The University of Akron
IdeaExchange@UAkron

Williams Honors College, Honors Research Projects

The Dr. Gary B. and Pamela S. Williams Honors College

Spring 2023

Predicting Housing Prices Using AI

Eric Sconyers
eas174@uakron.edu

Follow this and additional works at: https://ideaexchange.uakron.edu/honors_research_projects

 Part of the Artificial Intelligence and Robotics Commons, and the Data Science Commons

Please take a moment to share how this work helps you through this survey. Your feedback will be important as we plan further development of our repository.

Recommended Citation

Sconyers, Eric, "Predicting Housing Prices Using AI" (2023). *Williams Honors College, Honors Research Projects*. 1733.
https://ideaexchange.uakron.edu/honors_research_projects/1733

This Dissertation/Thesis is brought to you for free and open access by The Dr. Gary B. and Pamela S. Williams Honors College at IdeaExchange@UAkron, the institutional repository of The University of Akron in Akron, Ohio, USA. It has been accepted for inclusion in Williams Honors College, Honors Research Projects by an authorized administrator of IdeaExchange@UAkron. For more information, please contact mjon@uakron.edu, uapress@uakron.edu.

Predicting Housing Prices Using AI

Eric A. Sconyers

College of Engineering, University of Akron

May 3, 2023

Abstract

I have created an AI model that can predict housing prices with 70 percent accuracy in Ames Iowa. I was able to use data from a website called Kaggle.com which is a website that provides datasets to the public so they can create AI models with the data. I found the dataset pertaining to housing prices in Ames Iowa. With this data, I was able to create an AI model that can predict the housing price of these homes. The technology I used in this project was Python as the programming language, and I used the scikit-learn library which has many useful tools that I used when creating this model. I give a run down on the steps I took to create this model as well as my thought process behind each decision. I go in-depth on the importance of feature selection and data preparation to increase the accuracy of the model. As well as other techniques such as encoding and ensemble models to increase the accuracy as well. In conclusion, I found that this was a great learning experience for me and I hope that those who wish to create similar AI models in the future can utilize some of the techniques that I cover.

Predicting Housing Prices Using AI

The first step in creating any AI model is securing the data needed to create the model you wish to create. Kaggle.com is a great source to practice building AI models and also has a great community that will help and provide feedback on your work. This is the site I used to obtain the housing data needed to create the AI model. There are at least two data sets needed when creating an AI model; a training dataset and a testing dataset. Both have the same number of columns, but they have different data stored in the rows. This allows us to use the training data to train the model and then use the testing data to measure the accuracy of the model. The way I like to think of it is like looking in the back of the book with the answers when doing homework so that you can learn, then testing your knowledge on the test. The dataset had 80 columns and 1460 rows of data. From that dataset, I then created a training and testing dataset using the train_test_split function from the scikit-learn library. This function breaks up the split into four variables which I named x_train, x_test, y_train, and y_test. I specified my testing size to be 33 percent of the original dataset. This means 30 percent of that data will be used just to test the data and the rest will be used to train the model. The reason why there are four variables is that the model needs to know what to predict, so we must specify are x and y variables which in this case the y variable was SalePrice which is what we want to predict and x is the other 79 columns.

Encoding and Imputation

Now that I have created my training and testing splits, I now need to encode the data so that the model can interpret the data properly. AI models are based on statistical models which need numerical data, not string data. I decided to label encode the string data; I did this by utilizing the LabelEncoder() function from scikit-learn which will label encode the string data. The way label encoding works is let's say in a dataset there was a column for gender, and in that column, the entries were either “Male” or “Female”. A label encoder would have “Male” equal 0 and “Female” equal 1 so that numeric data can

represent the categorical string data. Many times a dataset will have empty or Nan values which is not good for AI models. A way to get around this is to use an imputer. There are different strategies one could use for imputation, but I used the median of the missing values and inserted them into the missing values in the dataset. I used the SimpleImputer() function from the scikit-learn library.

Feature Selection

Feature selection is a key part of creating an AI model. Features are the columns in a dataset, and certain features may have more impact on the prediction than others. So selecting the important features and using those for the AI model not only makes the model faster to train but also increases the accuracy since there are fewer features that could be misleading to the AI model. The way I found which features to use was by using the SelectKBest() function from the scikit-learn library. Using this function I was able to refine the features from 79 down to 15.

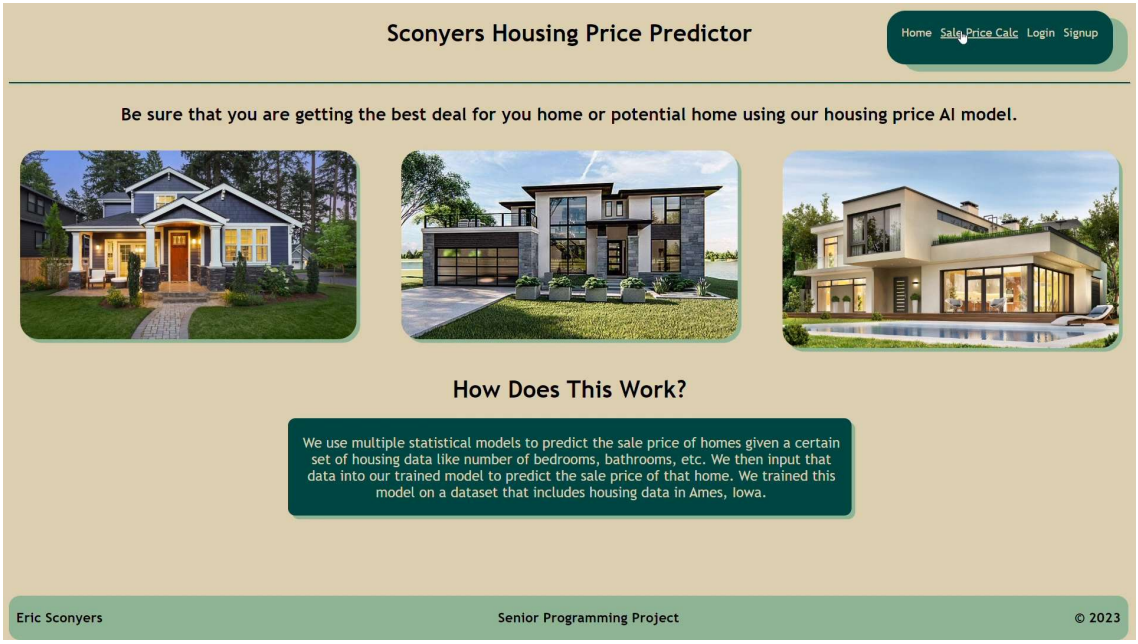
Ensemble Model

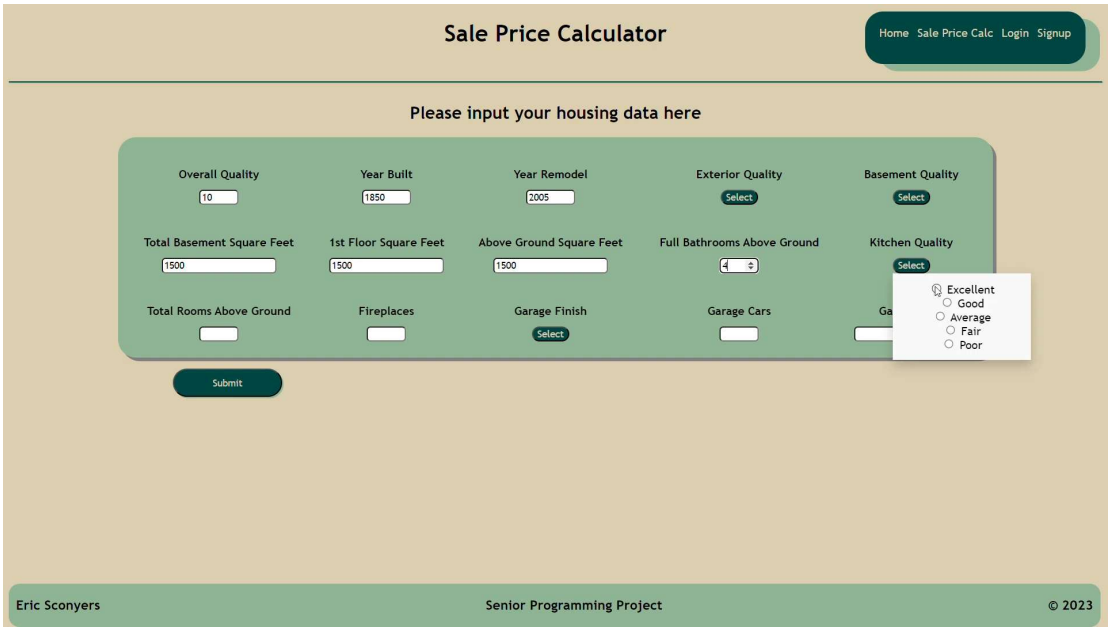
Now that I have the data encoded and the important features selected; now it is time to find the best statistical model to use. Since this AI model is a regression model I had to use regression statistical models to make my prediction. The regression models that I used from the scikit-learn library were LinearRegression(), RandomForestRegressor(), BayesianRidge(), and KneighborsRegressor().
In order to combine all four models into one ensemble model I have to weigh them appropriately. I do this by using the VotingRegressor() function which combines the models into an ensemble and then averages their individual predictions to form a final prediction. The reason why this is a useful technique is that different models may provide different predictions that can be useful for the accuracy of the model. Think of it like having a team of people to solve a problem; everyone has a slightly different perspective on the situation which could lead to better results.

Finally, I made my prediction on the testing with my ensemble model. Then using the testing data I was able to measure the accuracy of the model. The model scored 70.39% accuracy.

Creating a Website to Host the Model

I then created a website to host the model using HTML, CSS, and PHP. The reason why is so that a user can input housing data through the website and receive a prediction from the AI model. Below are some screenshots of the website itself.





Once the user inputs the data into the website, the data then gets processed using the same steps I described earlier and then displays the prediction on the screen.

Conclusion

There are many different techniques and approaches one can use when creating an AI model. I know that others on Kaggle.com had slightly different approaches to creating a housing price AI model. The goal was to show my thought process and help explain some of the many techniques that are used in AI model creation. Some of the important techniques include feature selection, encoding, and ensemble models.