**Project title :­- Predicting house price using machine learning**

**Problem definition:**

**Let’s say we are a real estate agent, and we are in charge of selling a new house. We don’t know the price, and we want to infer it by comparing it with other houses. We look at features of the house which could influence the house, such as size, number of rooms, location, crime rate, school quality, distance to commerce, etc. At the end of the day, what we want is a formula on all these features which gives us the price of the house, or at least an estimate for it**

**Landscape**: Through data analysis, understand the characteristics of each house price, by

Cost of land, requirements and features

**Forecast Future Registration Trends**: Utilize advanced Artificial Intelligence techniques to build predictive models.

**Design Thinking**:

Step 1: **Data Collection:**

Data Source: Obtain a real dataset containing information about house price from different house holders and neighbour. This dataset should include attributes such as land space,features, class, category, registration date.

Example Dataset: A dataset of "houseprice.csv" with the relevant attributes.

Step 2: **Data Preprocessing:**

* **Data Cleaning**: Use tools like Python and Pandas to remove duplicates, handle missing values, and correct data inconsistencies.
* **Handling Missing Values**: Identify and handle missing values appropriately, either by imputing them or removing rows/columns with excessive missing data.
* **Categorical to Numerical**: Convert categorical features like Company Status, Class, and Category into numerical representations using techniques like one-hot encoding or label encoding.
* **Data Transformation:** Convert categorical features into numerical representations using techniques like one-hot encoding.

Step 3: **Exploratory Data Analysis (EDA):**

* **Statistical Summaries**: Provide basic statistics like mean, median, standard deviation, and quantiles for numerical features.
* **Data Visualization**: Use charts and graphs (histograms, box plots, scatter plots, etc.) to visually explore the distribution, relationships, and outliers within the data.
* **Pattern Identification**: Identify any interesting patterns or anomalies in the data that could inform the predictive models.
* **Tools**: Utilize Python libraries such as Pandas, Matplotlib, and Seaborn for EDA.

Step 4**: Feature Engineering:**

* Create new features or transform existing ones that could be valuable for predictive analysis. Some potential feature ideas include:
* Age of the company (calculated from registration date)
* Capital utilization ratio (Paid-up Capital / Authorized Capital)
* Time trends and seasonality indicators

Step 5**: Predictive Modelling:**

* **Data Splitting**: Split the dataset into training and testing sets to train and evaluate the predictive models.
* **Model Selection**: Choose appropriate AI algorithms for predictive modeling. Potential options include:

1. Regression models (e.g., Linear Regression, Random Forest Regression)
2. Time series forecasting models (e.g., ARIMA, Prophet)
3. Classification models if predicting categorical outcomes (e.g., Logistic

Regression Random Forest Classifier)

* **Hyperparameter Tuning**: Fine-tune model hyperparameters to optimize performance.
* **Training and Validation**: Train the selected models on the training data and validate them on the testing data.

Tools: Use Python's Scikit-Learn, Statsmodels for time series analysis, and potentially

TensorFlow or PyTorch for deep learning.

Step 6: **Model Evaluation:**

* **Performance Metrics**: Evaluate the predictive models using appropriate metrics depending on the problem type (regression or classification). Common metrics include accuracy, precision, recall, F1-score, Mean Absolute Error (MAE), Mean Squared Error (MSE), etc.
* **Cross-Validation**: Implement cross-validation techniques to ensure model robustness.
* **Visualization**: Visualize model predictions and compare them with actual data to understand model performance.

**Tools and Technologies**:

Python (Pandas, NumPy), Matplotlib, Seaborn (for data visualization), Scikit-Learn, Stats models, TensorFlow, PyTorch (for modelling), Jupyter Notebooks, Git.