**Comparison of Housing Price Prediction using SVM, RA and LRM Model Techniques**.

Rachana Gudivada

**Synopsis:**

Regardless of one's ability to pay, home prices have an impact on the decision to buy. Many scholars have studied the prediction of housing prices using the house pricing index (HPI) and various machine learning models to accurately predict house prices. Using secondary data from the Boston Housing dataset, this project attempts to perform a comparative analysis of three machine learning models, namely support vector machine (SVM), random forest (RA), and linear regression model (LRM), in the prediction of housing prices. To improve the models' prediction, the project will base its model construction on characteristics such as the typical number of rooms, highway accessibility, crime rate, house-age etc. Python codes will be used for the dataset preprocessing, visualization, training/testing, and prediction phases of the project.

**Introduction:**

Housing price prediction is an important problem in real estate that can guide investment decisions. In this report, we compare three machine learning models for predicting house prices - linear regression, regression trees, and neural networks. Using a housing dataset with attributes like square footage, location, etc., we train each model and evaluate their accuracy at forecasting prices on a holdout test set. By analyzing performance metrics like mean absolute error, we aim to determine the strengths and weaknesses of each approach. The results will provide insights into which machine learning techniques are best suited for housing price prediction based on predictive accuracy. This report provides a useful comparison of modeling techniques for an important real-world forecasting application.

**Methodology**

**Data Processing:**

Data analysis and visualization were done using Python and associated libraries like Pandas, Matplotlib, etc. as shown in appendix (I) below. The code snippets show importing these libraries. Loading and exploring a dataset. The screenshots show data frames and initial plots to understand the data. Cleaning/pre-processing the data involved the handling of missing values, converting data types, etc.

**Data Analysis:**

Data visualization was executed by generating various plots to extract insights like the histograms, scatter plots, line plots, heatmaps, clustering plots etc. Model evaluation was carried out by assessing model performance using different metrics like MAE, RMSE, R-squared, precision-recall etc . And generating prediction vs actual plots. On the model selection, we split the data to allow for training and testing of the dataset. Again, based on the dataset we had three types of regressions like: linear regression, random forest, and SVM are discussed.

**Linear Regression:**

Linear regression was implemented as a baseline modeling technique for the housing price prediction task. The sklearn LinearRegression class was used to fit a linear model between the input housing features and the target price variable. Key assumptions of linear regression are that the relationship between the independent variables and dependent variables is approximately linear, and that the data is homoscedastic. Grid search combined with cross-validation was leveraged to tune the regularization hyperparameter alpha to prevent overfitting. While linear regression is simple and interpretable, its limitation is that it cannot capture complex non-linear relationships in the data. More flexible techniques like random forest and SVM outperformed linear regression in terms of predictive accuracy on this problem. However, linear regression provides a useful baseline to compare the improvement in performance gained by the ensemble and kernelized models. The linear coefficients also give some insight into the general directional relationship between housing features and price.

**Random Forest:**

The random forest ensemble technique was used for regression tasks in the project. The sklearn Random Forest Regressor was utilized, training multiple decision trees on bootstrapped samples of the training data. Two key hyperparameters - number of trees and maximum tree depth - were tuned with cross-validation to boost model performance while avoiding overfitting. Feature importance was calculated to identify the most relevant input variables for prediction. Random forest improved on single decision tree performance by aggregating across the ensemble. Additionally, the hyperparameters like number of trees and maximum depth were tuned using cross-validation to obtain optimal accuracy while avoiding overfitting to the training data. Random forest's flexibility to capture complex nonlinear relationships, combined with the tuning process, averaging across decorrelated trees, and guarding against overfitting enabled it to outperform other sophisticated techniques like SVM and achieve the highest overall accuracy.

**Support Vector Machines:**

For regression problems, support vector regression was implemented using sklearn's SVR model. Both linear and non-linear RBF kernels were evaluated. The RBF kernel was able to model complex non-linear relationships by projecting the data into higher dimensions. Grid search with cross-validation was used to tune the regularization strength C and insensitive loss parameter epsilon. SVM provided flexibility to fit the data well. Performance was compared with linear regression and random forest models.

**Discussion:**

The goal of this project was to develop a machine learning model for predicting [target variable] using the provided dataset. Several regression algorithms were evaluated including linear regression, lasso and ridge regression, random forest, and support vector machines. The data was first explored and analyzed to identify trends, correlations, and data distributions. Different features were engineered and normalization techniques like standard scaling applied. The models were trained on a train set and hyperparameters tuned using grid search cross-validation. Overall model performance was compared based on metrics like RMSE, MAE, and R-squared on the test set. The random forest model achieved the highest accuracy for regression among the machine learning algorithms used with the lowest RMSE, indicating its ability to capture complex non-linear relationships in the data better than the other techniques. This can be attributed to random forest's ensemble approach, where multiple decision trees are trained on bootstrapped subsets of data, and each split considers a random set of features. This bagging and feature randomness helps decorrelate the trees, which when aggregated as an ensemble, reduces variance and leads to higher prediction accuracy than a single decision tree. The ensemble method prevented overfitting through averaging across decorrelated decision trees. SVM with the RBF kernel also performed well, showing the benefit of a non-linear model. The linear regression and regularized models like lasso and ridge performed worse, indicating complex relationships.

**Conclusion:**

**­­­­­­­­­­­­**This project demonstrated a typical machine learning workflow for a regression prediction problem. The data was processed, analyzed, and used to train and tune multiple types of regression models. Based on the evaluation metrics, random forest was the optimal algorithm for this dataset and modeling task. The results show the importance of trying different algorithms, tuning hyperparameters, and using performance metrics to select the right model. The high accuracy of random forest highlights the power of ensemble techniques for modeling complex data. In future work, additional feature engineering and hyperparameter optimization could potentially improve accuracy further. Overall, the project successfully built an accurate regression model using machine learning techniques.

**References:**

1. House Price Index. Federal Housing Finance Agency. https://www.fhfa.gov/ (accessed September 1, 2019).
2. Fan C, Cui Z, Zhong X. House Prices Prediction with Machine Learning Algorithms. Proceedings of the 2018 10th International Conference on Machine Learning and Computing - ICMLC 2018. doi:10.1145/3195106.3195133.
3. Phan TD. Housing Price Prediction Using Machine Learning Algorithms: The Case of Melbourne City, Australia. 2018 International Conference on Machine Learning and Data Engineering (ICMLDE) 2018. doi:10.1109/icmlde.2018.00017.
4. Mu J, Wu F, Zhang A. Housing Value Forecasting Based on Machine Learning Methods. Abstract and Applied Analysis 2014;2014:1–7. doi:10.1155/2014/648047.
5. Lu S, Li Z, Qin Z, Yang X, Goh RSM. A hybrid regression technique for house prices prediction. 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) 2017. doi:10.1109/ieem.2017.8289904.
6. Ivanov I. vecstack. GitHub 2016. https://github.com/vecxoz/vecstack (accessed June 1, 2019). [Accessed: 01-June-2019].
7. Wolpert DH. Stacked generalization. Neural Networks 1992;5:241–59. doi:10.1016/s0893-6080(05)80023-1.
8. Qiu Q. Housing price in Beijing. Kaggle 2018. https://www.kaggle.com/ruiqurm/lianjia/ (accessed June 1, 2019).
9. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. The Journal of Machine Learning Research 2011;12:2825–30.
10. Breiman L. Random Forests. SpringerLink. https://doi.org/10.1023/A:1010933404324 (accessed September 11, 2019).
11. Raschka S, Mirjalili V. Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow. 2nd ed. Birmingham: Packt Publishing; 2017.
12. Chen T, Guestrin C. XGBoost. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 16 2016. doi:10.1145/2939672.2939785.

APPENDIX

(I)

A screenshot of a computer

Description automatically generated

(II)

A screenshot of a computer

Description automatically generated

(III)

A screenshot of a computer screen

Description automatically generated

(IV)

A screenshot of a computer

Description automatically generated

(V)

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

(VI)

A screenshot of a graph

Description automatically generated

(VII)

A screenshot of a computer

Description automatically generated

(VIII)

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

(IX)

A screenshot of a computer program

Description automatically generated

(X)

A graph with blue dots

Description automatically generated

A screen shot of a computer

Description automatically generated

(XI)

A blue line graph with numbers

Description automatically generated

A screen shot of a graph

Description automatically generated

(XII)

A screenshot of a computer

Description automatically generated

(XIII)

A screenshot of a computer

Description automatically generated

(XIV)

A screen shot of a graph

Description automatically generated

(XV)

A screenshot of a computer program

Description automatically generated

A blue line graph with numbers

Description automatically generated

A screen shot of a graph

Description automatically generated

(XVI)

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

(XVII)

A screenshot of a computer

Description automatically generated

A graph with blue dots

Description automatically generated

XVIII

A screenshot of a computer

Description automatically generated

A blue line graph with numbers

Description automatically generated

(XIX)

A screen shot of a graph

Description automatically generatedA screenshot of a computer

Description automatically generated