**Project 1:**

Project 1:

Based on the scenario provided, design a machine learning design system, this has to be just a

high level design, that means just share on idea how will you handle data collection, processing,

model training, evaluation, and all the other things (you may refer the second session slides).

Make sure the answer is not more than 2 pages.

Scenario:

Imagine you are a machine learning engineer working for a real estate company that wants to

build a machine learning system to predict housing prices in a given area. Your goal is to create

a model that can accurately predict the price of a house based on various features such as

square footage, number of bedrooms and bathrooms, location, and other relevant factors.

Answer:

Here's a high-level overview of the steps we could take:

Data Collection: Gather a dataset containing information about past real estate transactions, including features such as square footage, number of bedrooms and bathrooms, location (e.g., ZIP code or coordinates), amenities, sale price, etc. We might also include data from other relevant sources like neighborhood demographics, crime rates, school quality, and local economic indicators.

Data Preprocessing: Clean the dataset by handling missing values, removing outliers, and encoding categorical variables. We may also need to normalize or scale numerical features to ensure they're on a similar scale.

Feature Engineering: Create new features or transform existing ones to better represent the underlying relationships in the data. For example, we could calculate the price per square foot, create binary indicators for certain amenities (like a pool or fireplace), or extract information from location data (such as distance to city center or proximity to parks).

Model Selection: Choose an appropriate machine learning algorithm for the task. Since we're dealing with regression (predicting a continuous variable), options might include linear regression, decision trees, random forests, gradient boosting, or neural networks. We'll likely want to experiment with multiple models to see which one performs best.

Model Training: Split the dataset into training and testing sets to evaluate model performance. We'll train the chosen model on the training data, tuning hyperparameters as needed to optimize performance. Techniques like cross-validation can help ensure robustness.

Model Evaluation: Evaluate the trained model using appropriate metrics for regression tasks, such as mean squared error (MSE), root mean squared error (RMSE), or mean absolute error (MAE). We'll compare the model's predictions against the actual sale prices in the test set to assess its accuracy and generalization ability.

Model Deployment: Once we're satisfied with the model's performance, we can deploy it to production where it can be used to predict housing prices for new listings. This might involve building a web application or integrating the model into an existing software system.

Monitoring and Maintenance: Continuously monitor the model's performance in production, retraining it periodically with new data to ensure it remains accurate over time. We'll also need to update the model if any underlying factors (such as market conditions or housing trends) change significantly.

By following these steps and iterating as needed, we can develop a robust machine learning system for predicting housing prices that provides valuable insights for both our company and our customers.