Decision Tree Regressions

What is CART?

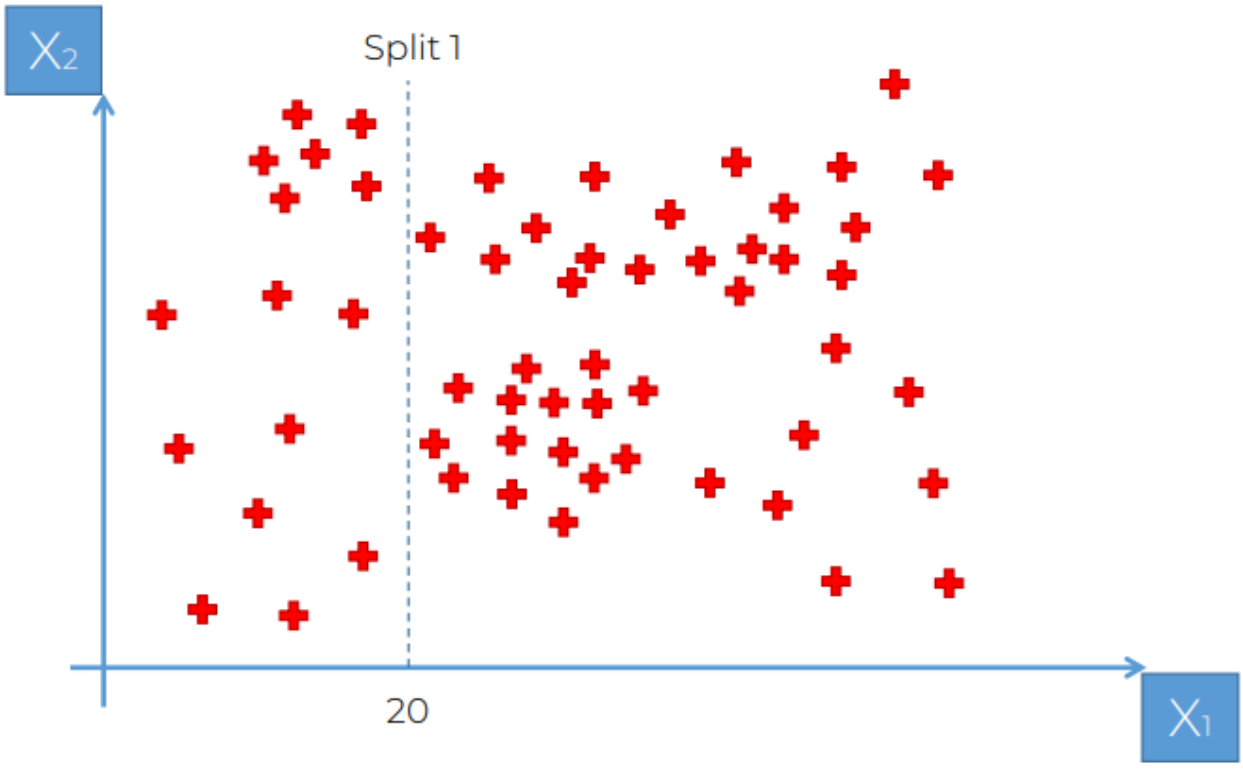
- CART stands as an acronym for classification and regression trees.

How do Decision Tree Regressions Work?

- Essentially, when all data points are plotted on a scatter plot, the algorithm will perform splits on the data. Where this split occurs is determined by the `information entropy`.
- Information Entropy essentially checks whether this split that we have increases the information we have about our points. The algorithm knows when to stop the split when a minimum of information is reached, once it determines that it cannot add more information to the setup it stops. The Algorithm is capable of performing the splits using information entropy on its own.

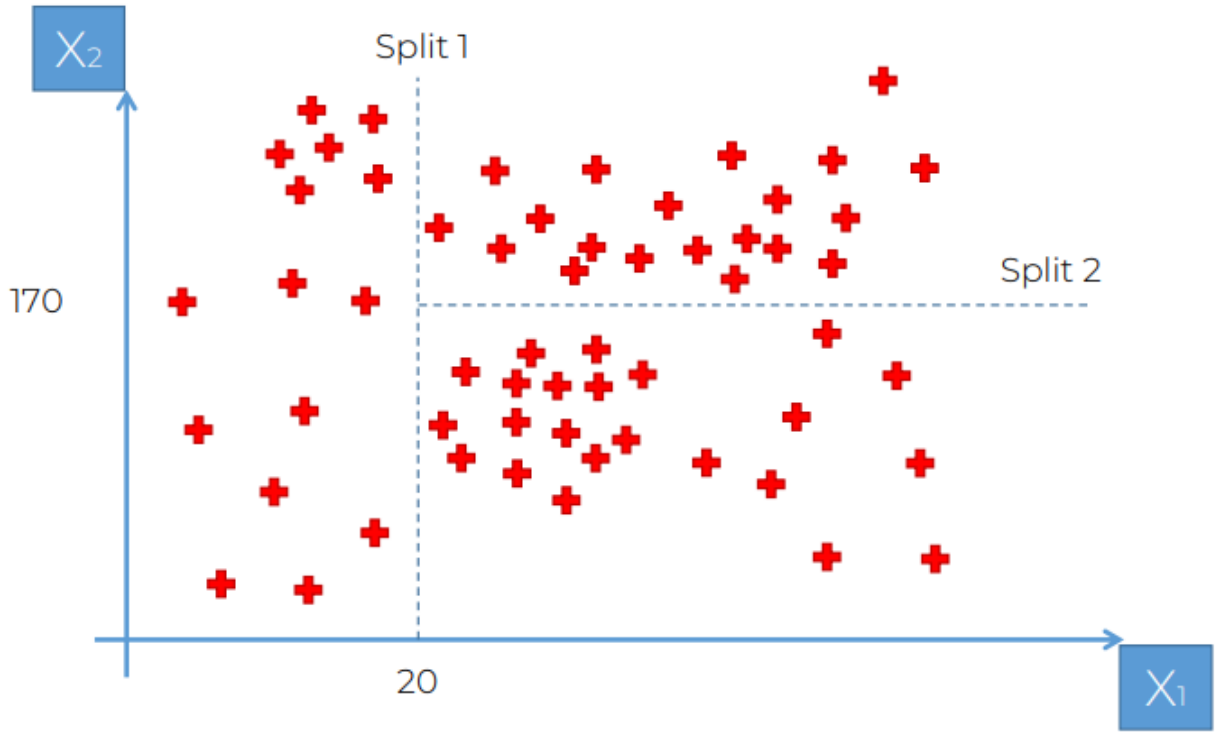
Example Decision Tree Splitting Flow:

1. Algorithm determines first split in a scatterplot:



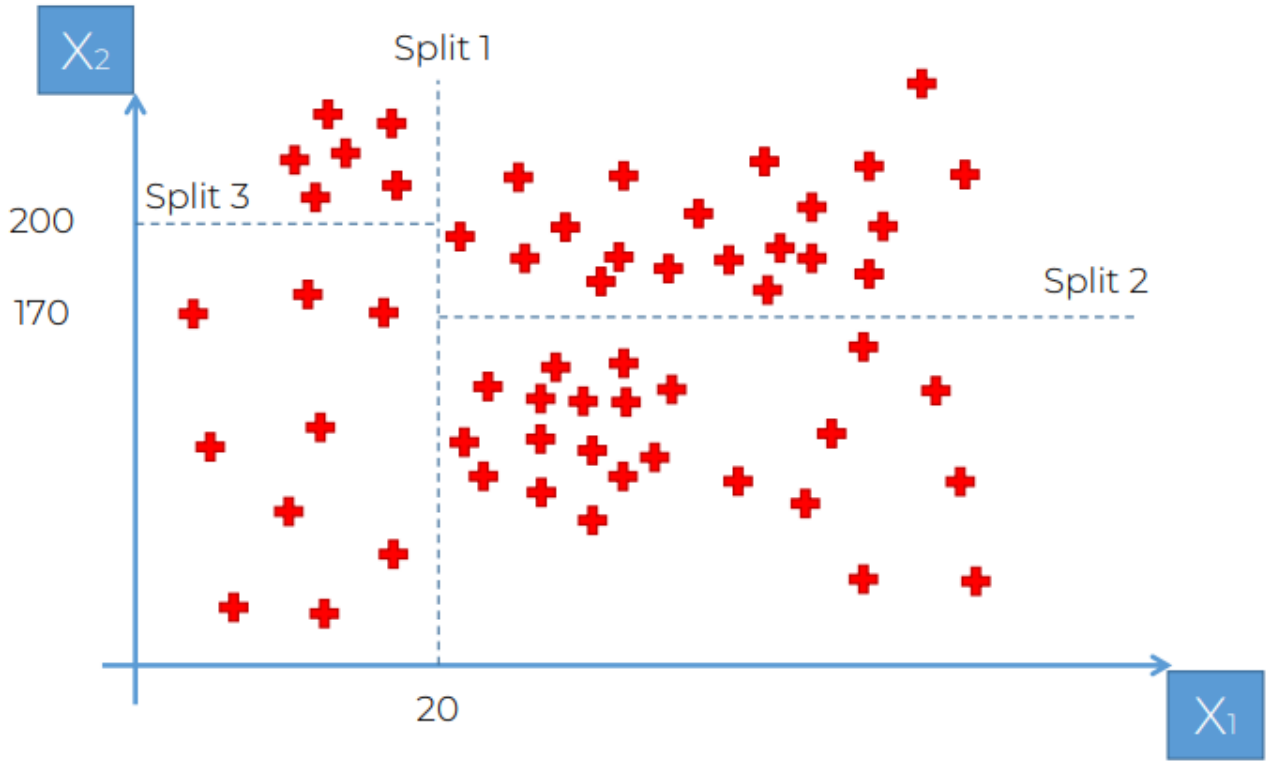
In this case, we have have the first split, where $X_1 < 20$

2. Second split occurs:



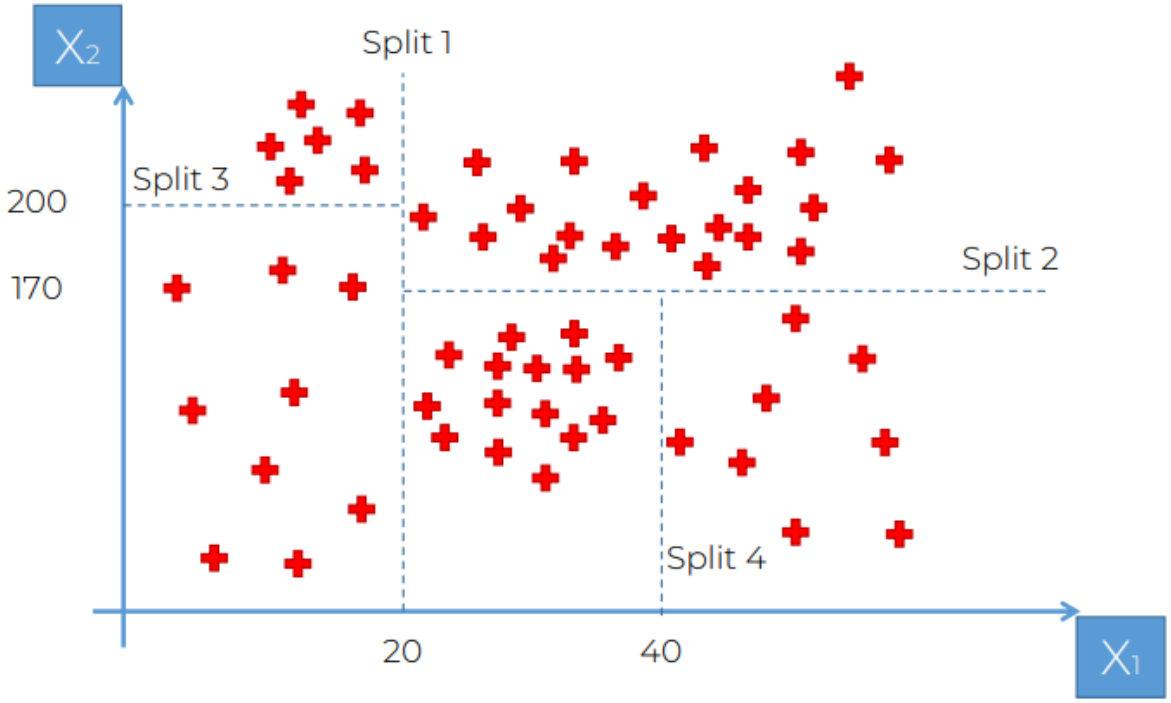
The second split, where $X_2 < 170$.

3. Third split occurs:



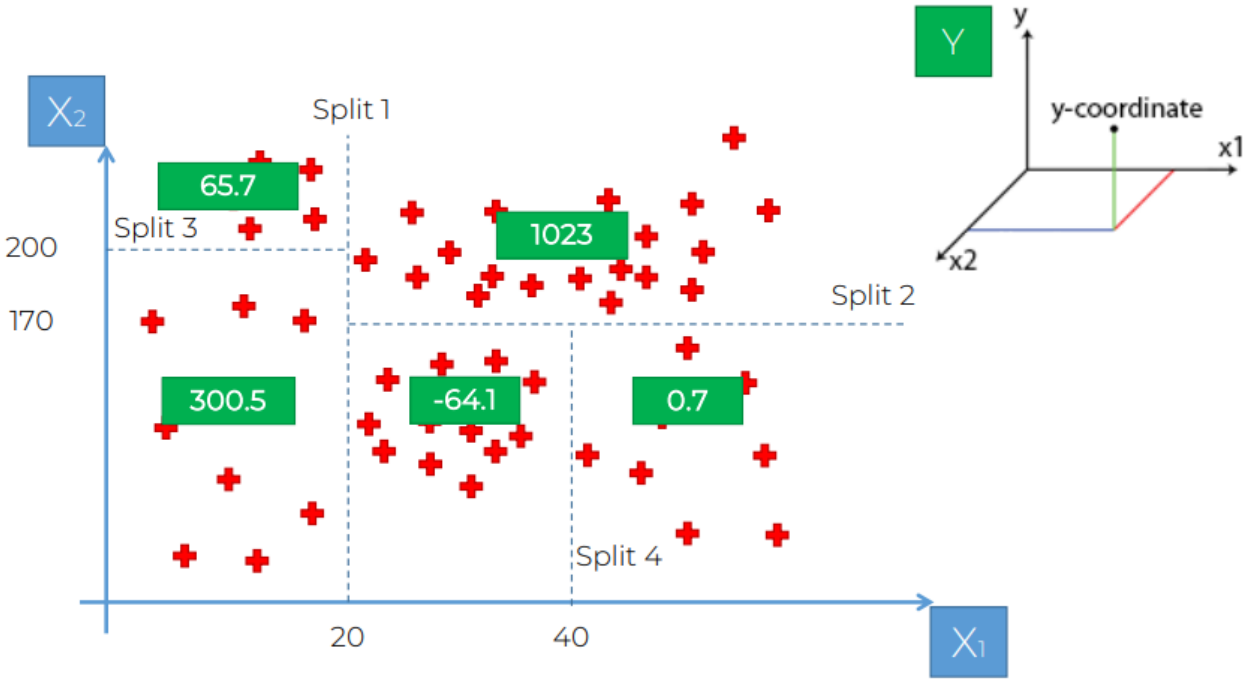
The third split, where $X_2 < 200$

4. The fourth split:

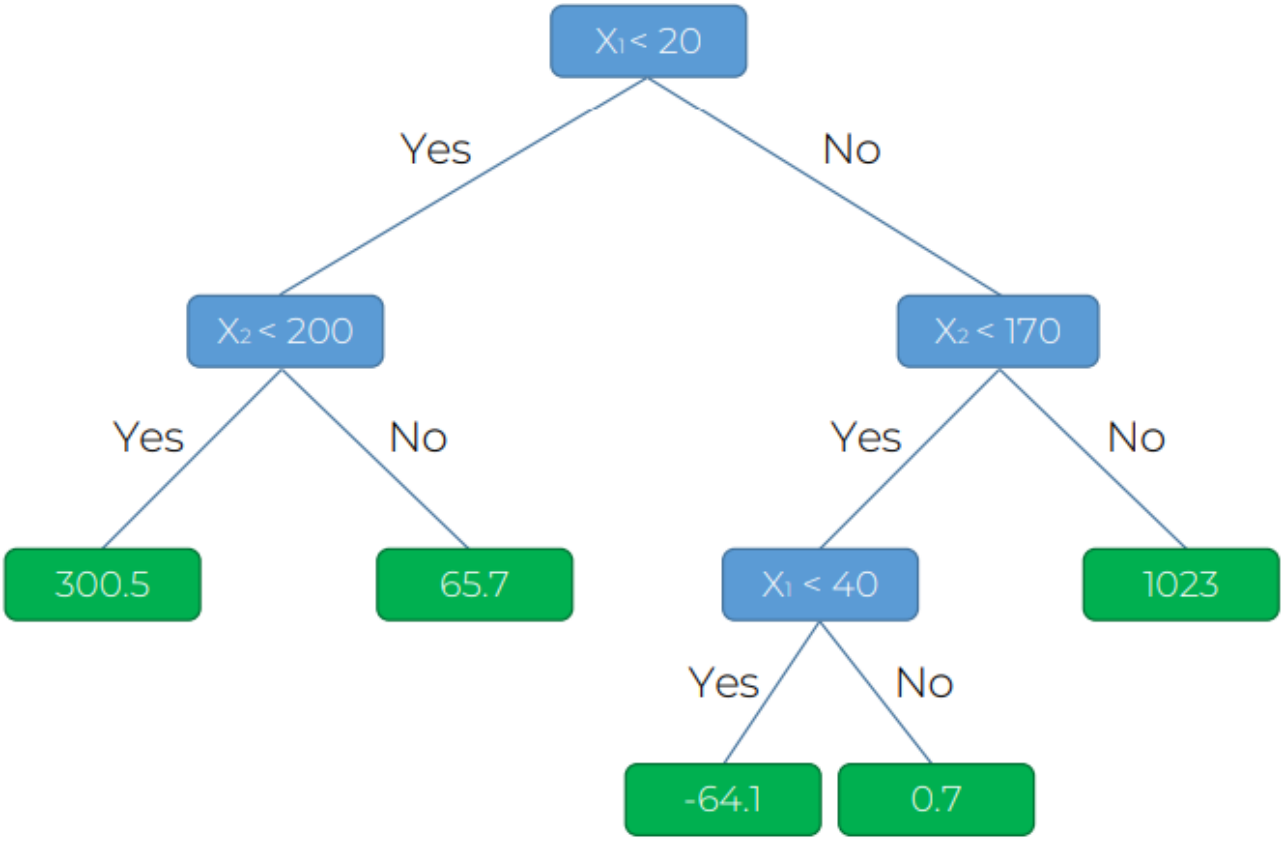


The fourth split, $X_1 < 40$

5. For the model to predict the dependent value from the two independent value, based on splits and decisions, the average from each leaf is returned as the predicted value. Lets say for example, each leaves average is as show:



6. So we can fill out the decision tree as such:



This tree flow diagram helps us understand how Decision Tree Regression Models gets accurate predictions.

Example:

We are using the same example referenced in the Polynomial Regression, where we have a non-linear relation between the matrix of features `Level`, relating to the position, and the dependent matrix of feature, the `salary`. In this case, we do not need feature scaling (in Decision Trees)

Applying Decision Tree Regressions

- **Import Libraries**

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

- **Import Dataset**

```
dataset = pd.read_csv('Position_Salaries.csv')
X = dataset.iloc[:, 1:-1].values
```

```
y = dataset.iloc[:, -1].values
```

- **Training the Decision Tree Regression model on the whole dataset**

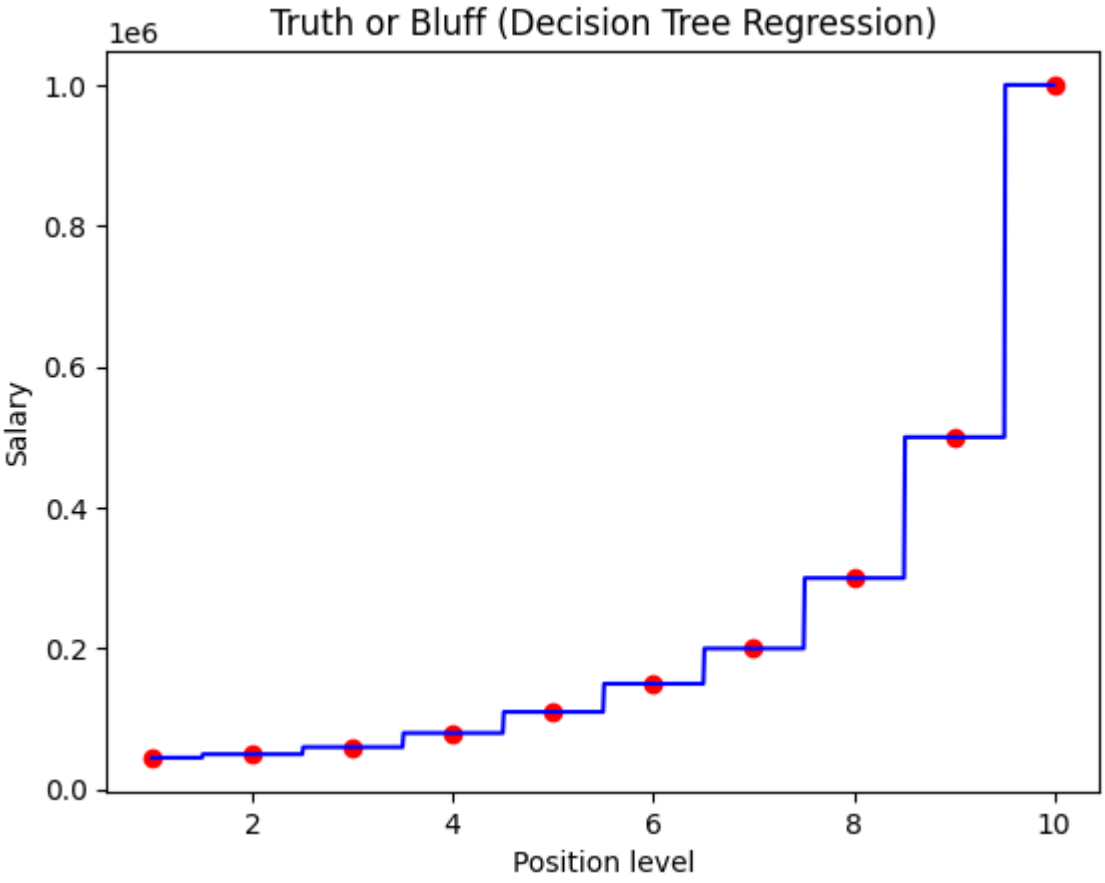
```
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor(random_state = 0)
regressor.fit(X, y)
```

- **Predicting a new result**

```
regressor.predict([[5]])
```

- **Visualizing the Decision Tree Regression results (higher resolution)**

```
X_grid = np.arange(min(X), max(X), 0.01)
X_grid = X_grid.reshape((len(X_grid), 1))
plt.scatter(X, y, color = 'red')
plt.plot(X_grid, regressor.predict(X_grid), color = 'blue')
plt.title('Truth or Bluff (Decision Tree Regression)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```


Truth or Bluff (Decision Tree Regression)
1e6
1.0
0.8
0.6
0.4
0.2
0.0
Salary
2
4
6
8
10
Position level