PREDICTING HOUSE PRICES USING MACHINE LEARNING

ABSTRACT

Real estate is the least transparent industry in our ecosystem. Housing prices keep changing day in and day out and sometimes are hyped rather than being based on valuation. Predicting housing prices with real factors is the main crux of our research project. Here we aim to make our evaluations based on every basic parameter that is considered while determining the price. We use various regression techniques in this pathway, and our results are not sole determination of one technique rather it is the weighted mean of various techniques to give most accurate results.

Predicting House Prices

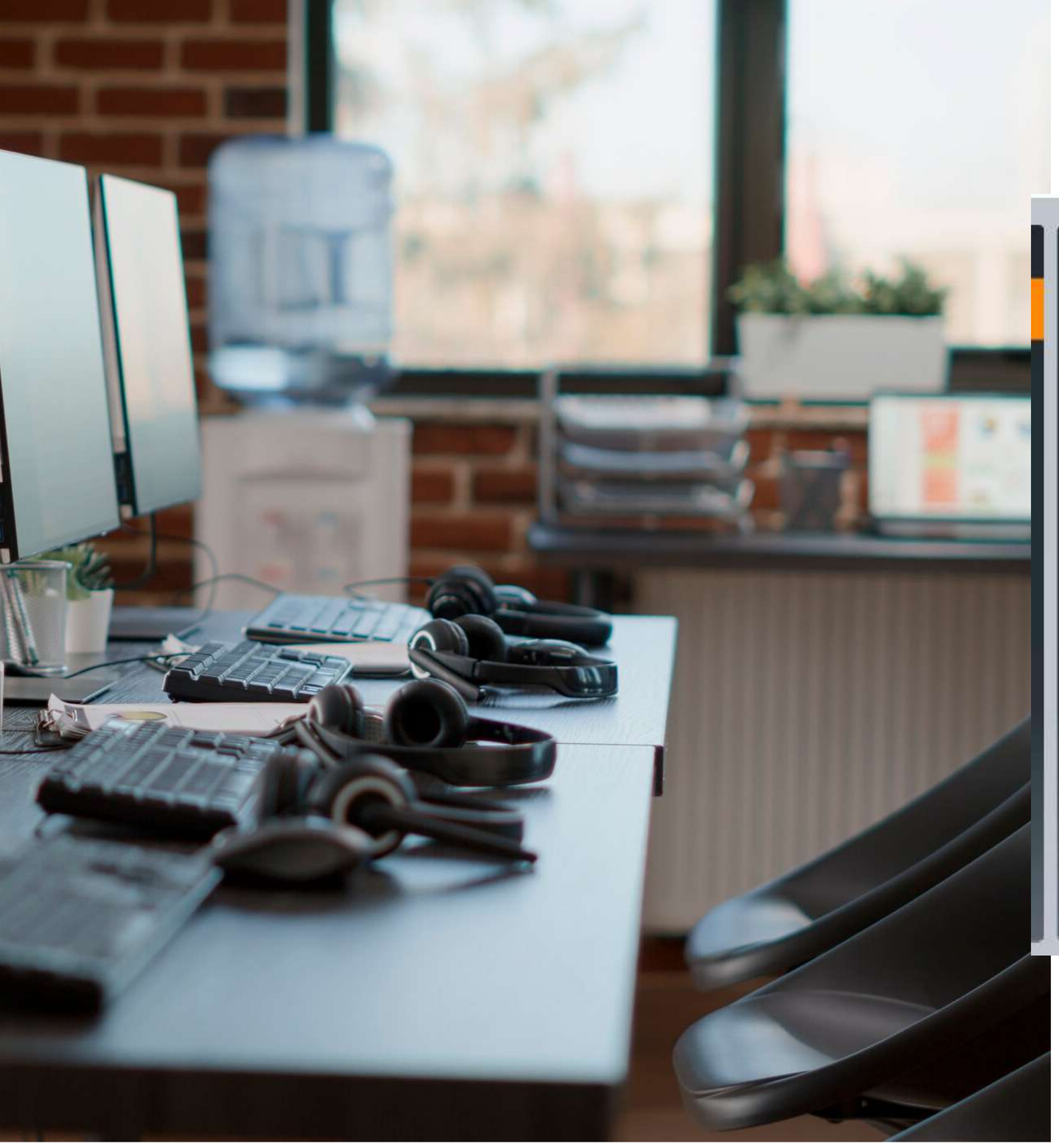
Now comes the exciting part! By feeding new data into the trained machine learning model, we can predict house prices with remarkable accuracy. This empowers stakeholders to make informed decisions when buying, selling, or investing in real estate, ultimately revolutionizing the industry.

Machine Learning Basics

Machine learning is a subset of *artificial intelligence* that enables computers to learn and make predictions without being explicitly programmed. By analyzing vast amounts of *data*, machine learning algorithms can identify patterns and make accurate predictions. This technology has immense potential in predicting house prices.

Understanding House Prices

Before delving into the power of *machine learning*, let's understand the factors influencing house prices. Factors such as *location*, *size*, *amenities*, and *market demand* play a crucial role. Accurately predicting house prices can empower buyers, sellers, and investors to make informed decisions.



EXISTING SYSTEM

Data is at the heart of technical innovations, achieving any result is now possible using predictive models. Machine learning is extensively used in this approach. Although development of correct and accurate model is needed. Previous algorithms like SVM are not suitable for complex real-world data.

PROPOSED SYSTEM

- Our main focus here is to develop a model which predicts the property cost based on vast datasets using data mining approach.
- The data mining process in a real estate industry provides an advantage to the developers by processing the data, forecasting future trends and thus assisting them to make favourable knowledge-driven decisions.
- Our dataset comprises of various essential parameters and data mining has been at the root of our system.
- We will initially clean up our entire dataset and also truncate the outlier values.
- Finally a system will be made to predict the Real Estate Prices using Deep Learning Approach.

FEASIBILITY STUDY

Basic areas to be covered are –

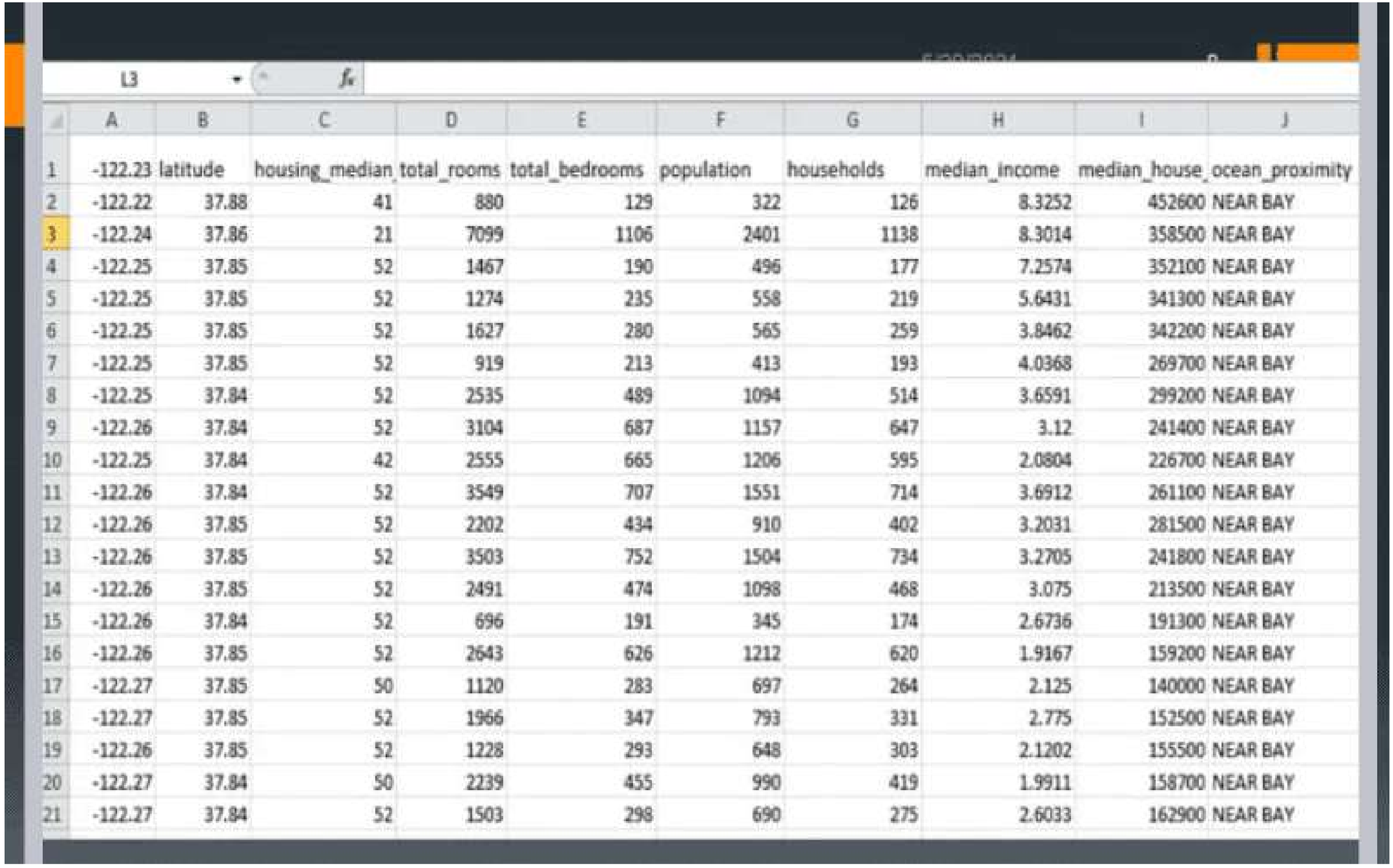
- Operational Feasibility: The applying can scale back the time consumed to take care of manual records and isn't dull and cumbersome to take care of the records, therefore operational practicableness is assured.
- Technical Feasibility: Minimum hardware requirements: 1.66 GHz Pentium Processor or Intel compatible processor. 1 GB RAM net property, eighty MB disk area.

- Economical Feasibility :Once the hardware and software package needs get consummated, there's no want for the user of our system to pay for any further overhead. For the user, the applying are economically possible within the following aspects: the applying can scale back tons of labor work. therefore the efforts are reduced. Our application can scale back the time that's wasted in manual processes. The storage and handling issues of the registers are resolved .

SYSTEM SPECIFICATIONS

Since this is an AI based Deep Learning approach, we are going to use a dataset extracting the features of houses and algorithmic models to train the dataset for predictions.

- Dataset- We are using an unsupervised dataset,i.e, data without class labels. The objective of this unsupervised technique, is to find patterns in data based on the relationship between data points themselves.



L3

	A	B	C	D	E	F	G	H	I	J
1	-122.23	latitude	housing_median	total_rooms	total_bedrooms	population	households	median_income	median_house	ocean_proximity
2	-122.22	37.88	41	880	129	322	126	8.3252	452600	NEAR BAY
3	-122.24	37.86	21	7099	1106	2401	1138	8.3014	358500	NEAR BAY
4	-122.25	37.85	52	1467	190	496	177	7.2574	352100	NEAR BAY
5	-122.25	37.85	52	1274	235	558	219	5.6431	341300	NEAR BAY
6	-122.25	37.85	52	1627	280	565	259	3.8462	342200	NEAR BAY
7	-122.25	37.85	52	919	213	413	193	4.0368	269700	NEAR BAY
8	-122.25	37.84	52	2535	489	1094	514	3.6591	299200	NEAR BAY
9	-122.26	37.84	52	3104	687	1157	647	3.12	241400	NEAR BAY
10	-122.25	37.84	42	2555	665	1206	595	2.0804	226700	NEAR BAY
11	-122.26	37.84	52	3549	707	1551	714	3.6912	261100	NEAR BAY
12	-122.26	37.85	52	2202	434	910	402	3.2031	281500	NEAR BAY
13	-122.26	37.85	52	3503	752	1504	734	3.2705	241800	NEAR BAY
14	-122.26	37.85	52	2491	474	1098	468	3.075	213500	NEAR BAY
15	-122.26	37.84	52	696	191	345	174	2.6736	191300	NEAR BAY
16	-122.26	37.85	52	2643	626	1212	620	1.9167	159200	NEAR BAY
17	-122.27	37.85	50	1120	283	697	264	2.125	140000	NEAR BAY
18	-122.27	37.85	52	1966	347	793	331	2.775	152500	NEAR BAY
19	-122.26	37.85	52	1228	293	648	303	2.1202	155500	NEAR BAY
20	-122.27	37.84	50	2239	455	990	419	1.9911	158700	NEAR BAY
21	-122.27	37.84	52	1503	298	690	275	2.6033	162900	NEAR BAY

- The dataset consists of more than 20,000 records.
- Around nine features are provided in the dataset like latitude, total rooms, total bedrooms etc.
- Data mining technique is used for processing of the dataset to extract relevant data to be further used in data models.

- Data model-The basic technique used for the model is Regression Analysis which is a statistical method to model the relationship between a dependent (target) and independent (predictor) variables with one or more independent variables. It predicts continuous/real values.

5/29/2021

- For our application we are using three types of algorithm train the model, which are-
- **KN-Neighbors**: K nearest neighbors is a simple algorith that stores all available cases and predict the numerical target based on a similarity measure (e.g., distance functions)
- **Support Vector Machine**: The goal of the SVM algorith to create the best line or decision boundary that can segregate n-dimensional space into classes so that we easily put the new data point in the correct category in t future.
- **Artificial Neural Network(ANN)**: It's a computational m in which artificial neurons are nodes, and directed edge weights are connections between neuron outputs and n

HARDWARE AND SOFTWARE REQUIREMENTS

SOFTWARE REQUIREMENTS-

- Python version- 3.8
- Libraries-

✓Pandas: library for manipulating data in numerical tables.

✓scikit-learn: machine learning library that features various classification, regression and clustering algorithms.

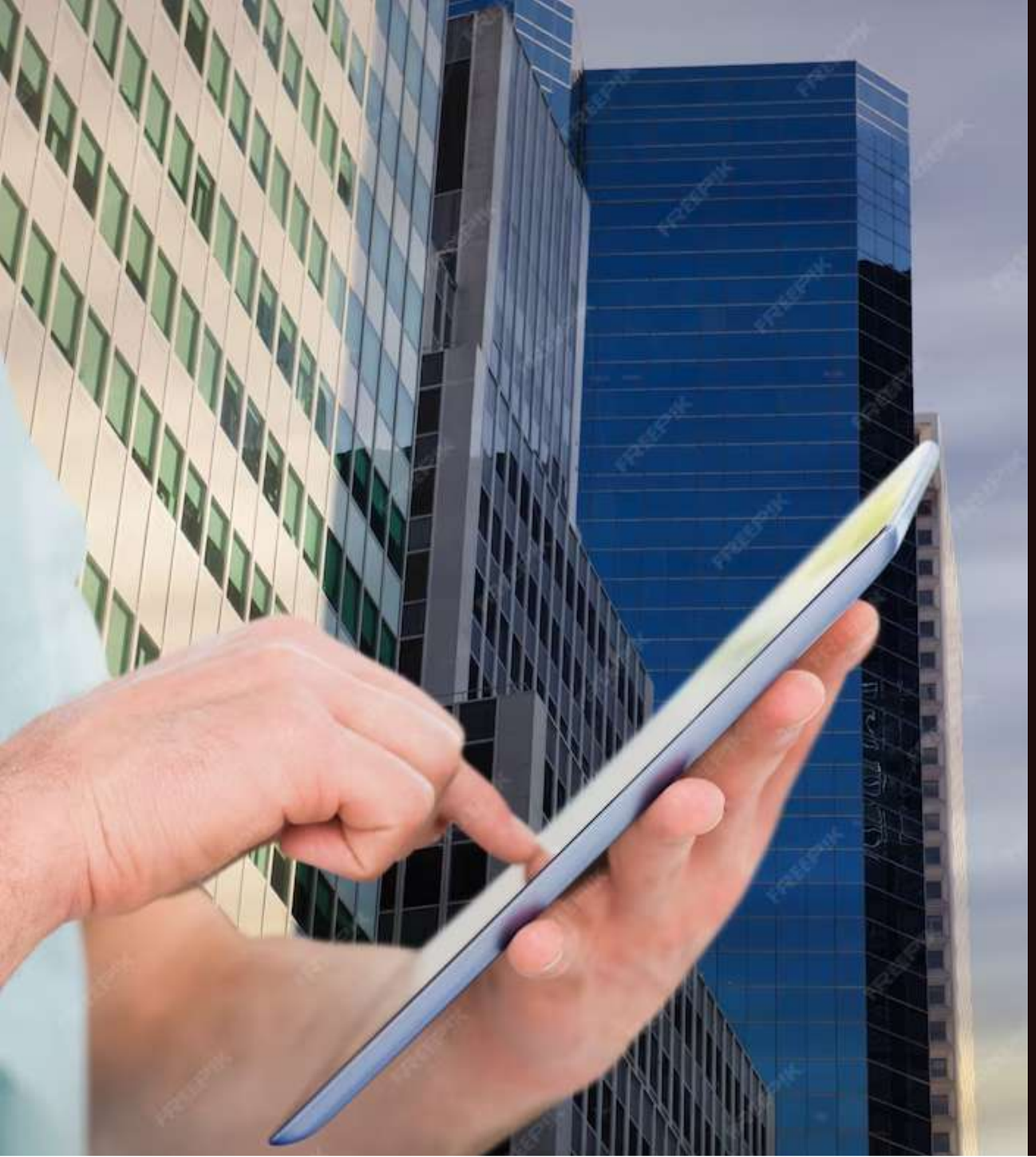
✓Matplotlib: Matplotlib is a plotting library for creating static, animated, and interactive visualizations in python.

5/29/2021

- ✓Seaborn: library based on matplotlib. It provides a h
  interface for drawing attractive and informative statis
  graphics
- ✓Tensorflow: The core open source library to help yo
  and train ML models.
- ✓Jupyterlab: JupyterLab is a web-based interactive
  development environment for Jupyter notebooks, co
  data.

HARDWARE REQUIREMENTS-

- ✓1.66 GHz Pentium Processor or Intel compatible processor.
- ✓1 GB RAM
- ✓ 80 MB disk area.



Benefits of Predictive Models

Predictive models powered by machine learning offer numerous benefits. They enable buyers to estimate fair prices, sellers to set competitive prices, and investors to identify lucrative opportunities. By reducing uncertainty and providing data-driven insights, these models enhance decision-making and improve market efficiency.

Conclusion

In conclusion, *machine learning* has the potential to revolutionize the real estate industry by accurately predicting house prices. By leveraging comprehensive datasets, training models, and evaluating their performance, stakeholders can make informed decisions. Embracing this technology will enhance market efficiency and empower buyers, sellers, and investors.


Thank you

Gradient Boosting with scikit-learn:

pythonCopy code

from sklearn.ensemble import GradientBoostingRegressor from sklearn.model_selection import train_test_split from sklearn.metrics import mean_squared_error # Load your dataset and split it into training and testing sets X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42) # Create a Gradient Boosting Regressor model gb_regressor = GradientBoostingRegressor(n_estimators=100, learning_rate=0.1, max_depth=3, random_state=42) # Fit the model to the training data gb_regressor.fit(X_train, y_train) # Make predictions on the test set y_pred = gb_regressor.predict(X_test) # Evaluate the model mse = mean_squared_error(y_test, y_pred) print(f"Mean Squared Error: {mse}")

XGBoost:

You'll need to install the xgboost library if you haven't already:

bashCopy code

pip install xgboost

Here's an example of using XGBoost for regression:

pythonCopy code

import xgboost as xgb from sklearn.metrics import mean_squared_error # Create a DMatrix from your data dtrain = xgb.DMatrix(X_train, label=y_train) dtest = xgb.DMatrix(X_test, label=y_test) # Set hyperparameters params = { 'objective': 'reg:squarederror', 'max_depth': 3, 'learning_rate': 0.1, 'n_estimators': 100, 'seed': 42 } # Train the XGBoost model xgboost_model = xgb.train(params, dtrain) # Make predictions on the test set y_pred = xgboost_model.predict(dtest) # Evaluate the model mse = mean_squared_error(y_test, y_pred) print(f"Mean Squared Error: {mse}")

These code examples demonstrate how to use Gradient Boosting and XGBoost for regression tasks. Make sure to adjust hyperparameters like n_estimators, max_depth, and learning_rate to fine-tune the model for your specific dataset and problem. Additionally, you can explore other gradient boosting libraries like LightGBM and CatBoost, which offer similar functionality and often perform well in different scenarios.