Introduction

Regression Model: It is used to examine the effect of one or more independent variables on the dependent variable. It is usually used to find a formula with independent equations and predict the value of the dependent variable.

As an example, stock prices and interest rates are usually inversely related. If there is an expectation in the market that interest rates will continue to rise, stock prices generally remain stable or fall. However, if the market expects interest rates to fall, stock prices rise. Therefore, if the market's expectations about interest rates can be quantified, it is possible to comment on future stock prices.

To give another example, there is a direct proportion between people's weight and height. As a person's height increases, their weight increases in parallel. By writing an equation between these two, a person's weight can be estimated by knowing only their height.

Models

Multiple Linear Regression: It is a regression model that examines the relationship between more than one independent variable on a dependent variable. The difference with Linear Regression is that there is only one independent variable in linear regression.

The formula is as follows:

$$Y=\beta0+\beta1X1+\beta2X2+...+\beta pXp+\varepsilon$$

- Y is the dependent variable,
- $X1, X2,...,Xp$ are independent variables,
- $\beta0,\beta1,\beta2,...,\beta p$ are the coefficients of the model,
- ε is the error term.

For example, it can be used to predict the price of a house, as we are doing in this project. The price of a house is determined by several independent variables such as the number of rooms, the proximity of the house to the sea, the high socio-economic level of the neighborhood where the house is located, and the number of bathrooms. Some of these independent variables have more influence on the price while others do not have

as much influence. The Multiple Linear Regression model helps us determine **the net effect of** each of these factors on price by generating coefficients that indicate **the magnitude and direction of their** effect on price.

Factors Affecting Model Accuracy:

1- Selection of Independent Variables: Inclusion of wrong or unnecessary variables in the model may increase the complexity of the model and lead to inaccurate predictions.
2- Linearity assumption: This model assumes a linear relationship between the independent variables and the dependent variable. In the absence of linearity, the model is not appropriate. For example, there may be a linear relationship between income and age, but income remains constant or decreases after a certain age. For such cases, more flexible and non-linear regression models or decision tree, support vector regression may be preferred.
3- Outlier Observations: Outlier observations can cause the model to give erroneous results, especially when the data set is small.

kNN Regression: It works by using the values of the k nearest neighbors to make a prediction. Once the k nearest neighbors are determined, it calculates the weighted average of the dependent variable. This calculated result is used as the prediction value for the dependent variable of the desired data.

kNN regression does not assume a linear relationship between independent variables and dependent variables. Thus, it provides a more flexible approach. It can also perform better when the data set is small. However, since it needs to keep the entire data set in memory, memory and computational costs can be high. It is also important to determine the value of k because it greatly affects the performance of the model. It is usually found by trial and error method.

It can be used for real estate price forecasting, weather forecasting and predicting customers' buying behavior.

Random Forest Regression: An algorithm that uses the average of multiple decision trees to predict the value of a dependent variable. Each decision tree is trained by random feature and random sample selection.

First, random subsets are created. This subset is created by repeatedly selecting samples from the dataset. Then random features are selected. According to this selected subset and features, decision trees are created using the "Decision Tree" algorithm. Essentially, Random Forest Regression is a more diverse and generalizable version of the Decision Tree algorithm with random subsets and random features. Finally, the predictions made by these decision trees are averaged and the prediction of the model is obtained.

The disadvantages of Random Forest Regression are high computational cost, large memory footprint and the presence of too many independent variables. On the other hand, its advantages are its high generalization capability, lack of linearity assumption, and being more resistant to outliers.

Random Forest Regression can perform better on complex and non-linear datasets. It can also perform better on datasets with many outlier observations due to its robustness. Due to the high computational cost, caution should be exercised when dealing with large datasets and many features.

The number of parameters (number of trees, subset size) should be taken into consideration when using it. The number of trees (n_estimators) determines how many Decision Trees to use. More decision trees usually lead to better generalization but require higher computational cost and may lead to over-fitting of the data after a certain point. This parameter is usually determined by cross-validation. Max_features determines the maximum number of randomly selected features of a decision tree. It allows each tree to be trained with different features and increases the diversity of the model. This parameter can also be determined by cross-validation. Finally, max_depth determines the maximum depth of the Decision Tree. A higher value will lead to a better fit to the data, but may also lead to over-fitting of the model. This can be determined either by manual experimentation or by cross-validation.

Support Vector Regression: It is an algorithm that adapts the SVM algorithm to regression. Its main purpose is to create a regression surface by ensuring that the dataset is within the area called the sensitivity band. Since it gives more importance to the data within this area, it is more resistant to outliers. It is generally used for non-linear relationships.

The disadvantages are the high computational cost and the difficulty in adjusting the parameters.

It can be used especially in financial forecasts where non-linear relationships are high and outliers are frequent.

Neural Network Regression: An algorithm that uses neural networks to predict the value of the dependent variable. It is a model inspired by the human brain.

Its main components are the following:

Layers: The model can have a minimum of three or more layers: input layer, hidden layer and output layer. Each layer consists of interconnected neurons.

Neurons: Each neuron receives input data, performs addition with weighted sums, uses an activation function and produces output. Thanks to the neurons in the hidden layer, the model starts to learn.

Activation Functions: Used to calculate the number at the output of each neuron. For example, the ReLu function converts negative inputs to 0 and outputs positive numbers in the same way.

Its advantages are that it can model complex and non-linear relationships and it can create extremely powerful and flexible models. The disadvantages are that such a powerful model can be overfitting, computationally expensive and difficult to parameterize.

This model is often used in image processing, natural language processing.

Gradient Boosting Regression: uses decision trees to calculate the dependent variable. Each new tree tries to reduce the errors of the previous one.

The advantages are that it is more resistant to outlier variables, can better model complex relationships and is resistant to overfitting. The disadvantages are that it is computationally expensive and the selection of parameters is difficult.

It is generally used in real estate price forecasting and financial forecasting.

PRE-PROCESSING

First of all, I saved the data to the df variable via pandas. I checked the number of columns and rows with df.shape to make sure that my dataset was big enough. Then I checked the data type of my columns with df.dtypes. I will have to check the object ones because the models can handle numeric data. I looked at how many null values there were in my data and I didn't see any null values. If I had encountered null values I would have discarded those rows unless they were in a very high percentage. I drew the graphs of the columns one by one. I did it separately to make them bigger.

There were too many unique values in the Adress column, so I decided to drop this column. I also dropped my Province column because I had the city and latitude and longitude information. Province is unlikely to be more specific when I have these, so I discarded it. In one of my duplicate files, I noticed that when I discarded both city and Province, the prediction values of the models dropped a lot. Separately, I tried to discard only the city column once, but the performance of the models dropped a lot again. In other words, I thought it would make the most sense to discard only the Province column in terms of the performance of the models, and I saw this with the experiments. I also decided to discard the Population column, because if the wages of a house will increase because a city is crowded, it will increase with the city label, so I discarded that column, thinking that it would not make significant changes. When I tried it with the experiment, it made very little difference and made the model more complex, so I decided to discard it. In addition, I discarded the outlier values. If I did not discard them,

the performance of the models, especially those that were more sensitive to outliers, decreased.

Then I divided the dataset into train and test with 25% of the dataset being test. Finally, I did Z score normalization with StandardScaler. I also noticed as a result of the experiment that the models run longer when they are tried to train without Z score normalization.

Comparison of Models and Conclusion:

Model	Mean Absolute Error (MAE)	R-Squared	Mean Squared Error (MSE)
Multiple Linear Regression	197954	0.5827	75067305749
kNN Regression	180487	0.6360	65477957088
Random Forest Regression	177905	0.6434	64150333284
Support Vector Regression	193126	0.55	80887521542
Neural Network Regression	190348	0.6038	71280928784
Gradient Regression	176174	0.6519	62617308547

When the models are analyzed, it is seen that the best prediction model is Gradient Boosting Regression and the second is Random Forest Regression. These two are similar types of algorithms and use decision trees. Such a result is probably due to non-linearities between some independent variables and dependent variables. For example, in Canada, house prices increase as you move towards the south-east. This is because it is both colder and more rural compared to the North. However, the price of the house is not completely linear with this latitude information. For this reason, algorithms using decision trees may have performed better.

The third model is the kNN regression, which estimates the price by calculating the average price of the 20 closest points. This performs well because the price is likely to be close to the average of the prices of the 20 closest houses in terms of the

characteristics of the house being predicted. This model can also capture non-linear relationships.

Support Vector Regression was the worst performing model. I think that the reason for such poor performance is exactly related to non-linear relationships and poor parameter choices. The second worst performer was Multiple Linear Regression. Since these two models have a similar structure, it is natural that they perform poorly.

As a result, I observed that Linear Regression underperformed due to the presence of some non-linear variables. However, I found that models using decision trees performed better. Decision Trees can be more effective in predicting house prices due to their better representation of non-linear relationships and flexible modeling structure.