MARIA KOILALOU *JUSTINE LE BOURG*

Artificial Intelligence

Laboratory 4

Variant 1

The aim of our code is to compare the performance of three regression models Linear Regression, Random Forest Regression, and Support Vector Machine Regression for predicting house prices. The dataset used for this comparison consists of house sale prices and includes features such as the number of bathrooms, floors, waterfront view, condition, grade, year built, and year renovated.

The dataset is first loaded into a pandas DataFrame, and categorical variables are label encoded. The dataset is then split into training, validation, and test sets using the train_test_split function from sklearn. Splitting the data into different sets is important to avoid overfitting and to evaluate the model's performance on unseen data. The models are trained on the training set and their hyperparameters are tuned using the validation set. The performance of the models is evaluated using metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared.

The necessary libraries are imported including pandas, numpy, and various modules from scikit-learn, such as train_test_split for splitting the dataset into training, validation, and test sets, LabelEncoder for encoding categorical variables, LinearRegression and RandomForestRegressor for building regression models, SVR for building an SVM regression model, and mean_squared_error and r2_score for evaluating the performance of the models.

Categorical variables have been converted to numerical using label encoding. This is a crucial step for most machine learning algorithms, as they cannot handle categorical data.

In our code, the hyperparameters for the Random Forest Regressor model are n_estimators, max_depth and random_state.

n_estimators: the number of decision trees in the random forest. Increasing the number of trees can improve the model's accuracy, but also increases the computational cost.

max_depth: the maximum depth of each decision tree in the forest. Increasing the maximum depth can also improve the model's accuracy, but can also lead to overfitting.

random_state: to ensure that the same random split of the data is obtained every time you run the code.

In this code, the n_estimators hyperparameter is set to 100, and the max_depth hyperparameter is set to 20. The number of trees in the random forest is specified by the n_estimators parameter. A larger number of trees will generally increase the model's accuracy but will also increase the training time. The max_depth parameter controls the maximum depth of each tree in the forest. A deeper tree can fit the training data better, but may also overfit.

In addition to the default values for the various hyperparameters of each model, the code sets several other values by default. For example, the test size used for splitting the data is set to 0.2, meaning that 20% of the data is used for testing, while the remaining 80% is used for training and validation. The random seed used for splitting the data is set to 42, which ensures that the results are reproducible.

Linear Regression is a statistical method used to establish a linear relationship between a dependent variable and one or more independent variables. It assumes that the relationship between the variables is linear, and it seeks to find the best-fit line that describes this relationship. Linear Regression is a simple and straightforward model that is widely used for prediction and inference.

Support Vector Machine (SVM) Regression is a machine learning algorithm that aims to find the best hyperplane that separates data points with maximum margin. In regression, SVM tries to find the best-fit hyperplane that minimizes the errors between predicted and actual values. SVM does not assume any specific functional form of the relationship between the dependent and independent variables and can handle both linear and nonlinear relationships.

Random Forest Regression, on the other hand, is a machine learning algorithm that combines multiple decision trees to predict numerical outcomes. It does not assume a linear relationship between the variables, and it can handle nonlinear relationships and interactions between variables. Random Forest Regression is a more complex model that can handle a wide range of data types and is often used for prediction tasks.

OUTPUT

The output includes the validation set metrics and test set metrics for Linear Regression, SVM and Random Forest Regression models.

The validation set metrics are used to evaluate the performance of the models during the training phase, where the model is trained on a subset of the data and evaluated on another subset. The test set metrics are used to evaluate the performance of the models on a completely new and unseen dataset.

For the Linear Regression model, the validation set MSE (mean squared error) is 45640720326.07223, indicating that the average squared difference between the predicted values and the actual values in the validation set is quite high. The R-squared value for the validation set is 0.6170202161771549, which indicates that the model explains about 61.7% of the variance in the data.
For the test set, the MSE is even higher at 59089921259.47383, indicating that the model's performance is worse on the unseen data. The RMSE (root mean squared error) is also provided, which is a more interpretable measure of the error in the model's predictions, and is 243084.18553964762. The R-squared value for the test set is 0.6095930949075279, which indicates that the model explains about 60% of the variance in the data.

```
Linear Regression Model:
Validation Set Metrics:
MSE: 45640720326.07223
R-squared: 0.6170202161771549
Test Set Metrics:
MSE: 59089921259.47383
RMSE: 243084.18553964762
R-squared: 0.6095930949075279
```

For the Random Forest Regression model, the validation set MSE is lower at 42539002329.6137, indicating that the average squared difference between the predicted values and the actual values in the validation set is lower than that of the Linear Regression model. The R-squared value for the validation set is higher at 0.6430473095112741, indicating that the model explains about 64.3% of the variance in the data.

For the test set, the MSE is also lower than that of the Linear Regression model at 57276956940.72399, indicating that the Random Forest Regression model performs better on the unseen data. The RMSE is also provided, which is 239326.04735114813. The R-squared value for the test set is 0.6215713438819709, which indicates that the model explains about 62.1% of the variance in the data.

```
Random Forest Regression Model:
Validation Set Metrics:
MSE: 42539002329.6137
R-squared: 0.6430473095112741
Test Set Metrics:
MSE: 57276956940.72399
RMSE: 239326.04735114813
R-squared: 0.6215713438819709
```

For the Support Vector Machine Regression model, the validation set MSE (mean squared error) is 126699773926.71526, indicating that the average squared difference between the predicted values and

the actual values in the validation set is higher than the other models. The R-squared value for the validation set is -0.06316139802766818. The negative R-squared value in the SVM indicates that the model is performing worse than the baseline model. The baseline model is a model that predicts the average of the target variable for all the samples in the dataset. A negative R-squared value suggests that the model is not able to capture any patterns in the data and its predictions are even worse than the baseline model.

For the test set, the MSE is even higher at 161240223133.49274, indicating that the model's performance is worse on the unseen data. The RMSE (root mean squared error) is also provided, which is a more interpretable measure of the error in the model's predictions, and is 401547.28629825497. The R-squared value for the test set is -0.06531359575765072 once again is a negative number.

Overall, the Random Forest Regression model outperforms the other models on both the validation set and the test set, as evidenced by its lower MSE and higher R-squared values. The RMSE for the Random Forest Regression model is also lower than that of the Linear Regression model. Therefore, it can be concluded that the Random Forest Regression model is a better choice for predicting numerical outcomes in this particular dataset.