00:00-19:08

Building a Machine Learning Model with K-Fold Cross-Validation and Grid CV

Introduction

In the previous tutorial, we focused on cleaning up our data frame to prepare it for model building. In this tutorial, we will dive into the process of building a machine learning model. Additionally, we will utilize K-Fold cross-validation and Grid CV to identify the best algorithm and parameters. Our dataset contains a location column, and since machine learning models cannot interpret text data, we need to convert this categorical information into numerical data. One common method for achieving this is one-hot encoding, also known as dummies. We will leverage the pandas `get_dummies` function to perform one-hot encoding.

One-Hot Encoding

To convert the location column into numerical data, we will use the `get_dummies` function from the pandas library. The process involves creating new columns for each unique location value and setting the corresponding values to 1 or 0. This straightforward encoding method allows us to represent categorical information numerically. Once the dummy columns are created, we will store them in a separate data frame and append them to our main data frame.

Model Building

After performing the necessary data processing steps, we can begin building our machine learning model. First, let's examine the shape of the data frame, which consists of 7,000 rows and 245 columns. To train our model, we need to separate the independent variables (X) from the dependent variable (price, Y). We will drop the price column from the data frame to obtain the X variable.

Next, we will split our dataset into training and test sets using the `train_test_split` method from the `sklearn.model_selection` module. We will allocate 20% of the samples for testing and use the remaining 80% for model training.

We will create a linear regression model and fit it using the training data. Once the model is trained, we can evaluate its performance by calculating the score. In this case, our linear regression model achieves a score of 84%, which is quite decent. However, to ensure we have the best possible model, we will explore other regression techniques.

K-Fold Cross-Validation

To assess the performance of different regression techniques, we will employ K-Fold cross-validation. This technique involves dividing the dataset into K subsets or folds, training the model on K-1 folds, and evaluating it on the remaining fold. We import the necessary modules and create a ShuffleSplit object to randomize the samples within each fold. By using cross-validation, we obtain scores consistently above 80%.

Grid Search CV

Although linear regression performs well, we want to try other regression algorithms to determine the best score. To accomplish this, we use Grid Search CV, which performs an exhaustive search over specified parameter values for each algorithm. By specifying the algorithms and their respective parameter grids, we can identify the best algorithm and its corresponding parameters. In this case, linear regression emerges as the winner with the highest score. The best parameter for linear regression is `normalized=False`.

Model Prediction

With our linear regression model selected, we can proceed to make property price predictions. We create a function, `predict_price`, that takes inputs such as location, square foot, bath, and BHK (bedrooms, hall, kitchen). By identifying the appropriate column index for the given location, we can set the value to 1 and predict the price based on the provided inputs.

Exporting the Model

Now that our model building procedure is complete, we need to export the artifacts required for deployment on our Python Flask server. We export the trained model using the pickle module, which allows us to serialize Python objects. Additionally, we export the column information as a JSON file. The model file size is small because it only stores coefficients, intercepts, and other parameters, without including the actual data. The JSON file contains the lowercase column names, ensuring consistency for making predictions in the future.

Conclusion

As data scientists, we go through various iterations, trying different models, performing grid search CV, and cleaning the data to arrive at an optimal model. We have successfully built a linear regression model with a score of 84%, demonstrating its suitability for predicting property prices. In the next tutorial, we will focus on developing a Python Flask server that utilizes the exported model and artifacts.