Predicting House Prices Using Machine Learning.

# Test Heading

## **Test Heading 2**

### Test Heading

This Settings is fixed.

Test Normal

# Heading 1

## Heading 2

### Heading 3

References :

* <https://www.globalpropertyguide.com/Asia/India/Price-History>
* Feature selection: <https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e>
* Feature importance : <https://www-sciencedirect-com.ezproxy.herts.ac.uk/science/article/pii/S1532046418301400?via%3Dihub>
* Features importance : <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-10-213>
* Feature Importance algorithm: select k best use chiquare selection feature: <https://towardsdatascience.com/chi-square-test-for-feature-selection-in-machine-learning-206b1f0b8223>
* Support Vector Machine: <https://shuzhanfan.github.io/2018/05/understanding-mathematics-behind-support-vector-machines/>
* Support Vector Machine : <https://en.wikipedia.org/wiki/Support-vector_machine>
* Random Forest : <https://en.wikipedia.org/wiki/Random_forest>
* Research : https://www.houselogic.com/sell/how-to-sell-step-by-step/home-market-analysis/
* Research Tool: <https://www.bricknbolt.com/cost-estimator>
* Gradient Boosting Regressor: <https://machinelearningmastery.com/gradient-boosting-machine-ensemble-in-python/>
* Gradient Boosting Regressor: <https://en.wikipedia.org/wiki/Gradient_boosting>
* ExtraTreesRegressor - https://machinelearningmastery.com/extra-trees-ensemble-with-python/

References with citation

* Gnu.org. 2021. *GNU General Public License v2.0 - GNU Project - Free Software Foundation*. [online] Available at: <http://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html> [Accessed 18 August 2021].
* kumar, a., 2020. *House Price Prediction Challenge*. [online] Kaggle.com. Available at: <https://www.kaggle.com/anmolkumar/house-price-prediction-challenge> [Accessed 18 August 2021].
* En.wikipedia.org. 2021. *Real Estate (Regulation and Development) Act, 2016 - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Real\_Estate\_(Regulation\_and\_Development)\_Act,\_2016> [Accessed 18 August 2021].
* Géron, A., 2019. *HANDS-ON MACHINE LEARNING WITH SCIKIT-LEARN, KERAS, AND TENSORFLOW: CONCEPTS, TOOLS, AND TECHNIQUES*. SEBASTOPOL: O'REILLY MEDIA, pp.186-210.
* Ozdemir, S., 2016. *Principles of data science*. Birmingham, UK: Packt Publishing, pp.98-248.
* Brownlee, J., 2021. *How to Develop a Gradient Boosting Machine Ensemble in Python*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/gradient-boosting-machine-ensemble-in-python/> [Accessed 19 August 2021].
* En.wikipedia.org. 2021. *Gradient boosting - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Gradient\_boosting> [Accessed 19 August 2021].
* Brownlee, J., 2021. *How to Develop an Extra Trees Ensemble with Python*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/extra-trees-ensemble-with-python/> [Accessed 19 August 2021].
* Cython.org. 2021. *Cython: C-Extensions for Python*. [online] Available at: <https://cython.org/> [Accessed 20 August 2021].
* Jython. 2021. *Home*. [online] Available at: <https://www.jython.org/> [Accessed 20 August 2021].
* Pmay-urban.gov.in. 2021. *Pradhan Mantri Awas Yojana*. [online] Available at: <https://pmay-urban.gov.in/> [Accessed 20 August 2021].
* Bajajfinserv.in. 2021. *Home Loan Tax Benefit 2021: Tax Rebate Under Section 24, 80EE & 80C*. [online] Available at: <https://www.bajajfinserv.in/tax-benefits-on-home-loan> [Accessed 20 August 2021].
* 2021. *RERA Act*. [ebook] Delhi: Indian Government. Available at: <https://up-rera.in/pdf/reraact.pdf> [Accessed 20 August 2021].

# Appendices

## Appendix 1

In this appendix, how relators will increase the land prices. Even not worth it.



Chart, bar chart, histogram

Description automatically generated

The above code explains the top spends on the areas. The chart reveals a critical analysis of house prices. The figure shows unnecessary spikes. As this shows top central cities of India spends less amount than rural areas. As shown, rural areas have more prices than the more populated areas.

# Appendix 2

In our project, the main models are derived from several observations and even several models as well—this appendix discussing polynomial features.

Earlier, as discussed in chapter 3, research method polynomial features. This model is similar to linear regression. Polynomial features are not model changing the data before sending it to the model.

As seen in the linear regression house prices, the dataset was not linear. Each feature will multiply and create new features in polynomial features, and now the model is easy to separate and predict accurately. While multiplying features need to specify degrees of freedom, so features have that much freedom.

Graphical user interface, text, application, email

Description automatically generated

**Hyper-parameter fine tune**

Table

Description automatically generated

Implementing the basic first model is not sufficient. Here polynomial features data was divided with degrees of freedom. However, here it is unaware of how many degrees of freedom is well suited for models. To know this, we need to test each degree of freedom. That was implemented above, and results show.

## Appendix 3

Logistic regression is previously researched. This time is needed to implement using the housing dataset. Below, the implementation shows not that much fit with our dataset.

Graphical user interface, text, application, email

Description automatically generated

Even though logistic regression does not fit much with our dataset, logistic regression is also widely known for a penalty for optimising final output. Polynomial Features makes the model converge faster than the usual model, and the model performs more efficiently. In logistic regression, three penalties are available “L1, L2, Elastic Net”. Below are which penalty best suits for either L1 or Elastic-net. L2 is the default implementation if without mentioned anything.

**L1 best params**

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated with medium confidence

**Elastic net**

Graphical user interface, application

Description automatically generated

Text

Description automatically generated with medium confidence

## Appendix 4

In this appendix, Discussing “Gradient Boosting Regression” (Gradient boosting - Wikipedia, 2021). Gradient Boosting is one of the best models of ensemble methods. Here in this model, we will create each branch of feature in our dataset. All features in our dataset do not give full performance towards final predictions. So, this model divides the data into branches. If a branch is a weak supporter of the final output, then using gradient boosting regressor optimises full support (Brownlee, 2021).

Graphical user interface, text, application, email

Description automatically generated

This algorithm works astonishingly on the housing dataset. It predicted well without losing performance.

## Appendix 5

In this appendix, Discussing “Extra Trees Regression”. It is also one of the ensemble methods. Extra Trees regression will create many randomised features such as nodes/leaves and predictions (Brownlee, 2021).

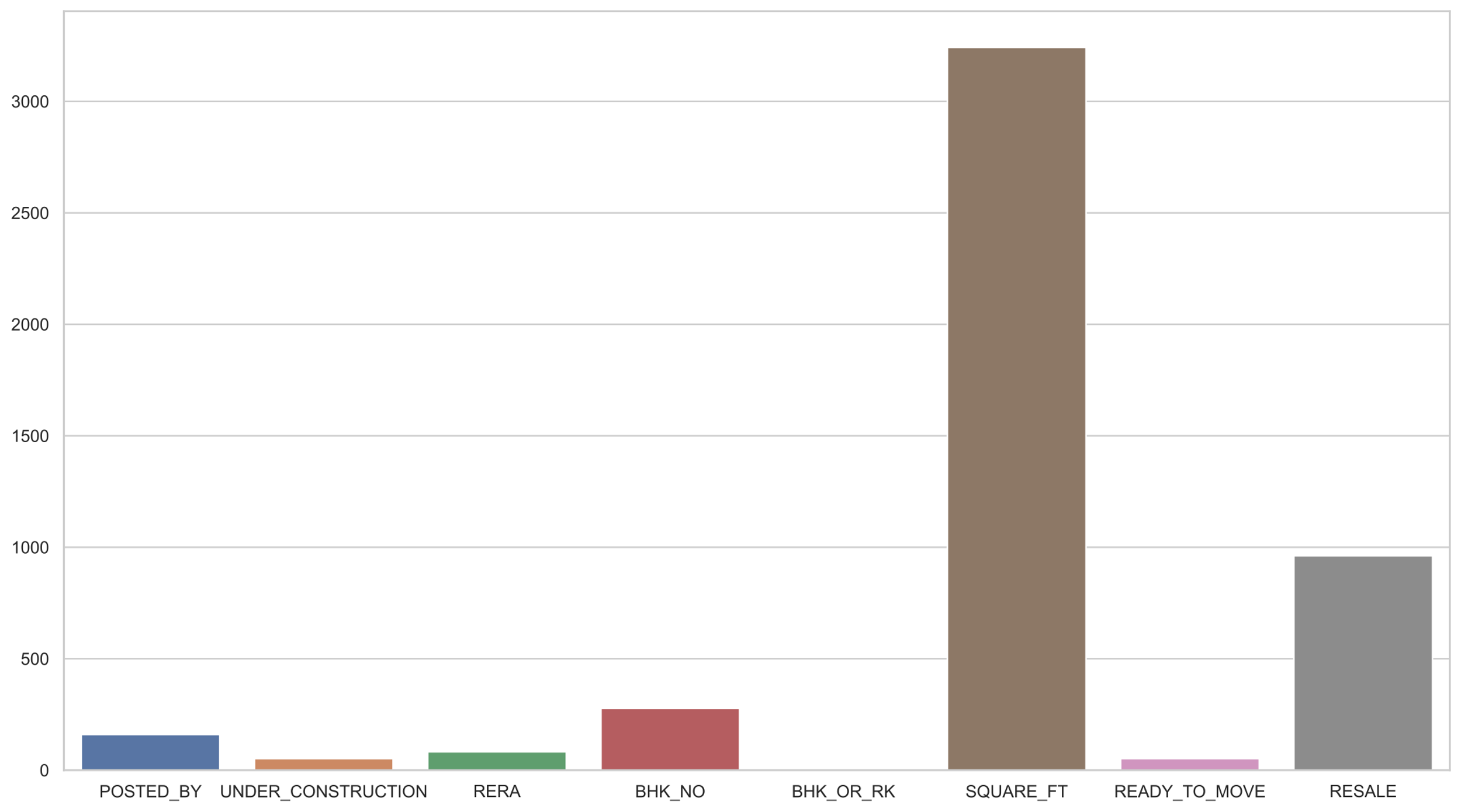
Graphical user interface, text, application, email

Description automatically generated

# Appendix 6

Feature Importance

Every data has specific attributes that support the final output. Even the housing dataset has specific attributes for price prediction. To know which features, help most of the output.



In the housing dataset, it looks like square feet and resale most supporting factors for final predictions.

# Appendix 7

# Recurrent Neural Network

Recurrent Neural networks are special time-based neural networks. Here neurons converge is based on time. When neurons are converging through time, the major problem is that throughout gradients will vanish through time.

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Figure A.7 shows the recurrent neural networks implementation.

The above code shows an implementation of recurrent neural networks. There are some errors during the training time. These errors are occurring due to dimensions issues. Taking care of time factors, these implementations have not been considered.

# Appendix 8

# PyCaret Framework

PyCaret is a framework developed specially designed for machine learning and implementation. In machine learning to conclude a model, data needs to transform and experiment with several models and needs to be final. To cut the time off, this framework was developed.

Table

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Figure: A.8 shows the PyCaret implementation

Only using two lines of code, this framework used almost all the most used models, and the error rate was also predicted.

# Appendix 9

# Students T Distribution

Text

Description automatically generated

 Figure A.9 Implementation of Students T Distribution

This implementation for students t distribution applied in making price intervals.

# Appendix 10

# Similar houses in same location and similar house prices.

Text

Description automatically generated

Figure A.10 Implementation of similar houses and prices

This implementation looked similar houses and sold prices in the exact location and shows comparable house prices. PyCaret also helps end-users to understand how other prices will be. Due to the time factor, this implementation is not included.