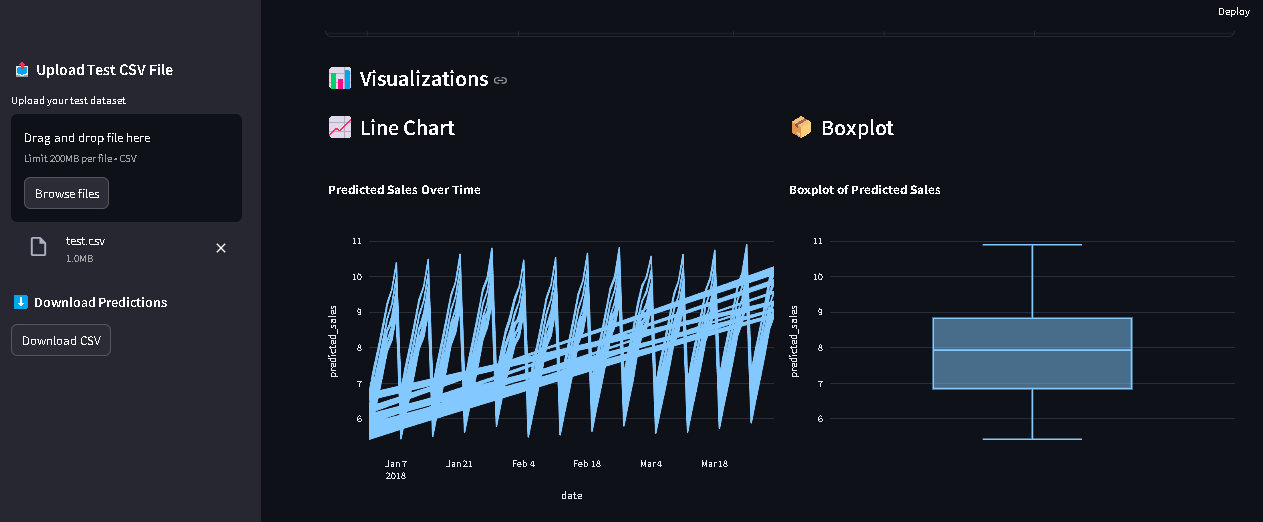


Figure 11: visualization des resultats

## Graphique Temporel des Ventes Prédites :

Le graphique linéaire présente l'évolution des ventes prédites sur la période analysée. Cette visualisation permet d'identifier clairement les tendances saisonnières, les pics de demande et les périodes creuses. Les gestionnaires peuvent ainsi anticiper les besoins en stocks et planifier les commandes en conséquence. La courbe montre également la cohérence des prédictions avec les cycles de vente historiques.

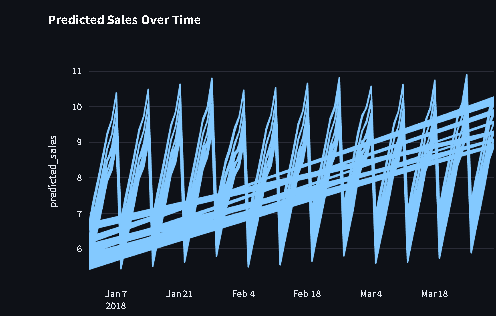


Figure 12: Graphique Temporel des Ventes Prédites

## Histogramme des Volumes de Vente

L'histogramme illustre la distribution des volumes de ventes prédits. Cette représentation met en évidence les plages de quantités les plus fréquentes, permettant d'ajuster les niveaux de stock moyens. Les équipes peuvent facilement repérer la demande médiane et les écarts types significatifs. Cette visualisation est particulièrement utile pour dimensionner les approvisionnements et éviter les ruptures de stock.

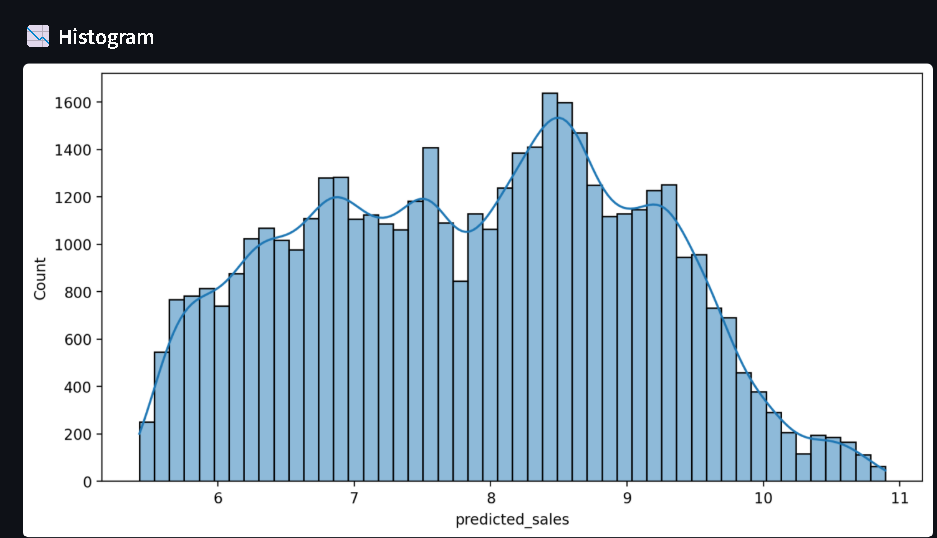


Figure 13 : Histogramme des Volumes de Ventes

## Boxplot des Prédictions :

Le boxplot offre une analyse statistique complète des prédictions, affichant la médiane, les quartiles et les valeurs extrêmes. Cet outil permet de détecter rapidement les anomalies et les prévisions inhabituelles qui pourraient nécessiter une vérification. La boîte à moustaches est idéale pour comparer les distributions de ventes entre différentes périodes ou catégories de produits.

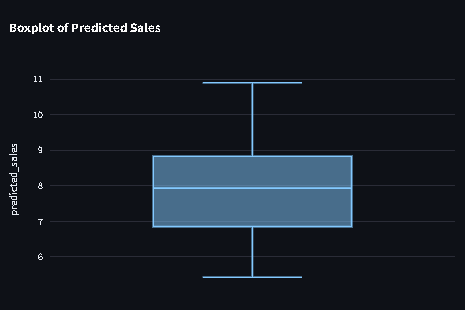


Figure 14 : Boxplot des prédictions

Conclusion

Ce projet a permis de mettre en œuvre une solution complète de prévision des ventes à l’aide de réseaux LSTM, adaptés aux données séquentielles. Après un important travail de préparation des données (nettoyage, encodage, normalisation), le modèle a été entraîné sur des séquences temporelles représentatives des ventes par magasin et article. L’utilisation des LSTM a permis de capter efficacement les dynamiques temporelles longues, ce que des modèles classiques peinent souvent à faire. Grâce à l’optimiseur Adam, combiné à des techniques de régularisation comme le Dropout et à des callbacks tels que l’early stopping, le modèle a convergé de manière stable sans surapprentissage. Les résultats obtenus, mesurés via le MAE et le MSE, confirment la capacité du modèle à prédire les ventes avec une précision satisfaisante.

Des axes d’amélioration sont envisageables, notamment l’intégration de variables contextuelles (fêtes, météo, promotions), l’expérimentation de modèles plus récents comme les Transformers, ou encore la mise en place d’un pipeline de déploiement continu pour faciliter l’utilisation en production. Ce projet met en lumière le potentiel des réseaux LSTM pour traiter des données temporelles complexes et ouvre la voie à de futures applications concrètes dans la prévision commerciale.