Day49_Decision_Tree_Regressor

July 18, 2025

Today we are learning about the **Decision Tree Regression model**

— a supervised machine learning algorithm used for both classification and regression tasks. It works by splitting the data into smaller and smaller parts based on certain rules, much like how a flowchart works.

In this notebook, we'll explore:

- What is a Decision Tree?
- How it works
- Its use cases
- Python implementation with visualization

  Note: Although Decision Tree is often used for **classification**, this is a regression example due to the dataset's nature (numerical salary output).

What is Decision Tree?

A **Decision Tree** is a supervised machine learning algorithm used for both **classification** and **regression** tasks.

It mimics human decision-making by splitting data into smaller subsets based on feature values.

Why Decision Tree Regression?

Even though Decision Trees are mainly used for **classification**, we can use them for regression tasks where the target variable is **continuous**, like salary.

How it Works:

- The algorithm asks **yes/no** (or **true/false**) style questions at each node.
- Each decision splits the dataset into branches based on features.
- The tree continues splitting until a stopping condition is met (e.g., max depth, pure leaf nodes).
- Final prediction is based on the **leaf node** reached.

Key Features:

- Simple and easy to understand (tree-like structure)
- Handles **numerical and categorical** data
- Can be prone to **overfitting**, especially on small datasets
- Works well when data has clear rules and patterns

Use Cases

- **Medical**: Disease diagnosis
- **Business**: Credit risk analysis, customer segmentation
- **Retail**: Product recommendation, sales forecasting
- **Education**: Student performance prediction
- **Agriculture**: Crop yield prediction

Types of Decision Trees:

- **Classification Tree**: When the target variable is **categorical** *(e.g., spam vs. not spam)*
- **Regression Tree**: When the target variable is **numerical** *(e.g., predicting house prices)*

1 Import Libraries

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

2 Load Dataset

[2]:
```
dataset = pd.read_csv(r"C:\Users\Lenovo\Downloads\emp_sal.csv")
dataset
```

[2]:

	Position	Level	Salary
0	Jr Software Engineer	1	45000
1	Sr Software Engineer	2	50000
2	Team Lead	3	60000
3	Manager	4	80000
4	Sr manager	5	110000
5	Region Manager	6	150000
6	AVP	7	200000
7	VP	8	300000
8	CTO	9	500000
9	CEO	10	1000000

3 Feature Selection

[3]:
```
X = dataset.iloc[:, 1:2].values  # Level (2D)
y = dataset.iloc[:, 2].values    # Salary
```

4 Train the Decision Tree Regressor

[4]:
```
from sklearn.tree import DecisionTreeRegressor

dt_reg = DecisionTreeRegressor(random_state=0)
dt_reg.fit(X, y)
```

[4]:
```
DecisionTreeRegressor(random_state=0)
```

5 Predict Salary

[5]:
```
# Predict salary for level 6 (Region Manager)
dt_reg.predict([[6]])  # Output: array([150000.])
```

[5]:
```
array([150000.])
```

[6]:
```
# Predict salary for level 9 (CTO)
dt_reg.predict([[9]])  # Output: array([500000.])
```

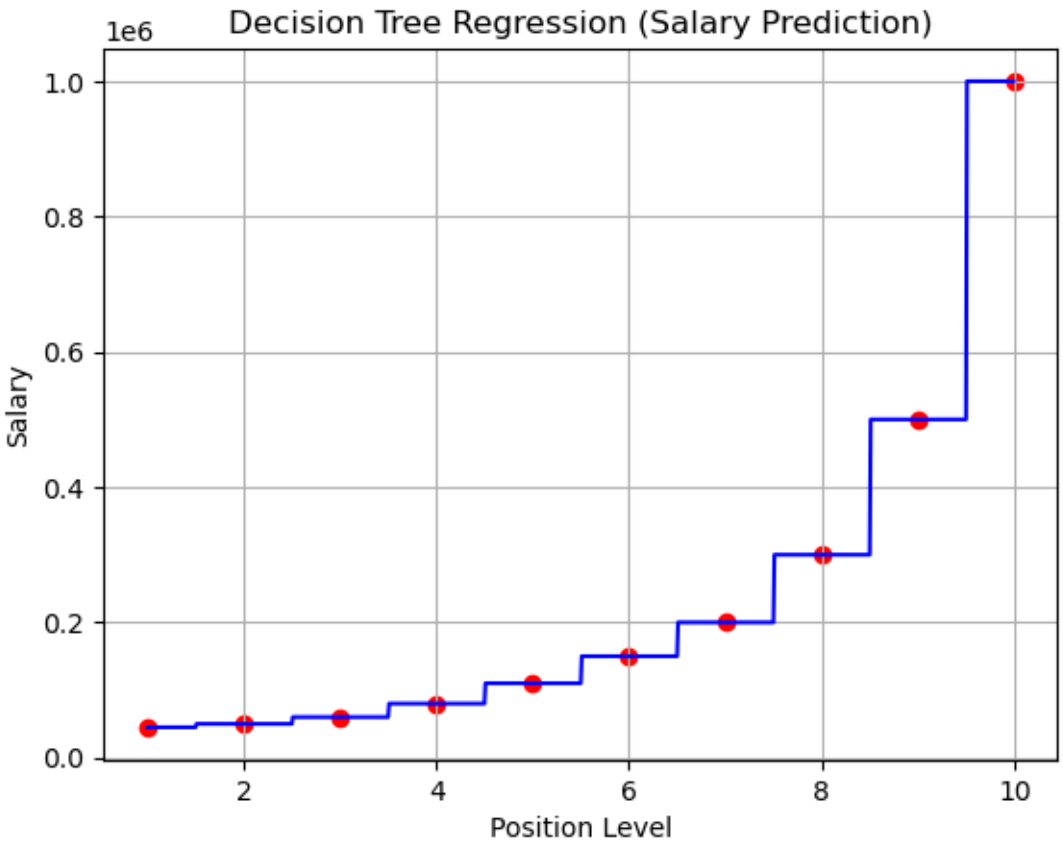
[6]:
```
array([500000.])
```

6 Visualization

[7]:
```
# Plotting Decision Tree Regressor result
X_grid = np.arange(min(X), max(X), 0.01)  # for smooth curve
X_grid = X_grid.reshape(len(X_grid), 1)

plt.scatter(X, y, color='red')
plt.plot(X_grid, dt_reg.predict(X_grid), color='blue')
plt.title("Decision Tree Regression (Salary Prediction)")
plt.xlabel("Position Level")
plt.ylabel("Salary")
plt.grid()
plt.show()
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_15404\10411790.py:2:
DeprecationWarning: Conversion of an array with ndim > 0 to a scalar is
deprecated, and will error in future. Ensure you extract a single element from
your array before performing this operation. (Deprecated NumPy 1.25.)
  X_grid = np.arange(min(X), max(X), 0.01)  # for smooth curve
```



7 Summary

- We understood what a **Decision Tree** is and how it can be used for regression.
- We implemented it using the `sklearn` library.
- We saw how the model splits data and makes predictions.
- Visualized the decision tree structure for better understanding.

Decision Trees are great for interpreting predictions and are widely used in real-world applications.