Executive Summary

This project is a regression task about predicting the Sale Prices of houses in the state of Iowa, Ames. The dataset used in this project comes by the people staying in this region when asked to explain all the relevant features that they would consider about, while choosing to buy a house. There are 79 features which explains all the aspects of a house, one feature for the id of the house and one target variable (SalePrice). We have pre-processed the dataset by handling the missing values and performed feature engineering by implementing TruncatedSVD to choose the most impactful 30 features to train our models. Then the trained models predict the Sale Price of the house as per the user entry through a user interactive website. The overall models and structure of this project can be used in various industries which are regression tasks like Insurance Policy calculation, Product sales price, Tuition Fee prediction, etc.

Potential Work for Follow-up Projects: This will include the optimization of the model's performance, extension of feature engineering, deployment of models, and improvement on explanation to dig deeper into the trend of housing markets. And a great improvement that we can do for this project is analysing each and every feature and its effect on the prediction value with depth to it.
Git Repository Link : https://github.com/4sameek4/HousePricePrediction

1. Introduction

Predicting house prices is an extremely challenging yet extremely important task that has very significant implications for real estate markets, financial institutions, buyers and sellers, and for policymakers. Housing prices are determined by a wide range of factors from the physical attributes and characteristics of the particular property or neighbourhood to broader market trends which are all covered in this project. The challenges of building a robust predictive model for this project is due to the huge number of features in the dataset which needs to be ranked in accordance to their impact on the prediction value. The dataset used is from the Kaggle problem, "House Prices - Advanced Regression Techniques," which provides information on house data for residential properties in Ames, Iowa. The prime objective of the development and testing of the machine learning models here is to accurately predict house prices and highlight key influencing factors. In terms of machine learning, this is a regression task. Kaggle : https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/overview

2. Scope

The scope of this project includes a broad scope starting from understanding the data to training the models and finally deploying it through a website. It can be divided into the following steps:

Data Exploration and Preparation:

- **Exploration:** Conducted an initial analysis of the dataset to understand its structure, identify key features, and examine correlations. This includes visualizing relationships between features using techniques like heatmaps and identifying any missing data.

- **Data Cleaning:** Identified missing values and treat them appropriately to ensure a clean dataset for modeling. Missing values for numerical features should be replaced by the mean, while for categorical features, use the mode.
- **Feature Segregation:** Separated numeric and categorical columns for specific preprocessing.

Data Preprocessing and Feature Engineering:

- **Numerical Features:** Set up a preprocessing pipeline that does mean imputation of missing values followed by standard scaling to standardize the distribution of the data with StandardScaler.
- **Handling Categorical Features:** Used a separate pipeline of preprocessing, which applies mode imputation on missing values and one-hot encoding for categorical features.
- **Dimensionality Reduction:** TruncatedSVD Dimension reduction on the space of features succeeds in only capturing significant information but generally minimizes noise. This helps in improving the efficiency of the model.

Modelling:

- **Model Selection**: Trained the selected models and evaluated their performance to select the best models for this dataset. For this project, we have selected two models:
    - **Linear Regression** : A baseline regression model that assumes a linear relationship between the features and the target variable.
    - **CatBoost** : Very advanced gradient boosting to deal with categorical features and feature interaction of any complexity.

Model Performance Measure:

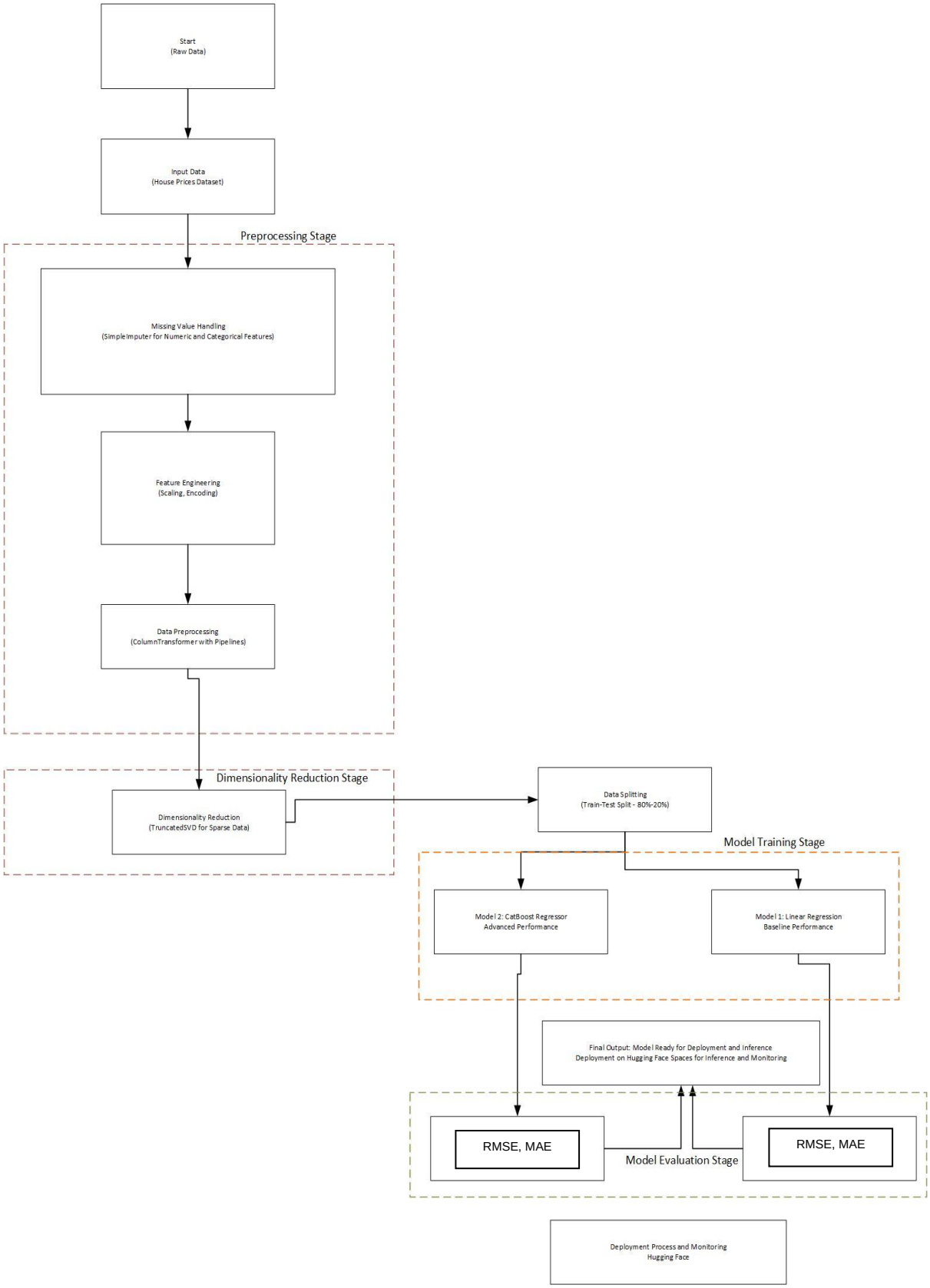
For this project we have chosen the following for measuring the performance of the Models:

- **Mean Absolute Square (MAE)**
- **Root Mean Square Error (RMSE)**

Project Deployment

- **Hugging Face :** We have used the feature Spaces provided within Hugging Face for the deployment of our user interactive website for this project.

3. Architectural Flow



4. Modelling

Data cleaning and missing value handling:

- **Numerical Features:** Missing values are filled up by the mean of respectively feature. This loses no information but at the same time keeps the scale invariant.

- **Categorical Features:** The missing values were filled using the mode of each column to maintain the statistical distribution of the categorical features.

Feature Transformation:

- **Scaling Numerical Features:** The scaling of numerical features was done using StandardScaler. Scaling it ensured that no individual feature with larger ranges dominated the learning procedure.

- **Encoding Categorical Features:** One-hot encoding encoded categorical data into columns as binary numbers. This enables the model to process categorical information in numeric terms.

Combining Preprocessing Pipelines:

A combined pipeline of preprocessing was developed using ColumnTransformer. It integrates both numerical and categorical features' separate pipelines to ensure preprocessing is uniformly applied and made efficient.

Dimensionality Reduction:

SVD was applied in the one-hot encoded data, as one-hot encoding usually leaves several sparse features, which makes the data sparse. It simplifies the data, and that is why this is an efficient model that retains all the important information that it needs.

Models Used:

1. **Linear Regression**:
    - This is very simple in contrast with the more complex model, with an easy interpretation of the relationship of the features to the target variable. In terms of having some level of baseline, its linearity confines its ability to capture practically complex patterns that exist within data.
2. **CatBoost Regressor**:
    - This is an advanced gradient boosting model built with categorical feature in mind, meaning it doesn't demand a lot of preprocessing to handle such features. It's robust and is excellent in detecting interactions between features, so it is ideal for real-world data with mixed types of features.

Hyperparameter Tuning:

- **Linear Regression**: This model does not have significant hyperparameters to tune and is mainly used as a baseline model for performance comparison.
- **CatBoost Regressor**:
    - **Iterations**: Set to 1000 to allow sufficient boosting rounds.
    - **Depth**: Set to 6, providing a balance between model complexity and performance.
    - **Learning Rate**: Set to 0.1 for moderate convergence speed.
    - **Loss Function**: Root Mean Squared Error (RMSE) was used as the loss function during training to minimize prediction error.

Hyperparameters for CatBoost were chosen based on preliminary experimentation to optimize performance while maintaining reasonable training times.

Performance Metrics Used:

1. **Mean Absolute Error (MAE)**:
    - Measures the average magnitude of errors between predicted and actual values, irrespective of the direction. Low MAE means good performance.
2. **Root Mean Squared Error (RMSE)**:
    - This gives a measure of the square root of the average of squared differences between predicted and actual values. RMSE compares to MAE where large errors are more penalized; thus, RMSE is useful in identifying models which produce large outliers.

Test Results:

- **Linear Regression**:
    - **MAE**: 21,172.16
    - **RMSE**: 35,424.29
- **CatBoost Regressor**:
    - **MAE**: 18,433.13
    - **RMSE**: 34,305.00

Result Significance

The empirical results show how advanced modeling techniques are superior to simple models in house price predictions. The MAE and RMSE values obtained were even lower for the CatBoost Regressor model as compared to the Linear Regression model, which proves its superior ability to deal with complex feature interactions and capture nonlinear relationships as well. This, in turn, generates more accurate predictions. The CatBoost's lower error values also suggest that it is a more suitable model for real-world datasets containing high variability and mixed types of features.

This gave a useful baseline but was not capable of capturing more intricacies in the dataset, as shown by relatively higher error metrics. That underscores the need to select models that may correspond well with data complexity.

5. Implementation

Machine Learning Tools –

Python Libraries:

- **Pandas**
- **NumPy**
- **Matplotlib & Seaborn**
- **Scikit-Learn (sklearn)**:

Local Deployment Tools –

Gradio

- **Gradio Interface for Model Deployment**: Gradio is a simple, intuitive tool that lets us create web interfaces for our machine learning models and, interestingly, build and deploy interactive interfaces to test models locally and share with others.
- **Features of Gradio**:
  - It will easily make sliders for textual and number inputs to accept user input for house features.
  - Displays the model's predictions interactively with options for real-time updates.
- **Local Host Deployment**: Gradio apps can then be run locally using python app.py, making testing models in a local host environment really easy before finally deploying to an external platform such as Hugging Face Space.

Local Host Deployment Process –

1. **Preparation and Packaging**:
   - It trained and tested models locally, with preprocessing pipelines being developed for data transformation.
   - Use the trained models: namely Linear Regression and CatBoost, for easy loading and prediction with the joblib.
2. **Creating a Gradio Interface**:
   - An interactive interface was developed using Gradio to take user input on key features in relation to house price prediction.
   - Once the user enters his input, real-time predictions were generated from both Linear Regression and CatBoost models.
3. **Running the App Locally**:
   - The Gradio app can be executed on a local host using a simple python command, allowing users to test and evaluate model predictions interactively.

- Following this, the local deployment of this model was accompanied by its public release in greater-reaching platforms: Hugging Face Space.

6. Conclusion

Achievements

1. **Successful End-to-End Machine Learning Pipeline**:
    - Created a robust pipeline that includes how to tackle missing values, scale numerical variables, and encode categorical features. Modularity and reusability were achieved through ColumnTransformer and Pipeline.
    - Implemented dimensionality reduction by using Truncated SVD to reduce the complexity of the data while preserving important information.
2. **Model Training and Evaluation**:
    - Developed and trained a base Linear Regression model for predicting house prices.
    - Implemented a more advanced CatBoost Regressor, which helped to increase accuracy by capturing complex relationships and interactions between features.
    - Models have been tested against specific metrics, including MAE and RMSE, as an example of the merit of complex models over simple linear models.
3. **Deployment on Local Host and Hugging Face Space**:
    - Developed an interactive web application using Gradio to make live price predictions of a house based on userinputted features.
    - Deploy the trained models and preprocessing pipeline in Hugging Face Space so that the model can be shared much widely among users for testing and inference purposes.
4. **Interpretability and Model Comparisons**:
    - Compared performance between baseline Linear Regression and the CatBoost Regressor. Demonstrates the advantage of use over more complex models for datasets with intricate feature interactions.

Limitations

1. **Handling of Feature Complexity**:
    - Thus, though preprocessing and feature transformation were effective, high dimensionality and complexity of interactions in the dataset may have resulted in information loss during the process of dimensionality reduction.

- One-hot encoding of categorical features leads to the presence of lots of sparse features; that, in turn, can lead to inefficiencies in processing and possibly increased computational overhead.

2. **Model Generalization**:
   - The trained models were tested on a split of the given dataset, but generalization to new data or unseen properties in different geographic regions is still fully to be tested.
   - While doing very well on this dataset, CatBoost could be seen as a merged version with possibly XGBoost, LightGBM, or even neural networks for optimal performance.
3. **Data Quality and Limitations**:
   - It was bestowed with missing values and potential outliers and was so managed with basic imputation and scaling techniques. More advanced imputation or outlier detection may be needed to ensure quality data.
   - Though important features like the economic trend, neighbourhood crime rates, or school quality are not within the dataset, they can provide additional predictive power.

Avenues for Future Work

1. **Advanced Hyperparameter Tuning**:
   - To automate the hyperparameter tuning methods, such as grid search, random search, or Bayesian optimization, that might tune up a CatBoost model and potentially any other model.
2. **Feature Engineering**:
   - More comprehensive feature engineering would be possible by considering the interactions, polynomial features, and domain-specific transformations to enhance the accuracy of model prediction.
   - Incorporate external data sources, such as economic indicators or crime statistics, to provide additional context to house price predictions.
3. **Exploration of Additional Models**:
   - Train and compare more models, including XGBoost, LightGBM, Random Forests, and deep learning architectures, to find the best model for this problem.
   - For modeling large interaction structures within the data, deep models including feedforward neural networks have to be used.

4. **Deployment Enhancements**:
    - Improve the user interface and functionality of the deployed application on Hugging Face Space, including better input validation, visualization of predictions, and interactive feedback mechanisms.

Individual Contributions

Sameek Bhattacharya

- Applying TruncatedSVD for dimensionality reduction to optimize data representation.
- Training a baseline Linear Regression model and evaluating its performance.
- Setting up and training the CatBoost Regressor, including manual tuning of key hyperparameters (iterations, depth, learning rate).
- Applied TruncatedSVD for dimensionality reduction
- Trained and evaluated Linear Regression model
- Trained CatBoost model

Mahesh Chowdary Ponnaganti

- Importing and loading the dataset, initial exploration, identifying missing values, and creating visualizations (e.g., heatmap for correlation matrix).
- Implementing data cleaning strategies, handling missing values using SimpleImputer.
- Segregating numerical and categorical features.
- Creating preprocessing pipelines for numerical features (e.g., scaling) and categorical features (e.g., one-hot encoding).
- Initial data loading and exploration
- Implemented missing value handling
- Created preprocessing pipeline for numerical features

Tulasi Sai Jaliparthi

- Evaluating the performance of both Linear Regression and CatBoost models using metrics like MAE, RMSE.
- Comparing model results and documenting key insights and findings.
- Setting up the Gradio interface for user input and integrating both models for predictions.
- Preparing the deployment on Hugging Face Space and testing the model's functionality.

- Evaluated model performance using MAE and RMSE
- Created Gradio interface for house price prediction
- Deployed model on Hugging Face Space