Report ~ Prediction of Canadian House Prices

1)What Regression Is? I can explain this very formally but it wouldn't help anyone else to really grasp what regression is so I want to use some analogy to make sure everyone who read this report will understand what regression really is and how it works.

Let's imagine that we collected data by asking a lot of student about what their IQ is, how much they slept, how much they studied, what is their name and how they felt out of 10 that day and their GPAs.

Regression analysis in this context would involve examining the relationship between these variables and GPA. We want to understand how IQ, sleep duration, study hours, their name and daily mood rating collectively influence a student's GPA.

To create an analogy: Think of regression analysis as akin to baking a cake.

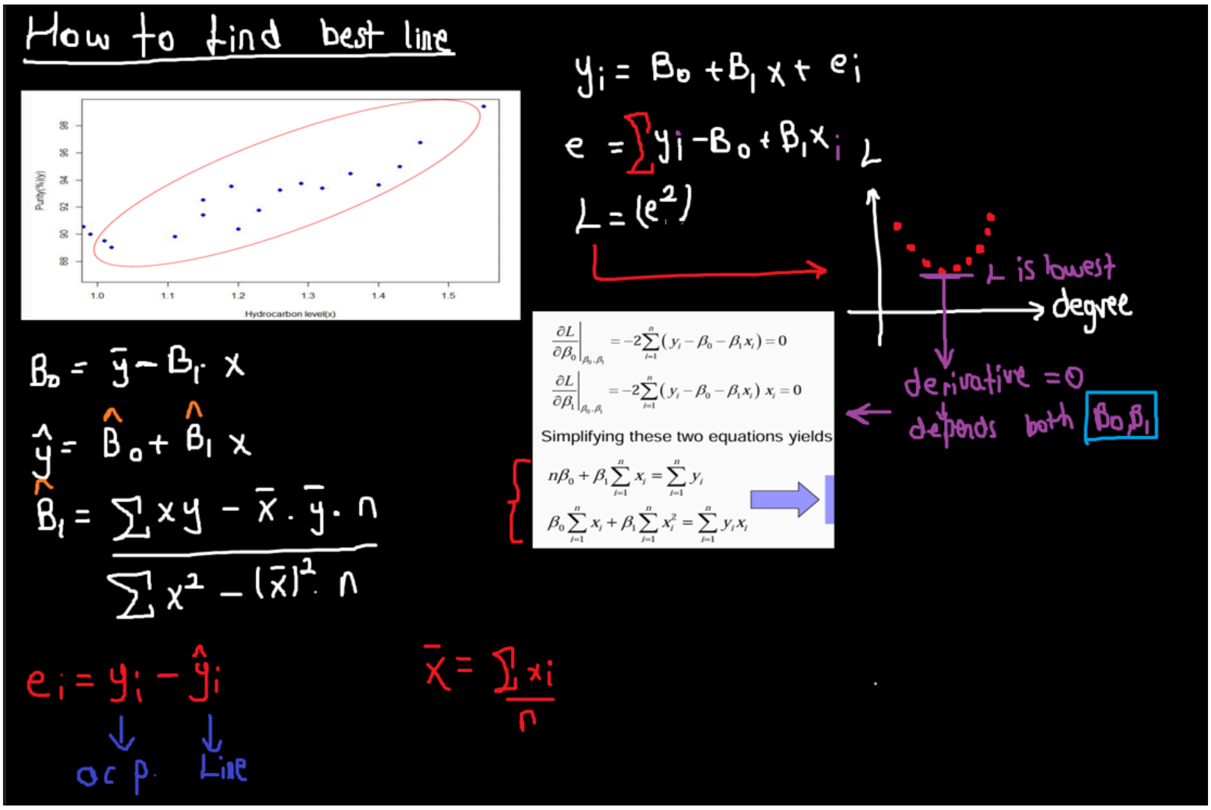
Ingredients:
IQ, sleep duration, study hours, names and daily mood rating are like the ingredients used in baking the cake. Each ingredient contributes to the final outcome.
Recipe:
Regression analysis is like following a recipe. We mix the ingredients (variables) in specific proportions to create the cake (GPA prediction model).
Outcome:
The cake (GPA prediction model) represents the relationship we've established between the ingredients (variables) and the final product (GPA). Just as a cake's taste and texture depend on the amounts of flour, sugar, eggs, etc. a student's GPA depends on their IQ, sleep duration, study hours, and daily mood rating.
Tasting the Cake:
We can "taste" the cake by testing our regression model. We see how well it predicts a student's GPA based on their IQ, sleep, study habits, and mood.
Adjusting the Recipe:
If the cake doesn't taste quite right, we may need to adjust the recipe (refine the regression model) by tweaking the amounts of ingredients (variables) or adding new ones. For example your name has nothing to do with your grades.

2)Explanation of models

a) Multiple Linear Regression: Linear regression is a way to find a straight line that best fits a set of data points. Imagine you have a bunch of data points scattered on a graph. Multiple Linear regression helps you draw a line through these points in such a way that the line comes as close as possible to each point.

Multiple Linear Regression can be applied in various dimensions. For instance, in 2D, it's represented by the equation y = B0+B1x, defining a line. In 3D, it's y = B0+B1x+B2z, forming a plane. The dimensionality can extend up to 10D or more, depending on the number of features correlated with the target variable we aim to predict.

Here is mathematically how you can find the best fitting line from scratch from my own notes:



How to find best line

$y_i = B_0 + B_1 x + e_i$

$e = \sum y_i - B_0 + B_1 x_i$ L

$L = (e^2)$

L is lowest

degree

derivative $= 0$

depends both $\boxed{B_0 B_1}$

$$\left.\frac{\partial L}{\partial \beta_0}\right|_{\beta_0,\beta_1} = -2\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i) = 0$$

$$\left.\frac{\partial L}{\partial \beta_1}\right|_{\beta_0,\beta_1} = -2\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)\, x_i = 0$$

Simplifying these two equations yields

$$\begin{cases} n\beta_0 + \beta_1 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i \\ \beta_0 \sum_{i=1}^{n} x_i + \beta_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} y_i x_i \end{cases}$$

$B_0 = \bar{y} - B_1 \cdot x$

$\hat{y} = \hat{B}_0 + \hat{B}_1 x$

$$\hat{B}_1 = \frac{\sum xy - \bar{x} \cdot \bar{y} \cdot n}{\sum x^2 - (\bar{x})^2 \cdot n}$$

$e_i = \underset{\text{oc p.}}{y_i} - \underset{\text{Line}}{\hat{y}_i}$

$\bar{x} = \frac{\sum x_i}{n}$

Multiple regression is well-suited for datasets where you have more than one independent variable (also called features or predictors) and one dependent variable (the variable you want to predict) and their relationships are linear.

b) KNN

We can express KNN in just 1 sentence ‘You are the sum of the k people around you’. Imagine you have a group of friends, each with their own unique traits like age, height, and hobbies. Now, when a new person arrives, you want to figure out which of your friends they're most like. KNN helps you do this by looking at the characteristics of the new person and comparing them to those of your friends. It measures how similar the new person is to each friend, considering factors like age, height, and interests. Then, it identifies the "K" friends who are most similar to the new person. Once it finds these nearest neighbors, it can make predictions or decisions based on what those neighbors are like. For instance, if the closest friends all enjoy playing basketball, you might guess that the new person also likes basketball.

Sometimes people fake their identities :) which can lead to overfitting. In such cases, we need to compare them to more people(larger k) to detect them accurately.

K-Nearest Neighbors (KNN) works great for smaller datasets with no clear patterns, especially when classes are easily distinguishable and when the relationship between features and the target variable is nonlinear.

Also when feature number is very high it can lead to Curse of dimensionality - Wikipedia.

KD-Trees is a way to make KNN algorithm much more efficient.

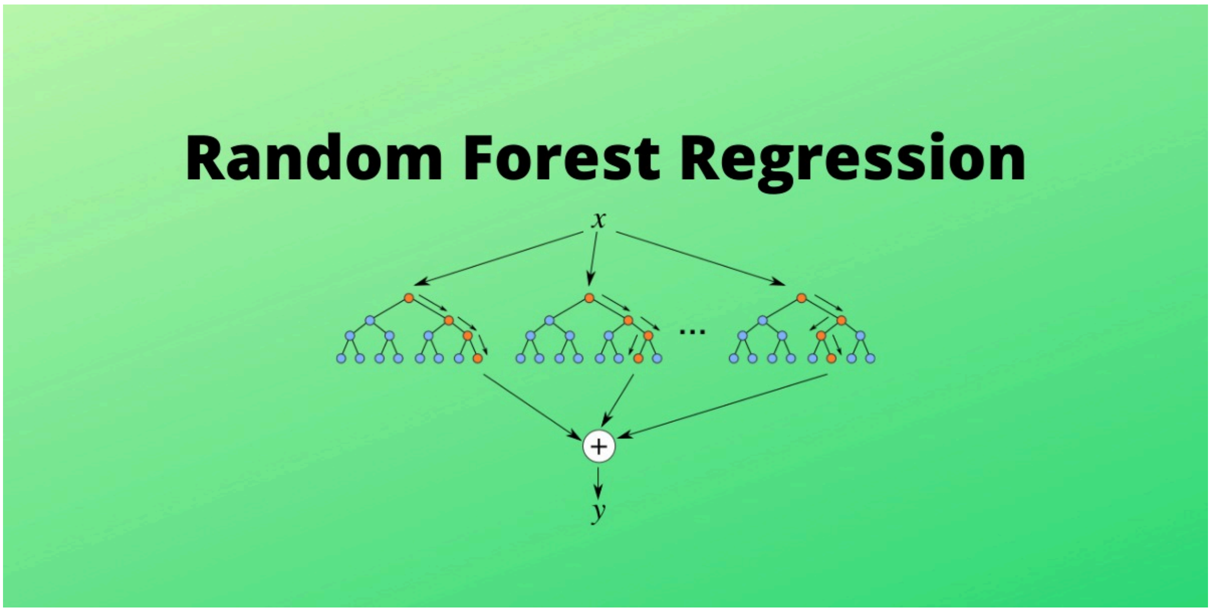
c) Random Forest Regression

We can express Random Forest Regression such imagine you’re gonna buy a computer so you are researching all throught internet from different resources to make the best decision.

Random Forest Regression entails a compilation of hundreds of Decision Trees. These trees are combined together using bootstrapped data, selecting different random attributes based on their effectiveness.

In essence, Random Forest Regression combines the predictions of many "trees" (or friends) to make a more reliable forecast. Each tree focuses on different aspects of the data, and by considering all of them together, Random Forest Regression helps to make a more accurate prediction.

Random Forest Regression is great for data with complicated relationships between features and the target. It handles large datasets and works well even when there are different types of data. Plus, it's robust to outliers and noise, making it reliable for many real-world problems.



d) SVM Regression

Support Vector Machine (SVM) Regression is a smart tool for predicting outcomes, especially when the relationships between different factors are a bit complicated. It works by finding the best line or curve to fit the data points, kind of like drawing a line through scattered dots on a graph.

The cool thing about SVM Regression is that it's really good at handling lots of different factors, even when there are a bunch of them. Also it's not easily thrown off by weird data points or noise in the data, which makes it reliable even in messy situations.

So, when you have a bunch of data and you want to predict something, like the price of a house based on its size and location, SVM Regression can step in and help you find the best prediction line or curve to make sense of it all.

Unless KNN, SVM Regression is life-saver when dealing with high dimensional data.

e) Neural Network Regression(MLP Regression)

Neural Networks are like powerful problem-solving tools inspired by the human brain. They consist of interconnected nodes, or "neurons," organized into layers. These layers typically include an input layer, one or more hidden layers, and an output layer.

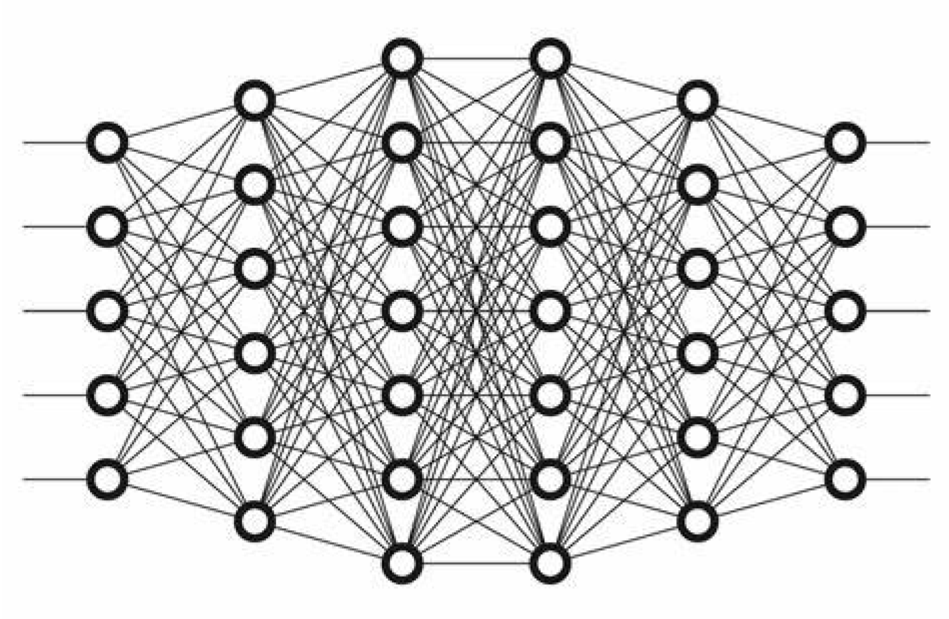
The process begins with the input layer, which receives the data to be processed. Each neuron in the input layer represents a feature or attribute of the data. The information then flows through the hidden layers, with each neuron performing a mathematical operation on the input and passing the result to the next layer. The final output layer produces the model's prediction or classification.

During training, the Neural Network adjusts the connections between neurons to minimize the difference between its predictions and the actual outcomes. This process, known as backpropagation, involves iteratively updating the weights assigned to each connection based on the error between the predicted and actual values.

Neural Networks are useful for a wide range of data types and tasks. They excel in scenarios where the relationships between variables are complex or nonlinear, making them suitable for tasks such as image recognition, natural language processing, and time-series prediction. Additionally, they can handle high-dimensional data and large datasets effectively, making them valuable tools in fields like computer vision, speech recognition, and financial forecasting.

A recommended playlist for understanding Neural Network better: [Peki sinir ağı nedir? | Bölüm 1, Derin Öğrenme (youtube.com)]

Understanding neural networks requires understanding vectors, gradient function and high dimensional spaces. It requires a solid Calculus background.



f)Gradient Regression

Gradient Boosting Regression is a powerful machine learning technique used for regression tasks. It belongs to the ensemble learning family, where multiple models are combined to improve predictive performance. Specifically, Gradient Boosting Regression works by combining a collection of weak prediction models, typically decision trees, into a single strong predictor. It reminds us to Random Forest

The process begins by creating a simple model, often a single decision tree, to make predictions on the data. The algorithm then focuses on the instances where this initial model performs poorly and builds a new model that specifically targets these areas of weakness. This iterative process continues, with each new model being trained to correct the errors made by the ensemble of previous models.

Unlike other ensemble methods, such as Random Forests, where the individual models are built independently, Gradient Boosting Regression constructs models sequentially, with each subsequent model learning from the mistakes of its predecessors. This sequential nature allows Gradient Boosting Regression to refine its predictions over multiple iterations, ultimately producing a highly accurate and robust final model.

Data Preprocessing:

Handling Unique Values: The Address column contains numerous unique values, so it is necessary to drop it.

Handling Null Values: I checked for null values and there is none.

Adding Quadratic Features: Although not based on any scientific technique, I included quadratic features as the number of rooms in a house increases, signifying a larger mean house size in real-life scenarios.

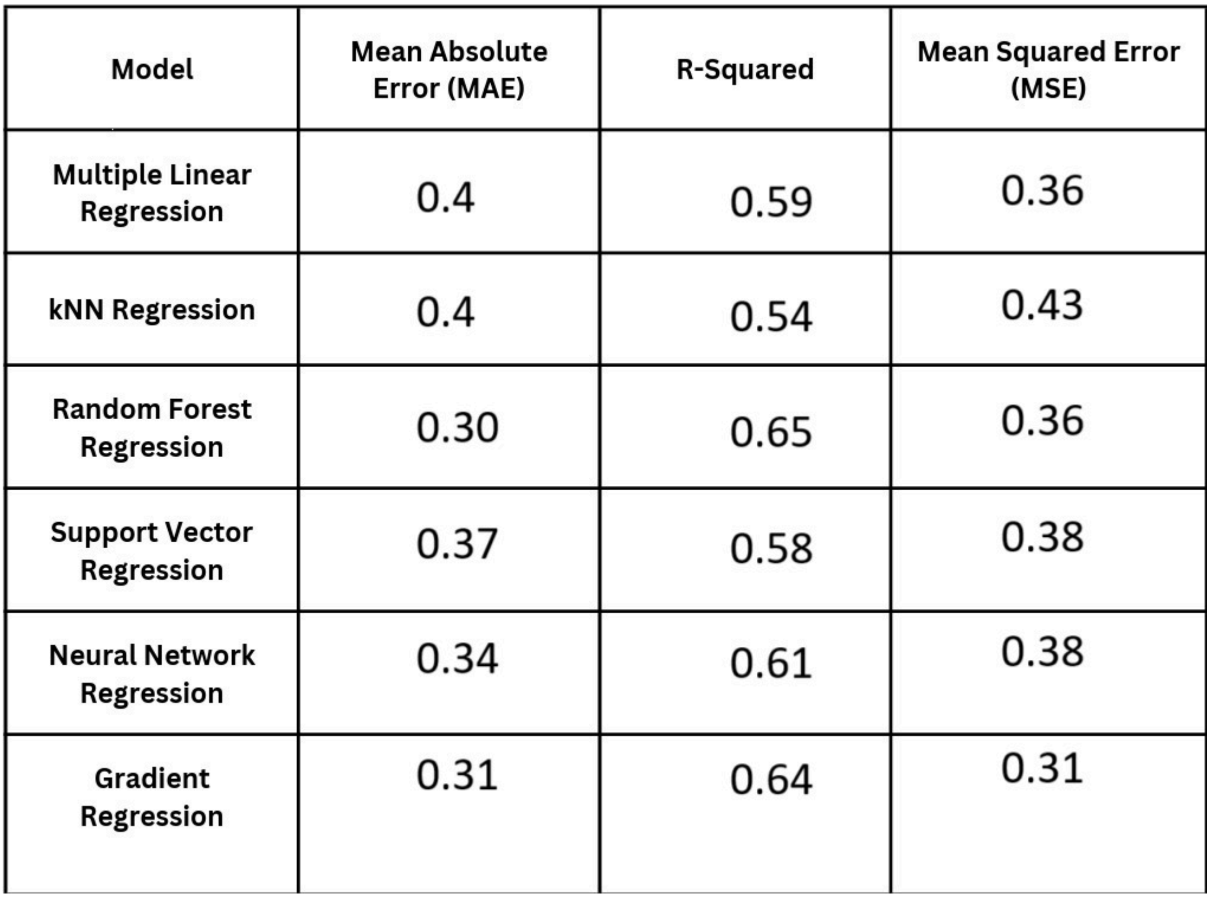
Handling Outliers (Part 1): I utilized the ±2 standard deviation method to address outliers.

Realizing the Effect of a Column & Handling Encode: The city is likely correlated with the price, so I encoded city names with the mean prices. Also we need to encode province.

Implementing Z-Score: I implemented the Z-score method for standardization.

Handling Outliers (Part 2): I expanded the outlier coverage by using the ±3 standard deviation range from the mean.

Final Table Evaluation and Conclusion



Model	Mean Absolute Error (MAE)	R-Squared	Mean Squared Error (MSE)
Multiple Linear Regression	0.4	0.59	0.36
kNN Regression	0.4	0.54	0.43
Random Forest Regression	0.30	0.65	0.36
Support Vector Regression	0.37	0.58	0.38
Neural Network Regression	0.34	0.61	0.38
Gradient Regression	0.31	0.64	0.31

The main reason for the following explanations is the manner in which data is dispersed: There isn't much linear relation, but there are clear patterns.

Random Forest Regression:

With a low MAE of 0.30 and MSE of 0.36, the random forest model demonstrates strong predictive accuracy and generalization ability. Its high R2 value of 0.65 indicates that a significant portion of the variance in the target variable is explained by the model. This suggests that the random forest algorithm is effective in capturing the underlying patterns and trends present in the data spread.

Gradient Boosting Regression:

Similarly, the gradient boosting model exhibits low MAE (0.31) and MSE (0.31), indicating accurate predictions and a good fit to the data spread. Its R2 value of 0.64 suggests that it explains a considerable amount of variance in the target variable. This indicates that the gradient boosting algorithm is robust in handling the variability present in the dataset.

Neural Network Regression:

While the neural network model also shows low MAE (0.34) and MSE (0.36), its R2 value of 0.61 suggests that it explains slightly less variance in the target variable compared to the random forest and gradient boosting models. However, it still performs well in capturing the spread of the data and making accurate predictions.

Multiple Linear Regression, Support Vector Regression, and K-Nearest Neighbors Regression:

These models demonstrate higher MAE and MSE values, along with lower R2 values, compared to the ensemble methods (random forest, gradient boosting, and neural network). This indicates that they may not capture the variability in the data spread as effectively as the ensemble methods, leading to less accurate predictions.

The best performing model is Random Forest Regression. It has the lowest MAE (0.30), the highest R2 (0.65), and a relatively low MSE (0.36). **The reason for this is the clear patterns in the distribution of data.**

The worst performing model is K-Nearest Neighbors (KNN) Regression. It has the highest MAE (0.40), the lowest R2 (0.54), and the highest MSE (0.43) among the models. **The reason for this is probably the inclusion of quadratic features, which causes a high dimensionality problem.**

In summary, models like random forest, gradient boosting, and neural network regression perform well in capturing the spread of the data and making accurate predictions, as evidenced by their low MAE and MSE values and high R2 values. These models effectively handle the variability present in the dataset, making them suitable choices for predictive modeling tasks.