Contents

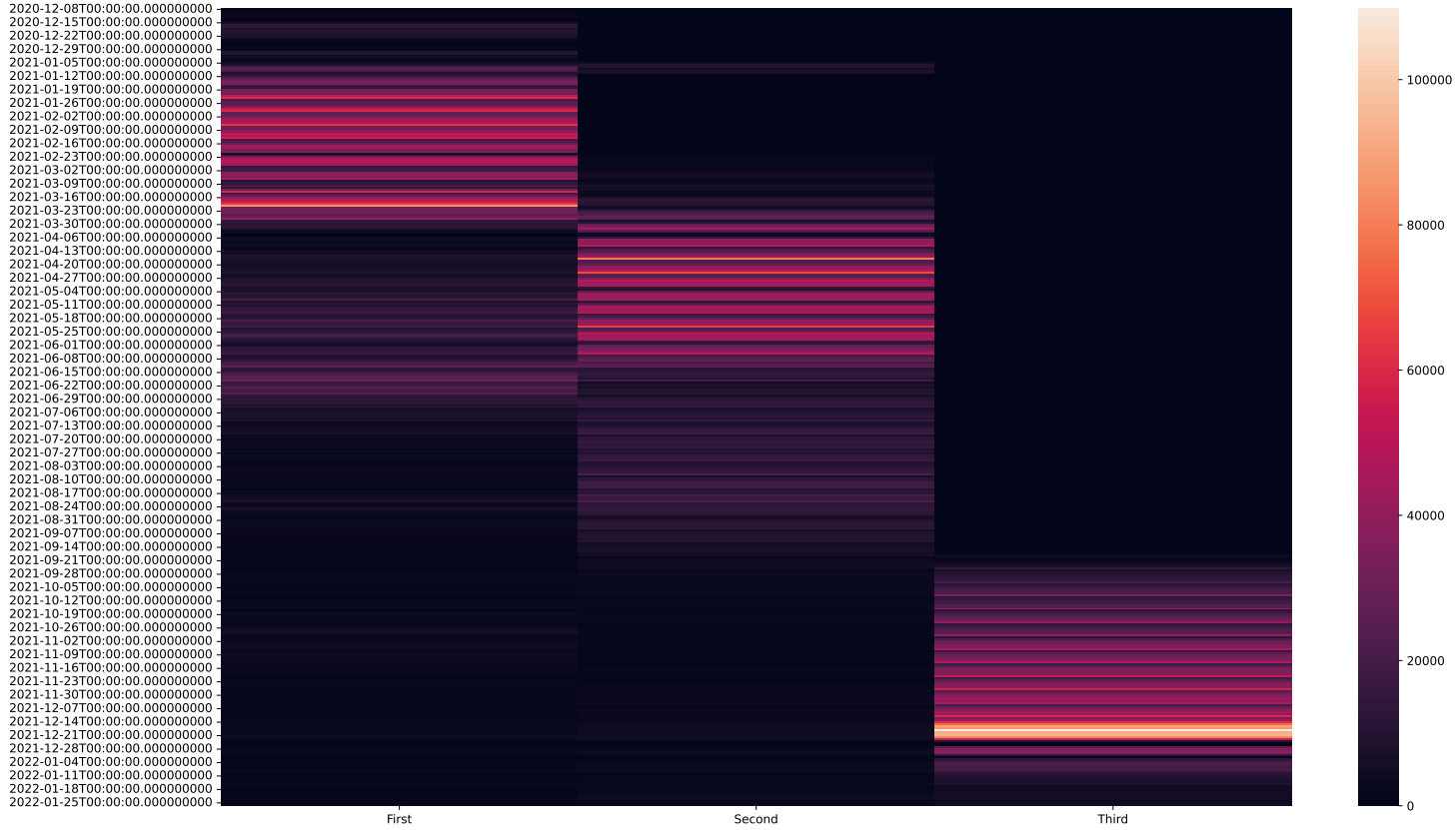
1 Step 4 Machine learning

1.1 Step 0: Look at and Modify the dataset

So, I am curious. Can I predict vaccination data?

I will work with the South West's vaccination data.

	First	Second	Third
2022-01-26	986	2520	4034
2022-01-25	899	1845	4283
2022-01-24	723	1445	3441
2022-01-23	1035	3007	3439
2022-01-22	1822	4709	5896
2022-01-21	1085	2362	4944
2022-01-20	1152	2330	5058
2022-01-19	1083	2524	5017
2022-01-18	1298	2126	5359
2022-01-17	946	1699	4374



As we can see, there are waves. So, the count of jabs depends on dates.

Let's get features: 1) Year 2) Month 3) Day etc.

	First	Second	Third	Year	Month	Day	DayOfYear	Weekday	Quarter	IsMonthStart	IsMonthEnd
2022-01-26	986	2520	4034	2022	1	26	26	2	1	FALSE	FALSE
2022-01-25	899	1845	4283	2022	1	25	25	1	1	FALSE	FALSE
2022-01-24	723	1445	3441	2022	1	24	24	0	1	FALSE	FALSE
2022-01-23	1035	3007	3439	2022	1	23	23	6	1	FALSE	FALSE
2022-01-22	1822	4709	5896	2022	1	22	22	5	1	FALSE	FALSE
2022-01-21	1085	2362	4944	2022	1	21	21	4	1	FALSE	FALSE
2022-01-20	1152	2330	5058	2022	1	20	20	3	1	FALSE	FALSE
2022-01-19	1083	2524	5017	2022	1	19	19	2	1	FALSE	FALSE
2022-01-18	1298	2126	5359	2022	1	18	18	1	1	FALSE	FALSE
2022-01-17	946	1699	4374	2022	1	17	17	0	1	FALSE	FALSE

First of all, I am going to use Regression Machine Learning models:

- Decision Tree
- Random Forest.

Let's look at the dataset carefully.

1.2 Step 1: Explore the dataset

1.2.1 Data types

	x
First	double
Second	double
Third	double
Year	double
Month	double
Day	double
DayOfYear	double
Weekday	double
Quarter	double
IsMonthStart	logical
IsMonthEnd	logical

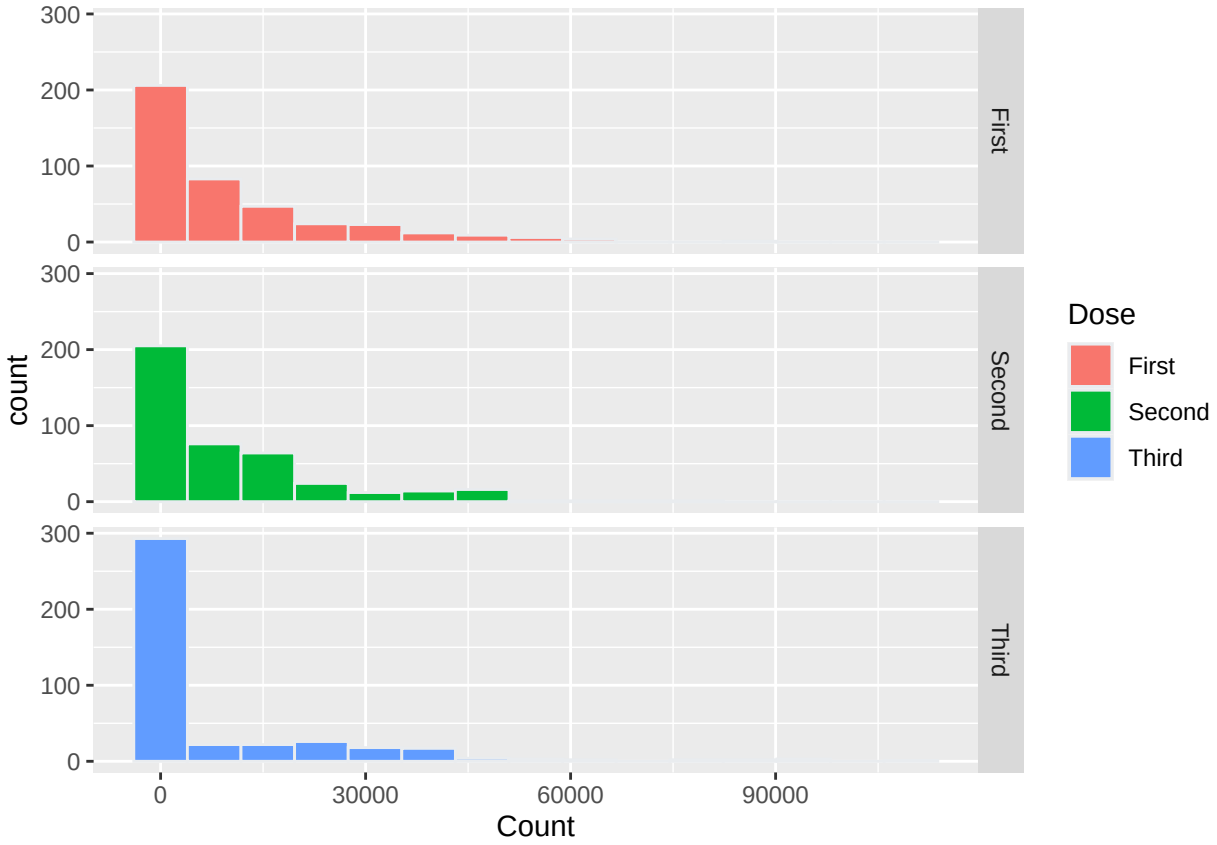
The good news is I don't need to convert my variables because they fit into Regression Machine Learning models.

1.2.2 Data description

Median and mean

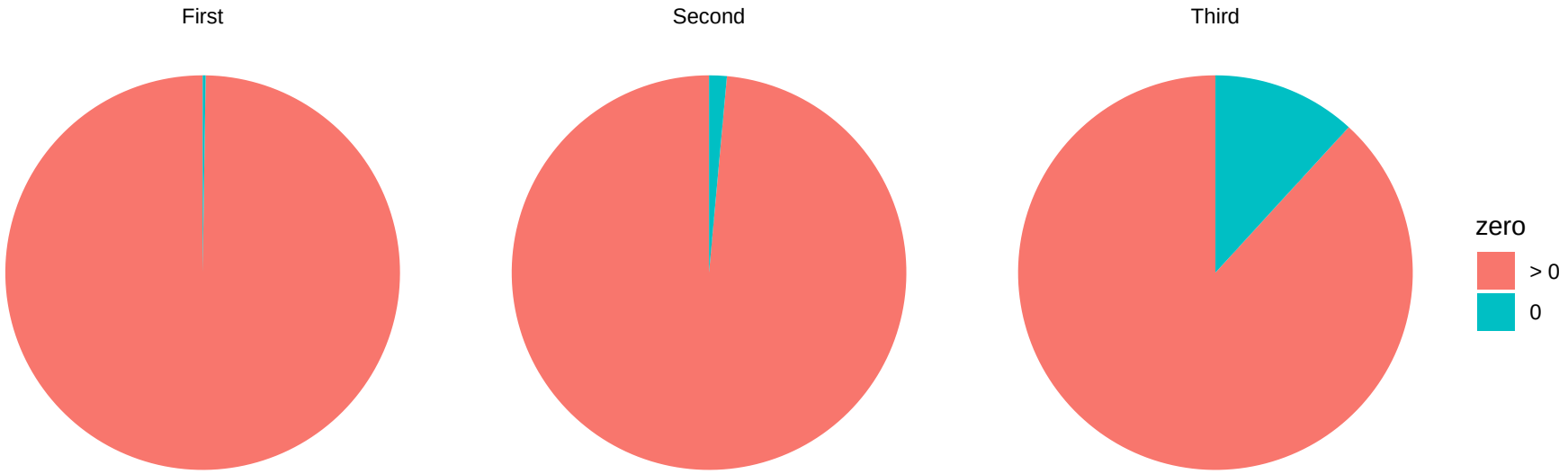
	mean	median
First	11037.61	4210
Second	10367.35	3998
Third	8507.13	6

Mean and median have a visible difference. What does it mean? There are large values that influence mean values.



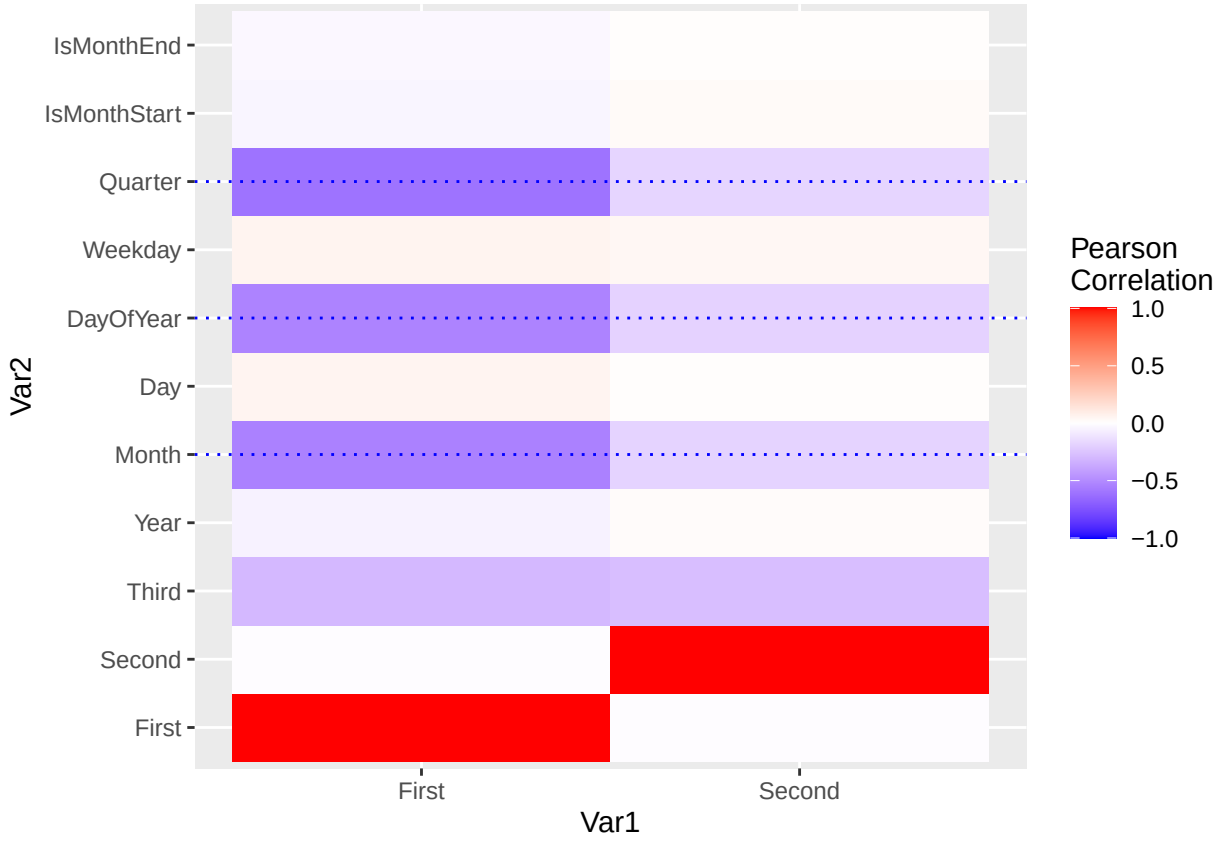
1.2.3 Zeroes

```
## `summarise()` has grouped output by 'Dose'. You can override using the `.groups` argument.
```



The column "Third" has more zero values than "First" and "Second; but, I think, it won't influence models' accuracy.

1.2.4 Correlations



Var1	Var2	value
First	Month	-0.5432969
Second	Month	-0.1888013
First	DayOfYear	-0.5343244
Second	DayOfYear	-0.1901012
First	Quarter	-0.6070344
Second	Quarter	-0.1799906

As we can see, the column “First” has strong relationships with

- “Quarter”,

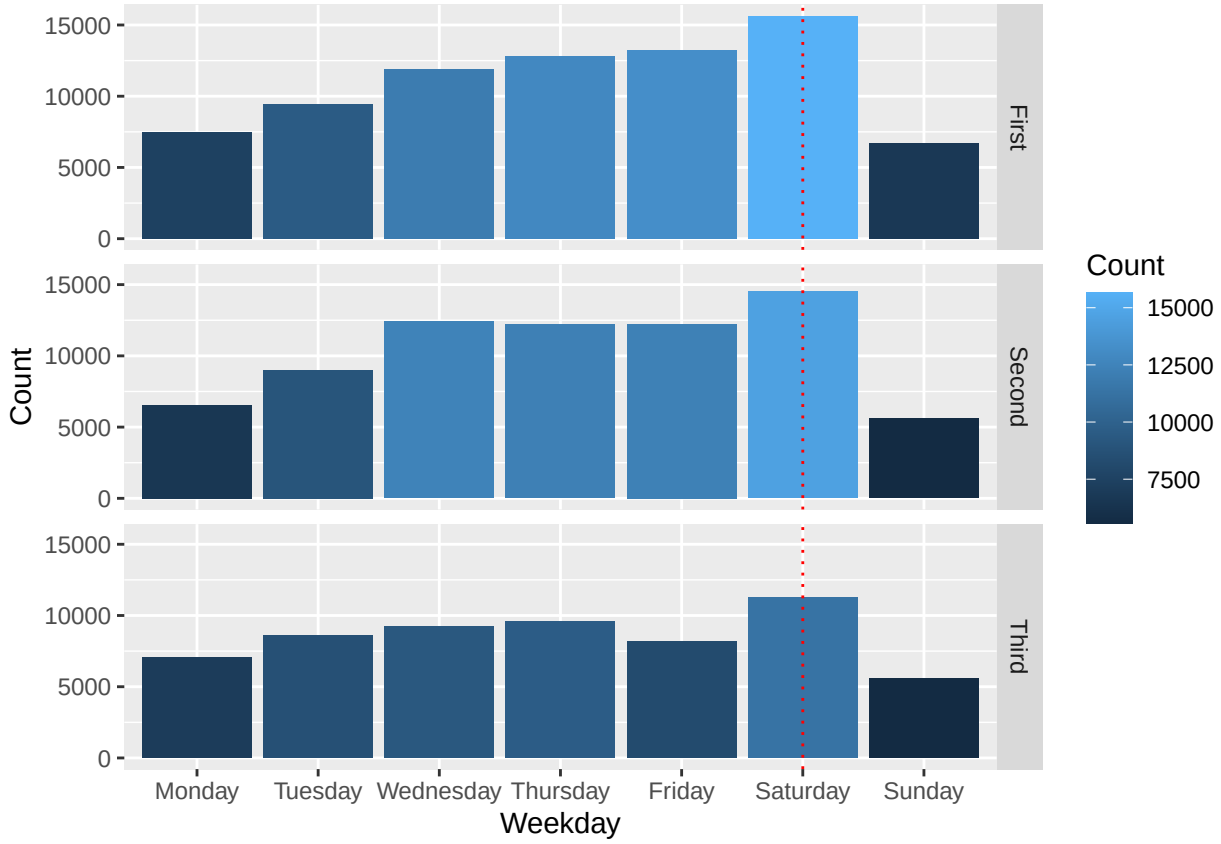
- “DayOfYear”,
- “Month”.

At the same time, the column “Second” doesn’t have strong relationships; but we can use the same columns.

1.2.5 Weekdays

As you remember, I have a question.

Let's answer.



So, most of South West's people prefer to get a jab on Saturdays.

1.2.6 Missing values

Calculate a count of dates in the dataset.

```
## 415
```

Calculate a count of dates between maximum and minimum dates.

```
## 415
```

There are no missing dates.

1.3 Step 2: Split sets, train a Machine Learning Model and Evaluate performance

Define necessary variables

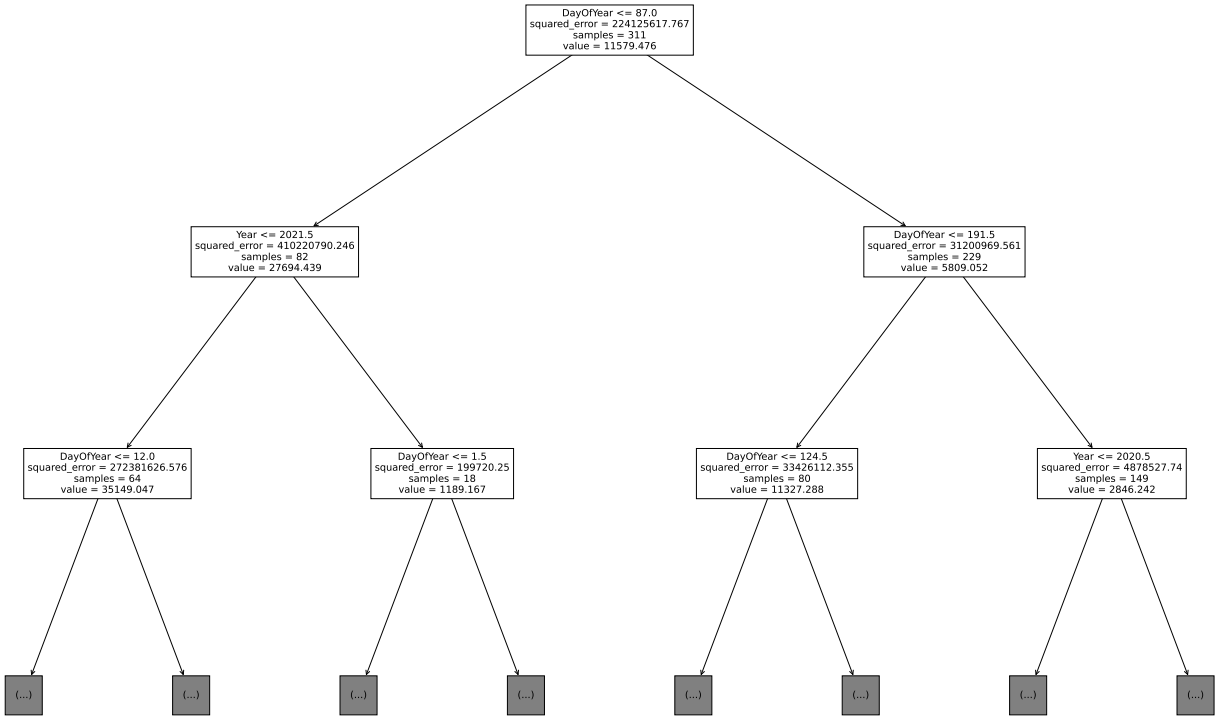
Prepare sets and train models using parameters.

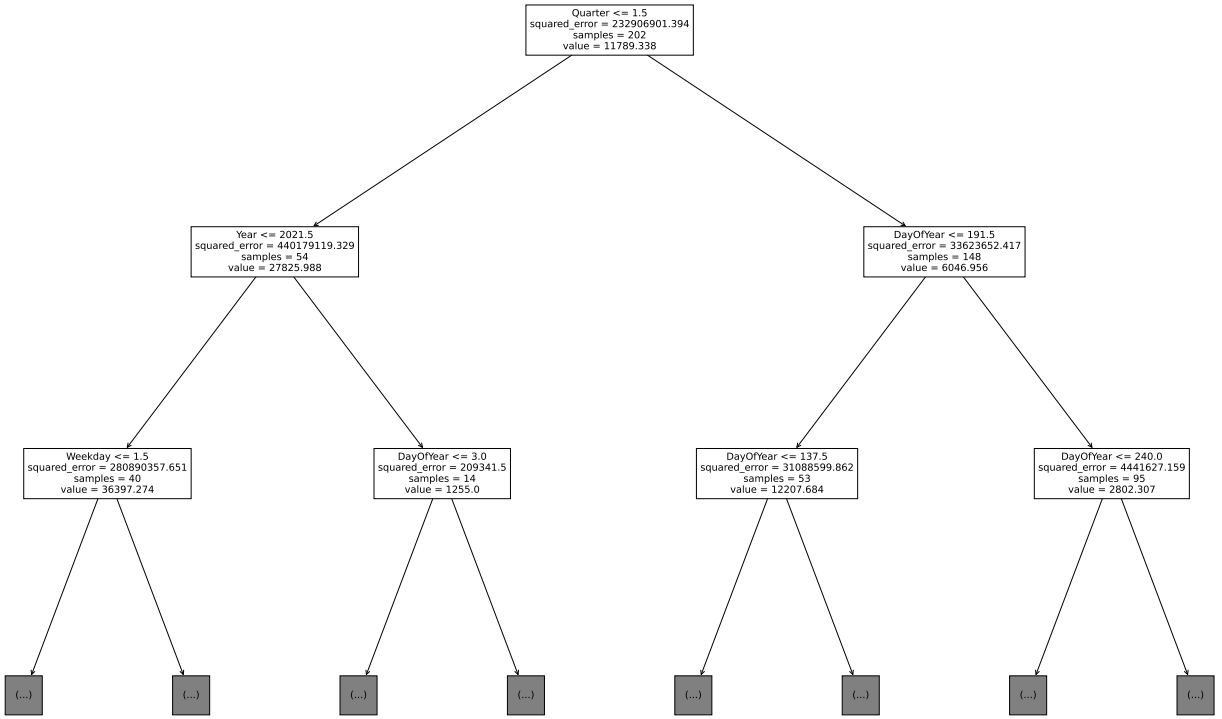
```
y_column = "First"
```

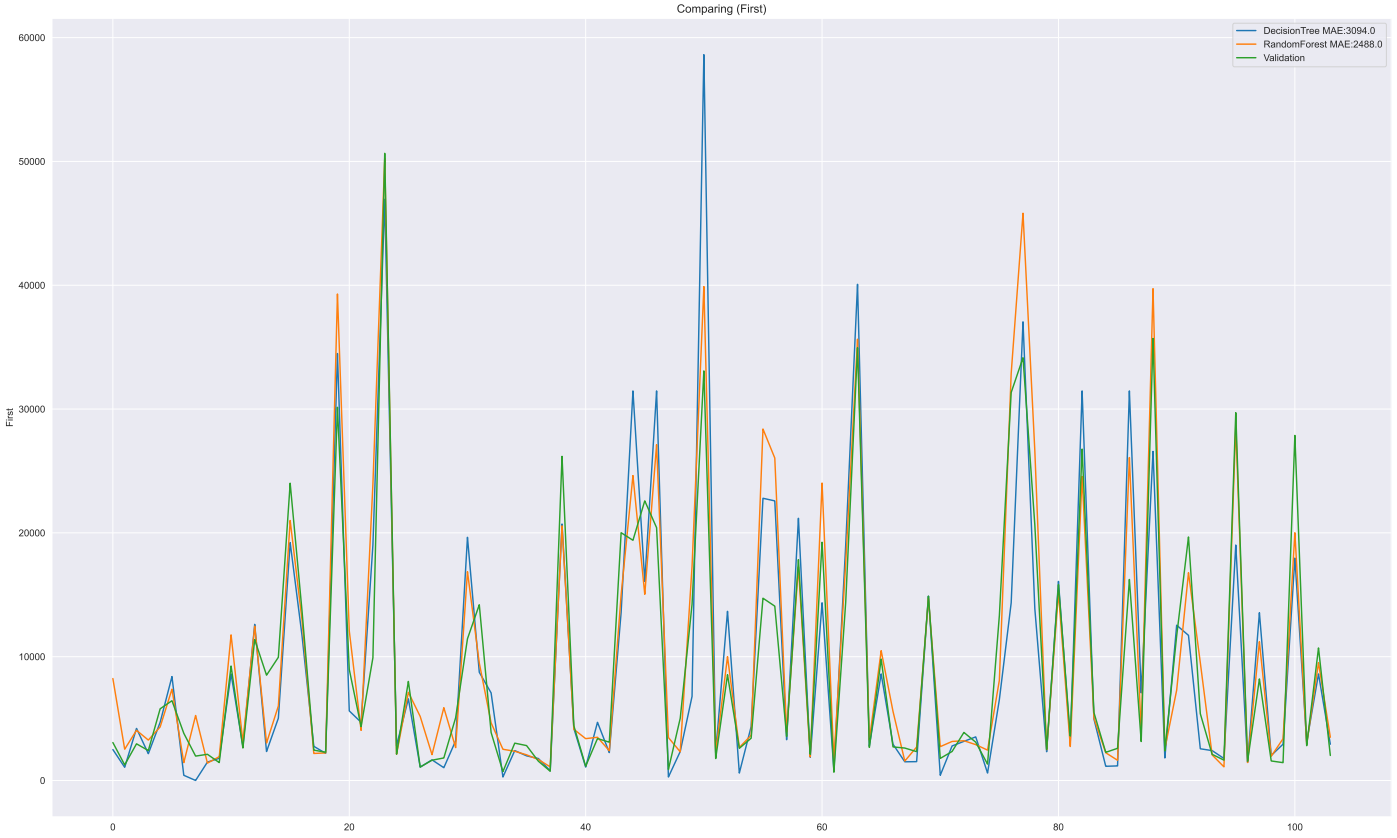
```
## DecisionTree:  0.719657929335243
```

```
## RandomForest:  0.774580856609961
```

Look at the tree

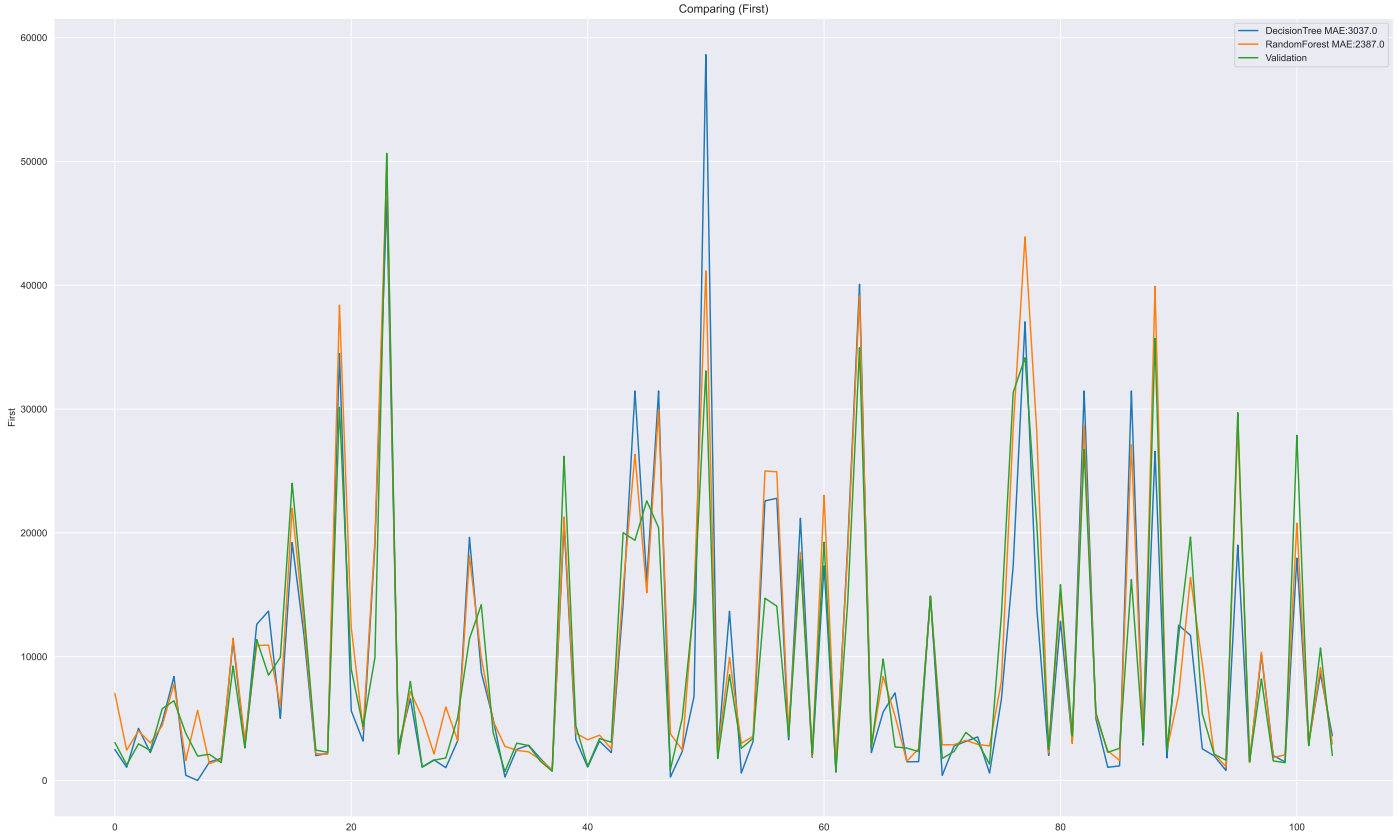

DayOfYear <= 87.0
squared_error = 224125617.767
samples = 311
value = 11579.476
Year <= 2021.5
squared_error = 410220790.246
samples = 82
value = 27694.439
DayOfYear <= 191.5
squared_error = 31200969.561
samples = 229
value = 5809.052
DayOfYear <= 12.0
squared_error = 272381626.576
samples = 64
value = 35149.047
DayOfYear <= 1.5
squared_error = 199720.25
samples = 18
value = 1189.167
DayOfYear <= 124.5
squared_error = 33426112.355
samples = 80
value = 11327.288
Year <= 2020.5
squared_error = 4878527.74
samples = 149
value = 2846.242
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)


Quarter <= 1.5
squared_error = 232906901.394
samples = 202
value = 11789.338
Year <= 2021.5
squared_error = 4401791 19.329
samples = 54
value = 27825.988
DayOfYear <= 191.5
squared_error = 33623652.417
samples = 148
value = 6046.956
Weekday <= 1.5
squared_error = 280890357.651
samples = 40
value = 36397.274
DayOfYear <= 3.0
squared_error = 209341.5
samples = 14
value = 1255.0
DayOfYear <= 137.5
squared_error = 31088599.862
samples = 53
value = 12207.684
DayOfYear <= 240.0
squared_error = 4441627.159
samples = 95
value = 2802.307
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)




```
## DecisionTree:  0.7248630326024768
```

```
## RandomForest:  0.7837038702898657
```

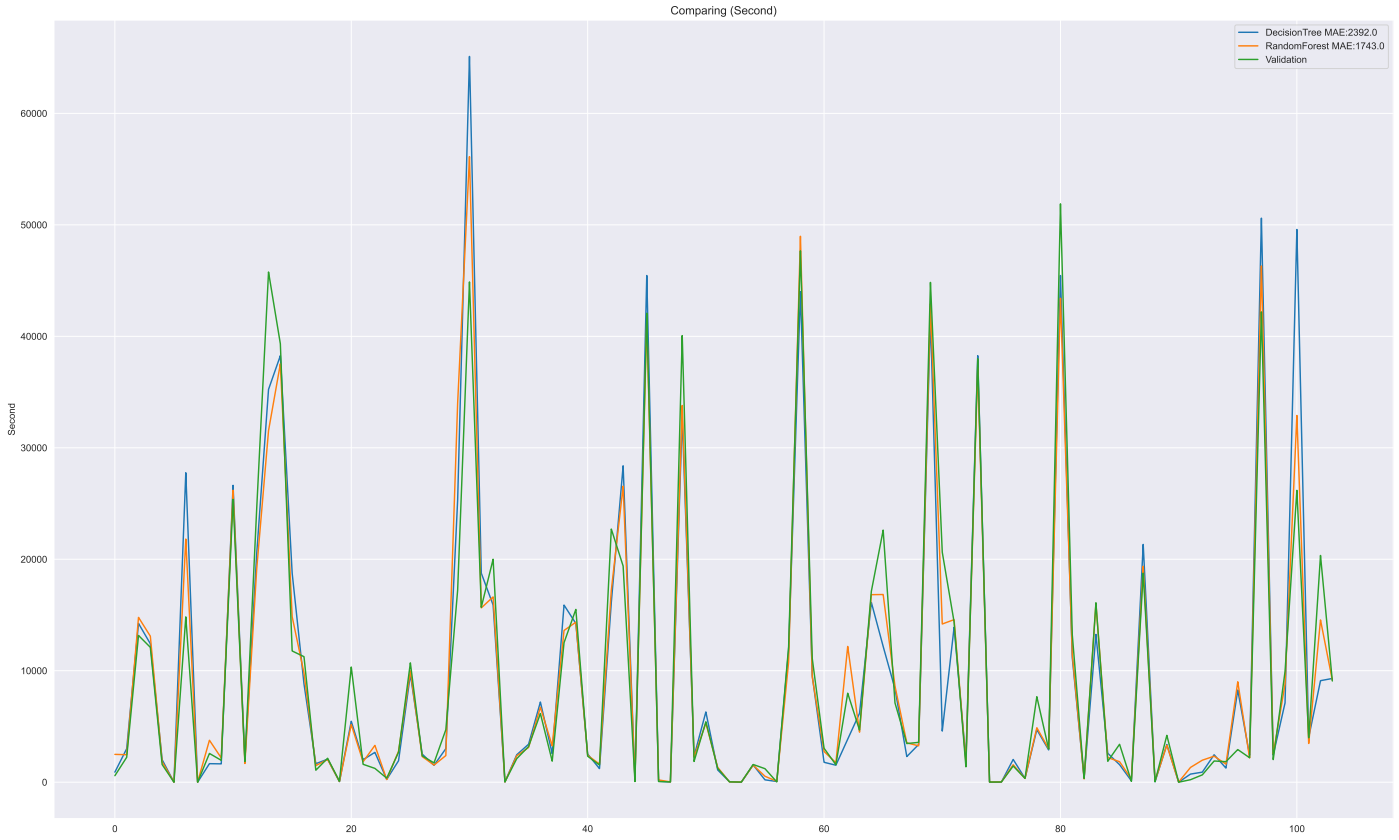


Repeat for the Second

```
y_column = "Second"
```

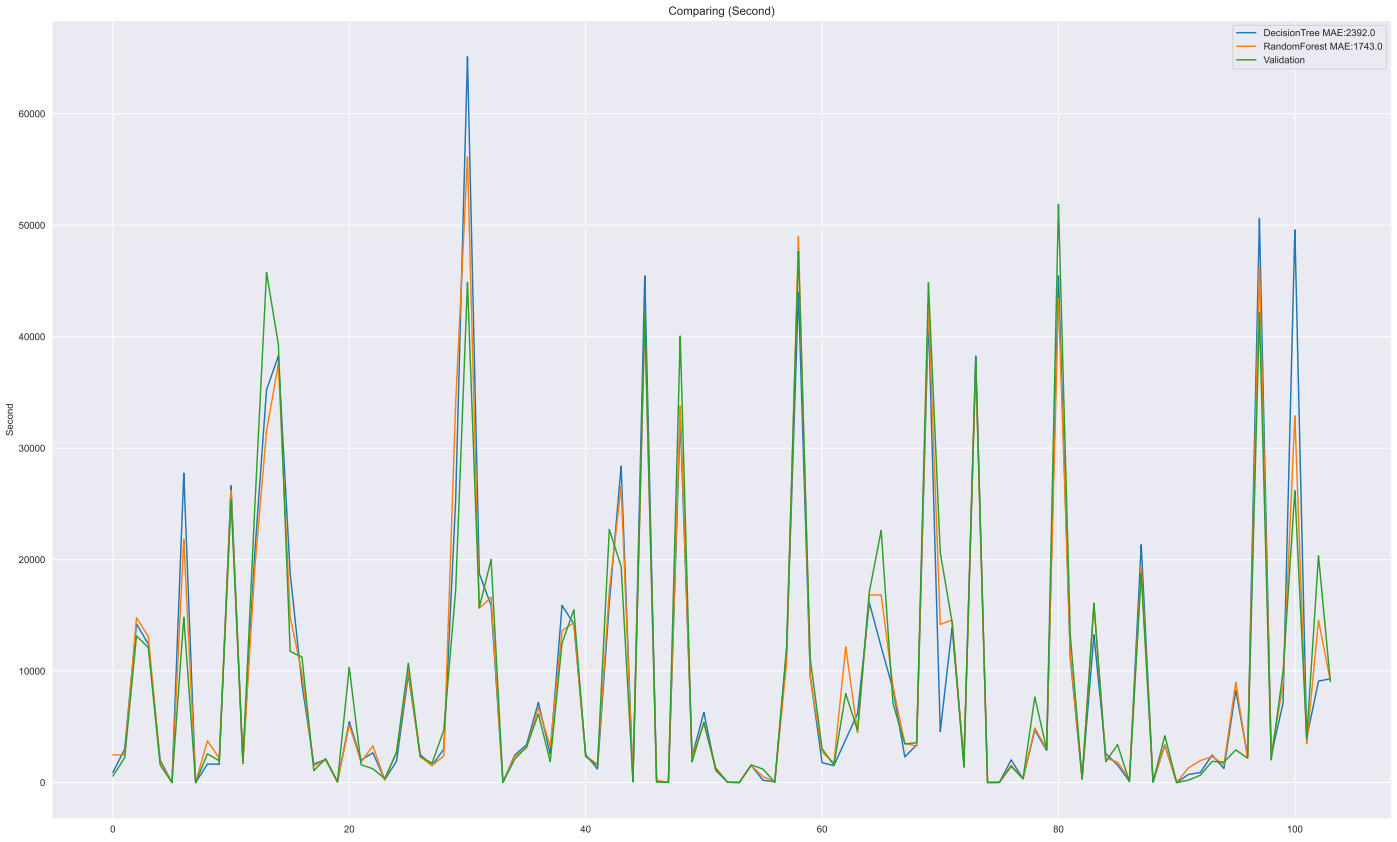
```
## DecisionTree:  0.7692636418171874
```

```
## RandomForest:  0.8318561653086721
```



```
## DecisionTree:  0.7692636418171874

## RandomForest:  0.8318561653086721
```

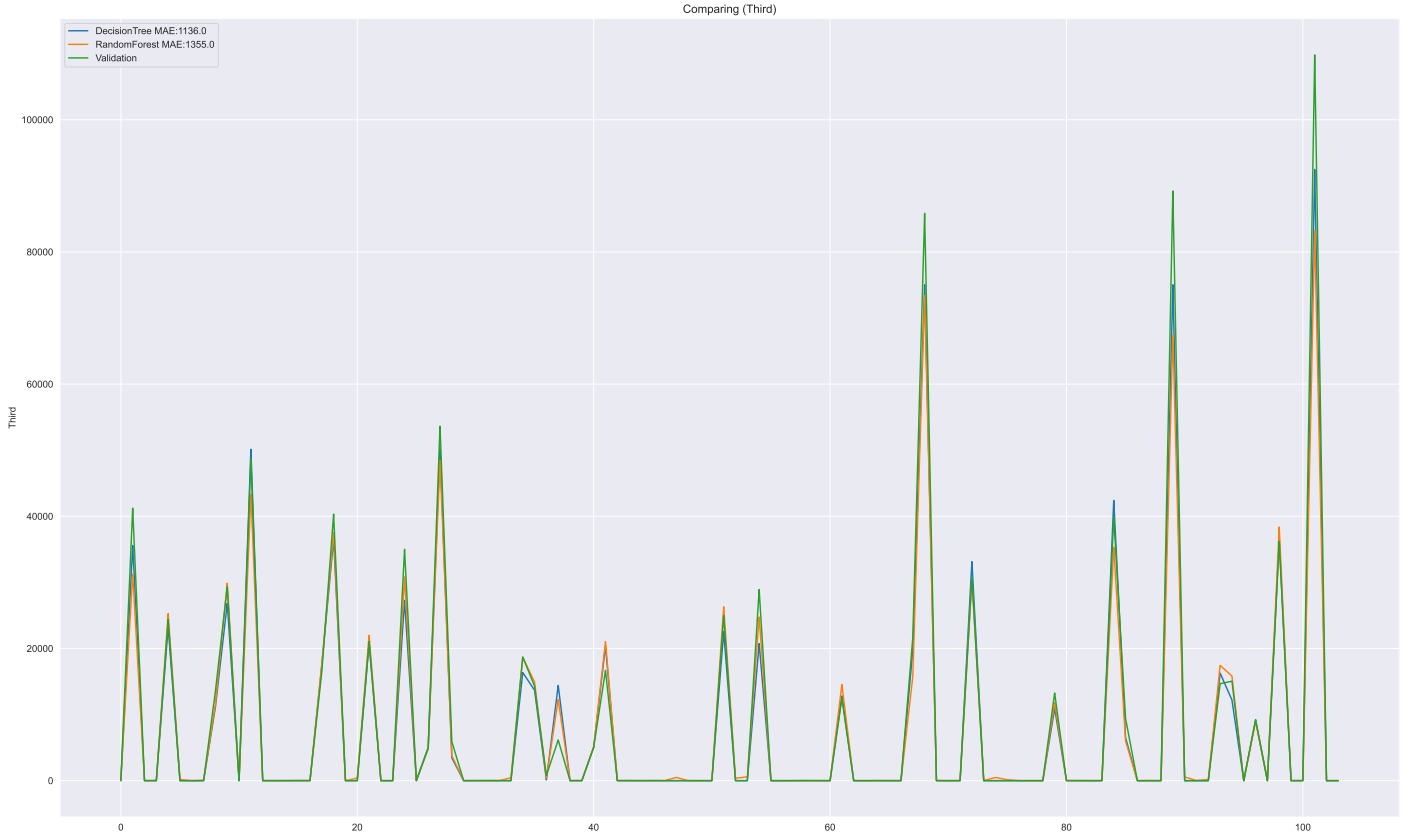


Repeat for Third

```
y_column = "Third"
```

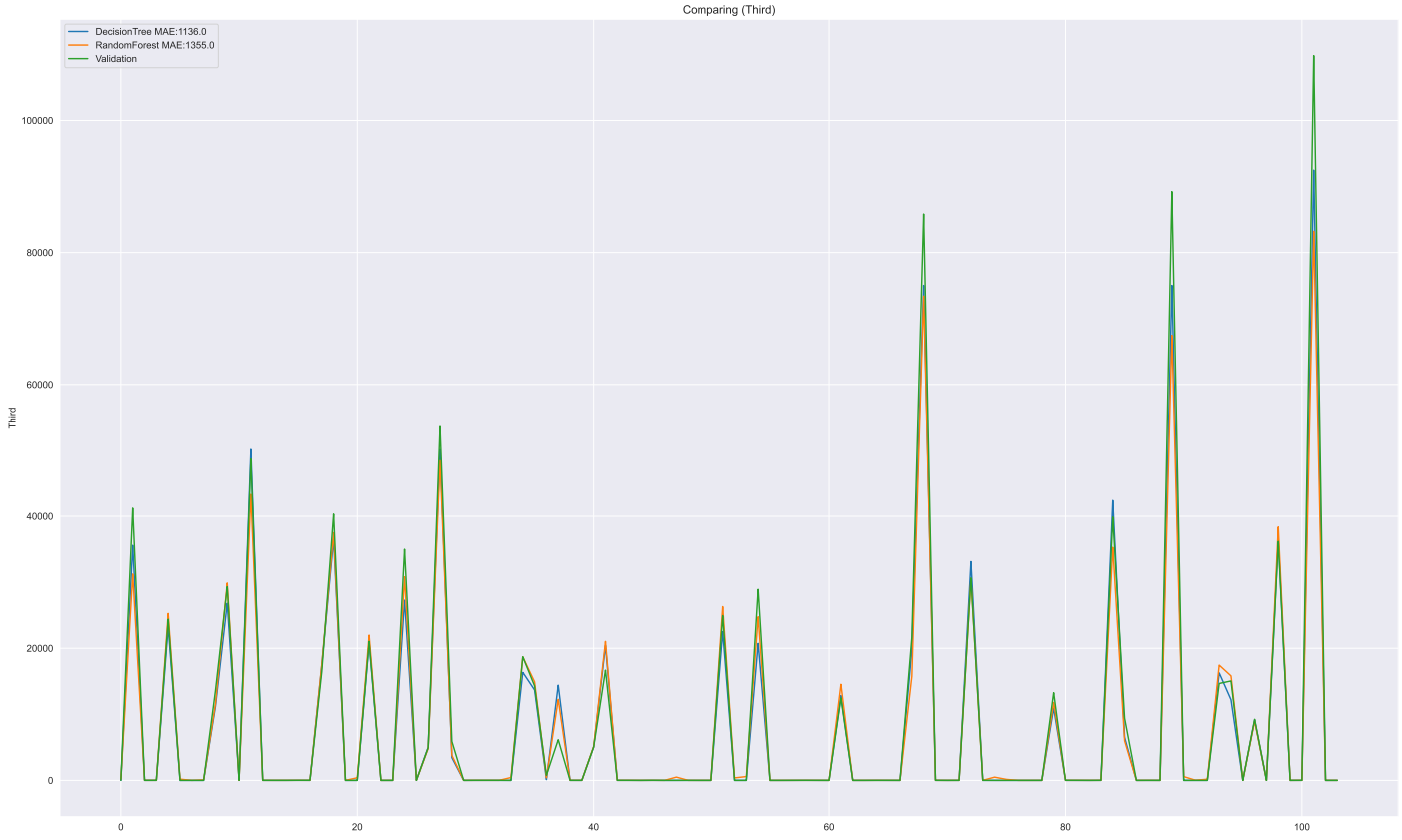
```
## DecisionTree:  0.8664423499345815
```

```
## RandomForest:  0.840745098066024
```



```
## DecisionTree:  0.8664423499345815

## RandomForest:  0.840745098066024
```



Compare the score with the mean value of the column that we predicted.

A combination of the following features give us the best result: Weekday, Year, DayOfYear.