Machine Learning for Business Applications
Project Update 2

The business problem being addressed is predicting the house prices. We have trained machine learning models using datasets that contain a variety of features related to housing characteristics, such as lot size, number of rooms, and other pertinent factors. By evaluating the performance of these models in predicting house prices, the objective is to develop a reliable tool for stakeholders in the real estate industry to accurately estimate property values, facilitating better decision-making processes and market insights.

In the process of model training and evaluation, we utilized various machine learning algorithms to predict house prices. Initially, a linear regression model was employed as a baseline approach, which captures linear relationships between the independent variables and the target variable - the sale price. Additionally, ridge and lasso regression models were explored to mitigate overfitting by introducing regularization through penalty terms. A random forest regressor was deployed to leverage ensemble learning and capture non-linear relationships within the dataset. These models were chosen due to their simplicity, interpretability, and ability to handle both numerical and categorical features present in the housing dataset.

The dataset utilized for training and evaluation comprised features relevant to housing characteristics, sourced from Kaggle. It included information such as lot size, number of rooms, neighborhood demographics, and building materials. The dataset was sufficiently large, containing roughly two thousand instances with 78 features. We performed preprocessing steps such as removing outliers and handling missing values to ensure data quality and model robustness.

In evaluating the models' performance, several common metrics such as root mean squared error (RMSE), mean absolute error (MAE), and coefficient of determination (R^2 score) were utilized. RMSE and MAE quantify the magnitude of errors in predicting the “sale price” of houses, with lower values indicating better model performance. Specifically, R^2 measures the proportion of the variance in the dependent variable (in this case, house prices) that is explained by the independent variables (features) included in the model. A higher R^2 score indicates a better fit of the model to the data. A value of 1 indicates that the model explains all of the variance in the data, while a value of 0 indicates that the model does not explain any of the variance. Therefore, higher R^2 scores signify better model performance in capturing the underlying patterns and trends in the data.These metrics were chosen based on their ability to provide actionable insights for stakeholders in the real estate industry.

In adhering to best practices, the team implemented robust data preparation techniques to ensure the reliability of the trained models. The dataset was divided into training and testing sets using a standard train-test split approach, allocating 80% of the data for training and 20% for testing - this helped us understand which is the best model for our use case. Apart from this

we had a separate test dataset which helped us evaluate the best models performance post deciding which is the best model. This allowed for independent validation of the models' performance on unseen data. We addressed data inconsistencies and missing values before training the data on different models. We filled the missing values using the imputation method for numerical and categorical parameters. For numerical parameters we used the mean whereas for categorical parameters we used the most frequent value in the dataset. We scaled them to ensure uniformity in feature magnitudes across the dataset. In case of categorical variables we used one hot encoding before we jumped on to the model training part.

For model training, we started with multilinear regression as the first step. The model gave us an R-squared error of 0.8772 on the test dataset. The other metrics performance is as attached in Appendix1. Next we used regularization techniques to combat overfitting, particularly the ridge and lasso regression models. By introducing penalty terms to the loss function, these techniques encouraged simpler models and reduced the likelihood of fitting noise in the data, enhancing the models' generalization ability. The R-squared on ridge regularization model was 0.8694 and the Root Mean Squared Error and Mean Absolute Error are as can be seen in Appendix 2. Now for the lasso regularization method, the R-squared score came out to be 0.9013 on the test dataset and the Root Mean Squared Error and Mean Absolute Error are as can be seen in Appendix3. The final model we applied was a random forest regressor. For this model, the R-squared score came out to be 0.8945 on the test dataset and the Root Mean Squared Error and Mean Absolute Error can be found in Appendix4.

In evaluating model performance, a comprehensive methodology was adopted to assess predictive accuracy and robustness. Metrics such as root mean squared error (RMSE), mean absolute error (MAE) were minimized, and coefficient of determination (R^2 score) were calculated on both training and testing sets to gauge model performance across different datasets. We considered strategies for handling challenges such as blank values and class imbalances. While luckily, the provided dataset did not exhibit significant class imbalances, techniques such as oversampling or undersampling could be employed if necessary to ensure fair evaluation and prevent biased model performance metrics. Overall, the adopted methodologies aimed to yield reliable and interpretable models while addressing potential challenges inherent in the dataset.

To organize the code and ensure reproducibility, a structured approach was adopted to facilitate efficient model training and evaluation. We utilized Jupyter notebooks as the primary tool for code development and experimentation and Python as the language, offering an interactive environment for exploring data, implementing algorithms, and visualizing results. The Python libraries we used includes pandas, numpy, sklearn, StandardScaler, ColumnTransformer, Pipeline, SimpleImputer, train_test_split, LinearRegression, Ridge, Lasso, RandomForestRegressor, OneHotEncoder, mean_squared_error, mean_absolute_error, r2_score, joblib, matplotlib.pyplot. The code was commented so as to make it easy to understand and organized into modular components with each section dedicated to specific tasks such as data preprocessing, model training, and evaluation.

As a next step, we are planning to implement an organized storage using GitHub. The entire codebase, including Jupyter notebooks, preprocessing scripts, and model training scripts, would be stored in a GitHub repository. This will ensure collaboration among team members and provide a centralized platform for managing the codebase. Additionally, Git branching strategies would be employed to facilitate parallel development while preserving the stability of the main current codebase.

To enhance reproducibility and facilitate knowledge transfer, a comprehensive runbook is being created detailing step-by-step instructions for replicating the entire workflow. The runbook will include guidance on setting up the development environment, downloading datasets, executing code scripts, and interpreting results. Documentation within the Jupyter notebooks provided inline comments and explanations for code sections, ensuring clarity and enabling future reference. By documenting all these at one place we can ensure the sustainability and scalability of the project for further collaboration and iterations.

We have outlined the process of training and evaluating machine learning models to address the critical business problem faced in the real estate industry of predicting house prices. Through implementing linear regression, linear regression with ridge regularization, linear regression with lasso regularization, and random forest regressor and assessing multiple evaluation metrics such as RMSE, MAE and R-squared scores, the team has developed models capable of accurately estimating property values based on housing characteristics. The best performance was displayed by the linear regression model with lasso regularization and it gave a R-squared score of 0.8945 on the test data whereas the RMSE and MAE are as seen in **Appendix5**. The model holds significant importance for stakeholders in the real estate industry, providing invaluable insights for decision-making processes and market analysis. Moving forward, the project could benefit from further refinement of model architectures, exploration of advanced algorithms, and incorporation of additional features and cutting down on unnecessary features to enhance predictive accuracy. Moreover, model adaptation to evolving market trends will be crucial for maintaining the relevance and effectiveness of the developed ML system.

Appendix

Appendix1: Multilinear regression model

```
Testing RMSE: 619801270.4219859
Testing MAE: 17411.274279914905
Testing R² Score: 0.8772631909740616
```

Appendix2: Multilinear regression model with ridge regularization

```
Ridge Testing RMSE: 659377788.2597266
Ridge Testing MAE: 17901.60288757132
Ridge Testing R² Score: 0.8694260087294119
```

Appendix3: Multilinear regression model with lasso regularization

```
Lasso Testing Training RMSE: 498024684.45528495
Lasso Testing Training MAE: 14956.009264372162
Lasso Testing Training R² Score: 0.9013781295663131
```

Appendix4: Random Forest Regressor

```
Random Forest Testing RMSE: 532666489.4167185
Random Forest Testing MAE: 15239.764175257733
Random Forest Testing R² Score: 0.8945181491132714
```

Appendix5: Multilinear regression model with lasso regularization on the main test data

```
Best Testing RMSE: 532666489.4167185
Best Testing MAE: 15239.764175257733
Best Testing R² Score: 0.8945181491132714
```