







Prédiction du Prix de l'Immobilier avec Machine Learning



Description

Application web d'intelligence artificielle permettant de prédire le prix d'une maison en fonction de ses caractéristiques. Le projet utilise des algorithmes de Machine Learning avancés (XGBoost, Random Forest, Ridge Regression) entraînés sur le célèbre dataset Kaggle "House Prices".

✨ Fonctionnalités

-  Prédiction en temps réel du prix d'une maison
-  Analyse comparative de 5 modèles de Machine Learning
-  Interface web moderne et intuitive avec animations
-  Visualisation des performances et de l'importance des features
-  Génération de fichiers de soumission pour Kaggle
-  Précision de prédiction > 90% (R^2 Score)

Démonstration

Application web en ligne : [Voir la démo](#)



Structure du Projet

ProjetImmobilier/

dataset/	# Données brutes
— train.csv	
— test.csv	
— processed/	# Données prétraitées
— X_train.csv	
— X_val.csv	
— y_train.csv	
— ...	
models/	# Modèles entraînés
— xgboost.pkl	
— random_forest.pkl	
— ridge_regression.pkl	
— model_comparison.csv	
submissions/	# Fichiers de soumission Kaggle
— submission_xgboost.csv	
— submission_ensemble.csv	
— ...	
01_exploration.py	# Exploration des données
02_preprocessing.py	# Prétraitement
03_train_models.py	# Entraînement des modèles
04_predict.py	# Génération des prédictions
app.py	# Application Streamlit
app_flask.py	# Application Flask
prediction_app.html	# Application web standalone
README.md	# Documentation

Installation

Prérequis

- Python 3.8 ou supérieur
- Anaconda (recommandé)

Étapes d'installation

1. Cloner le repository

git clone <https://github.com/votre-username/house-price-predictor.git>

cd house-price-predictor

2. Installer les dépendances

pip install pandas numpy scikit-learn xgboost matplotlib seaborn streamlit flask

3. Télécharger le dataset

- Allez sur [Kaggle House Prices Competition](#)
 - Téléchargez `train.csv` et `test.csv`
 - Placez-les dans le dossier `dataset/`
-



Utilisation

1. Exploration des données

python 01_exploration.py

Génère :

- Statistiques descriptives
- Visualisations de la distribution des prix
- Analyse des valeurs manquantes
- Matrice de corrélation

2. Prétraitement

python 02_preprocessing.py

Effectue :

- Gestion des valeurs manquantes
- Création de nouvelles features (feature engineering)
- Encodage des variables catégorielles
- Séparation train/validation

3. Entraînement des modèles

python 03_train_models.py

Entraîne 5 modèles :

- Régression Linéaire
- Ridge Regression
- Lasso Regression
- Random Forest
- **XGBoost** (meilleur modèle)

4. Génération des prédictions

```
python 04_predict.py
```

Crée des fichiers de soumission pour Kaggle.

5. Application web

Option A : Streamlit

```
streamlit run app.py
```

Option B : Flask

```
python app_flask.py
```

Option C : HTML Standalone

- Ouvrez [prediction_app.html](#) directement dans votre navigateur



Résultats

Performance des modèles

Modèle	R ² Score	RMSE	MAE
XGBoost	0.9156	0.1198	\$15,234
Random Forest	0.8892	0.1356	\$18,567
Ridge Regression	0.8845	0.1389	\$19,123





Lasso Regression	0.8823	0.1401	\$19,456
Régression Linéaire	0.8712	0.1467	\$20,789

Features les plus importantes (XGBoost)

1. **OverallQual** - Qualité générale
 2. **GrLivArea** - Surface habitable
 3. **TotalBsmtSF** - Surface du sous-sol
 4. **GarageCars** - Capacité du garage
 5. **YearBuilt** - Année de construction
-

Interface Web

L'application web propose :

-  **Prédiction Simple** : Interface intuitive avec sliders
 -  **Prédiction Avancée** : Upload de fichiers CSV
 -  **Analyse du Modèle** : Visualisation des performances
 -  **Design Moderne** : Animations fluides et glassmorphism
-

Méthodologie

1. Feature Engineering

Nouvelles features créées :

- **TotalSF** : Surface totale de la maison
- **TotalBath** : Nombre total de salles de bain
- **HouseAge** : Âge de la maison
- **IsRemodeled** : Indicateur de rénovation
- **QualityScore** : Score combiné de qualité

2. Preprocessing

- Imputation des valeurs manquantes (stratégie adaptée par type)
- Encodage des variables catégorielles (Label Encoding)

- Transformation logarithmique de la variable cible
- Suppression des valeurs aberrantes

3. Validation

- Split 80/20 pour train/validation
 - Cross-validation pour éviter l'overfitting
 - Métriques multiples : RMSE, MAE, R^2 , MAPE
-



Technologies Utilisées

- **Python 3.8+**
 - **scikit-learn** - Modèles de Machine Learning
 - **XGBoost** - Gradient Boosting
 - **Pandas & NumPy** - Manipulation de données
 - **Matplotlib & Seaborn** - Visualisation
 - **Streamlit** - Application web interactive
 - **Flask** - Framework web
 - **HTML/CSS/JavaScript** - Interface standalone
-



Contribution

Les contributions sont les bienvenues ! N'hésitez pas à :

1. Fork le projet
 2. Créer une branche (`git checkout -b feature/amelioration`)
 3. Commit vos changements (`git commit -m 'Ajout d'une fonctionnalité'`)
 4. Push vers la branche (`git push origin feature/amelioration`)
 5. Ouvrir une Pull Request
-



TODO / Améliorations futures

- [] Ajouter plus de modèles (LightGBM, CatBoost)
- [] Implémenter un système de stacking
- [] Ajouter des graphiques interactifs (Plotly)
- [] Optimisation des hyperparamètres avec GridSearchCV
- [] API REST pour les prédictions

- [] Mode sombre pour l'interface
 - [] Support multilingue
 - [] Tests unitaires
-



Auteur

Votre Nom

- GitHub: [@Ibrahima1012](#)
 - LinkedIn: [Votre Profil](#)
 - Email: IBA941500@gmail.com
-



Licence

Ce projet est sous licence MIT. Voir le fichier [LICENSE](#) pour plus de détails.



Remerciements

- Dataset fourni par [Kaggle House Prices Competition](#)
 - Inspiration des notebooks Kaggle de la communauté
 - Documentation officielle de scikit-learn et XGBoost
-



Support

Si vous avez des questions ou des problèmes :

- ✉ Email : votre.email@example.com
 - 💬 Issues GitHub : [Créer une issue](#)
-

★ Si ce projet vous a aidé, n'hésitez pas à lui donner une étoile sur GitHub !

Made with ❤️ and Python

