Linear model vs Decision Trees

Linear models and Decision Trees are both popular models used for Regression as well as Classification problems. When to use which model? Trying both models and choosing the best one works but is a naive approach.

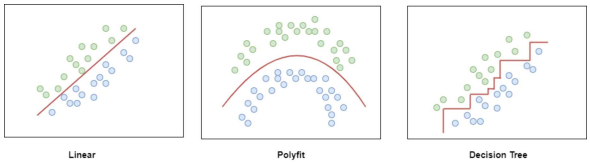
How to decide like a pro data scientist?

The answer lies in your Dataset!

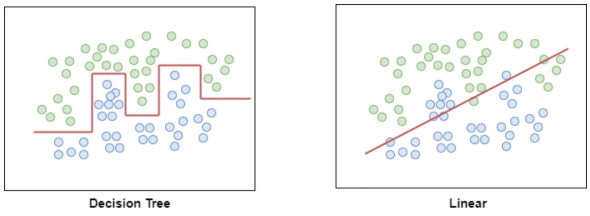
Is it that simple? Let us try to answer the following questions and see which model fits better based on your answers!

Is my data Linearly separable?

Classification problem: the output is a distinct set of classes eg. diabetic or not diabetic

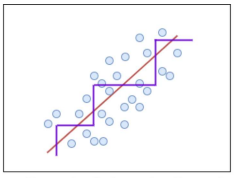


Logistic Regression models can only handle data with linear relationships. Can the data be separated using a line or at least a curvy line? Then logistic regression model works better for them while Decision Tree can get computationally more expensive.

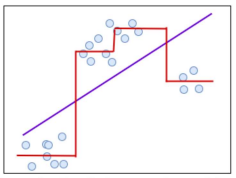


Decision trees are non-linear models and can handle non-linearly separable data well. Decision trees also select the splitting criteria known as feature selection implicitly.

Regression Problem: the output is continuous eg. stock prices



Linear Regression fits better than Regression Tree



Regression Tree fits better on clustered data

Decision trees are less suitable for regression tasks that require predicting a continuous output. However, if the data is highly clustered, then decision trees perform better than the Linear regression model.

Are the majority of features in my data Continuous or Categorical?

If **Categorical:**

Decision Trees can handle both numerical (continuous) and categorical data well while Linear models require preprocessing categorical variables to convert them to numerical variables. Linear models cannot handle pure Categorical (string) features.

Label Encoding

Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

→

One Hot Encoding

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

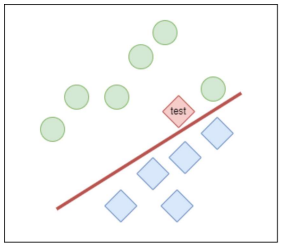
One of the techniques for making categorical features numerical is one-hot encoding. However, for a large number of categorical features, a lot of dummy variables are created. This can increase the dimension of the data and also make it highly sparse and hence computationally expensive. For a small dataset, this also leads to underfitting of the data due to a large number of features.

If **Continuous:**

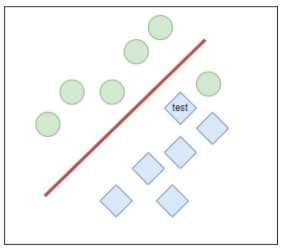
Decision trees sort numerical features in ascending order and compute the information gain for each value as the threshold point. This is computationally inefficient when the dataset contains a large number of continuous features. Linear models would be a better choice in this case.

Does my data contain outliers?

Outliers are values that are very different from the rest of the values in a dataset.

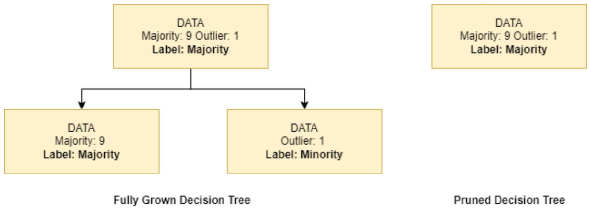


Bad Fit



Good Fit

Linear models tend to push the decision boundary towards the outliers leading to a bad fit and poor performance on test data. Outliers need to be manually detected and removed before using Linear models for regression as well as classification.

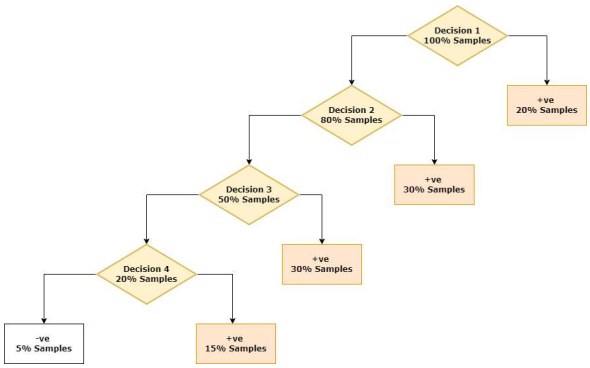


Decision trees can easily isolate outliers as separate leaf nodes. No preprocessing is required. However, this may lead to one leaf node created for every outlier. This problem is minimized in a pruned (not fully grown) tree since the class label corresponds to the class of the majority of samples.

Is my data highly biased?

Data is imbalanced/skewed if the number of classes is not evenly distributed. Such a dataset is also called biased data. For example, 95% of my data are positive and 5% are negative examples.

Both Linear models and Decision Trees do not perform well on imbalanced data. Improving class balance using oversampling (randomly generating minority samples) or undersampling (reducing the majority samples) is generally a good practice.



However, if a decision tree is allowed to fully grow (no pruning) it can handle imbalanced data, at the cost of overfitting, causing biased trees.

In Linear models, assigning “class_weight” by giving different weights to both the majority and minority classes also helps overcome bias in the data. This does not cause overfitting but requires the prior computation of class weights.

Does my data require a stable model?

This question may require a bit of thought. What is stability in a model? How a model handles a new data point or example is measured using the following:

Bias error: Our assumptions about the target function cause bias error. The more assumptions we make about the target function (the more restrictions we put on them), the higher the bias error. Because we have more rules on target functions, models with high bias are less flexible. A high bias error causes *underfitting*.

Variance error: Variance error is the difference between how a target function looks in different training sets. Changing a few samples in the training set will not significantly alter models with low variance error. Even little alterations to the training data might impact models with a large variance. A high variance error causes *overfitting*.

Linear models:

$$y = w_0 x + \beta$$

Clearly, the above linear function imposes a limit on the target function and hence simple linear regression model has a **High Bias**. Also, ***prone to underfitting*** the data. For example, this function cannot fit a quadratic curve well. However, a second-degree polynomial will have a low bias for the same curve. Thus, to reduce bias, a more complex model or reducing the number of features is required.

With sufficiently large training data, Linear models have a **low variance**. *Linear models are thus robust to less biased, high variance (heterogenous), linear data with lesser features, and/or large datasets.*

Decision Trees:

Almost no assumptions are made about the target function, hence Decision Trees have **Low Bias**.

But they are very sensitive to changes in the data. Depending on how the data is changed, a completely different tree could be made. Thus, Decision Trees have **high variance** and are ***prone to overfitting*** the data. To reduce variance, a more heterogenous dataset and methods such as Bagging and Boosting can be used.

Decision Trees are thus robust to highly biased, low variance (samples not too different from each other), non-linear data with more number of features, and/or small datasets.

Does my data contain missing values?

Real-world data is rarely perfect. It may contain a few or more missing values.

Logistic Regression cannot handle missing data; these values must be imputed using either mean, mode, or median. For a small number of missing values, Linear models fit well with imputation without the need for a more complex model like a Decision Tree.

However, if your dataset contains a large number of missing values, it may not be a good idea to impute them, as doing so changes the distribution of the data. **Decision Trees can account for missing values** without the need for any pre-processing or changes to data.

. . .

Summarizing

Data	Linear Model	Decision Tree
Is data linear?	Requires Linear relationship in data	Works better for non-linearly separable data
Is Data clustered?	Poor-fitting in regression tasks with clustered data	Works well with highly clustered data
More numerical features?	Less computationally intensive with one-hot-encoding	Can handle both numerical and categorical data. Highly computationally inefficient
More Categorical features?	Requires feature engineering. Computationally expensive due to sparsity in data.	Requires no feature engineering. Can handle both numerical and categorical data.
Outliers?	Requires to be removed to prevent poor performance on test data.	Can handle outliers as separate leaf nodes. Can improve accuracy using pruning.
Bias	High. Cannot fit complex data. Can be mitigated using complex models or lesser features.	Low. Makes no assumptions about the target function.
Variance	Low. Can handle heterogeneous data.	High. Cannot handle changes in data. Can be mitigated using bagging/boosting.
Missing Values?	Need to be imputed using mean, mode, or medium before training.	Requires no pre-processing. Can handle missing values.

Linear models work better for a large dataset that is linearly separable, contains a small number of features, is pre-processed for outliers and missing values, and is heterogenous (high variance in data).

Decision Tree works better for a small dataset that is non-linear, contains a large number of features (numerical/categorical), contains a large number of missing values and outliers, and is not heterogenous (low variance in data).