# President Trump’s

“Executive Time”

# Problem Formulation

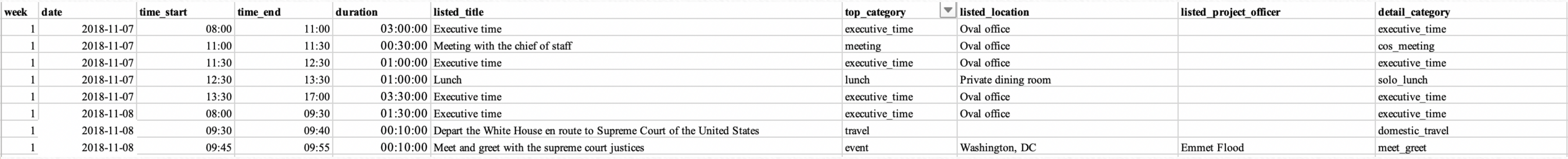
Each day at 14:30, we have the possibility of buying an ad on the president’s favourite TV channel, and that ad will be played the same day at 15:01. But buying an ad is very expensive, and we want to maximize our chances that he will be watching TV at that time on that day, which is only possible during his "executive time". Each day before making our decision, we have access to the top_category of what president Trump was doing at 9:01, 11:01, and 13:01 on that specific day. Can you help us better invest our money by predicting the top_category of what he does at 15:01 on that day?

- Input
  - top_category of what president Trump was doing at 9:01, 11:01, and 13:01 on that specific day

- Output(prediction)
  - top_category of what he does at 15:01 on that day

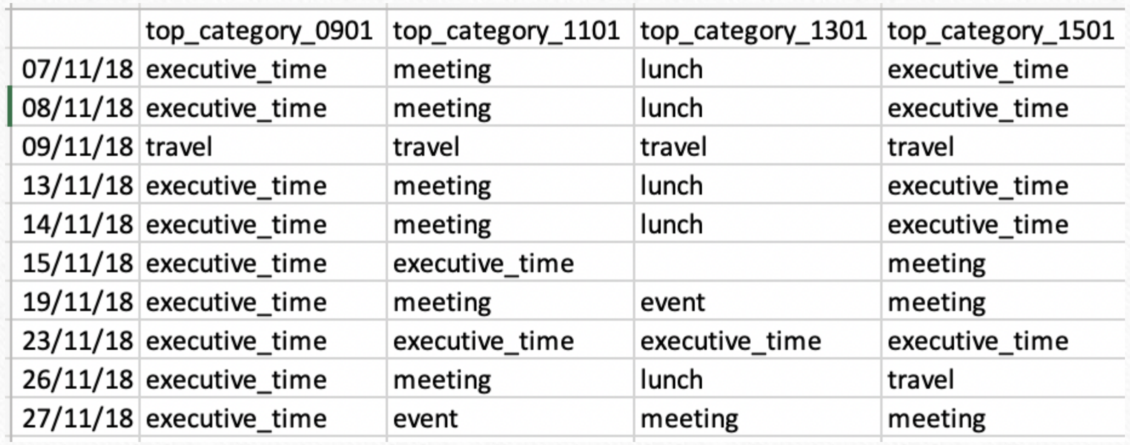
# Data Engineering Skills

- Raw Dataset



| week | date | time_start | time_end | duration | listed_title | top_category | listed_location | listed_project_officer | detail_category |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2018-11-07 | 08:00 | 11:00 | 03:00:00 | Executive time | executive_time | Oval office | | executive_time |
| 1 | 2018-11-07 | 11:00 | 11:30 | 00:30:00 | Meeting with the chief of staff | meeting | Oval office | | cos_meeting |
| 1 | 2018-11-07 | 11:30 | 12:30 | 01:00:00 | Executive time | executive_time | Oval office | | executive_time |
| 1 | 2018-11-07 | 12:30 | 13:30 | 01:00:00 | Lunch | lunch | Private dining room | | solo_lunch |
| 1 | 2018-11-07 | 13:30 | 17:00 | 03:30:00 | Executive time | executive_time | Oval office | | executive_time |
| 1 | 2018-11-08 | 08:00 | 09:30 | 01:30:00 | Executive time | executive_time | Oval office | | executive_time |
| 1 | 2018-11-08 | 09:30 | 09:40 | 00:10:00 | Depart the White House en route to Supreme Court of the United States | travel | | | domestic_travel |
| 1 | 2018-11-08 | 09:45 | 09:55 | 00:10:00 | Meet and greet with the supreme court justices | event | Washington, DC | Emmet Flood | meet_greet |

- Preprocessed Dataset



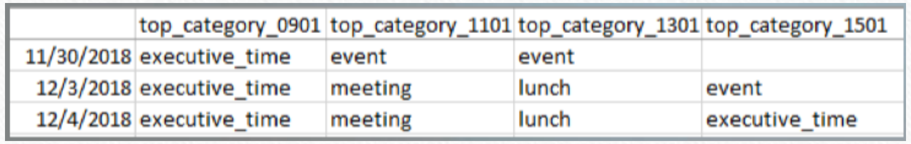
| | top_category_0901 | top_category_1101 | top_category_1301 | top_category_1501 |
|---|---|---|---|---|
| 07/11/18 | executive_time | meeting | lunch | executive_time |
| 08/11/18 | executive_time | meeting | lunch | executive_time |
| 09/11/18 | travel | travel | travel | travel |
| 13/11/18 | executive_time | meeting | lunch | executive_time |
| 14/11/18 | executive_time | meeting | lunch | executive_time |
| 15/11/18 | executive_time | executive_time | | meeting |
| 19/11/18 | executive_time | meeting | event | meeting |
| 23/11/18 | executive_time | executive_time | executive_time | executive_time |
| 26/11/18 | executive_time | meeting | lunch | travel |
| 27/11/18 | executive_time | event | meeting | meeting |

# Hypotheses

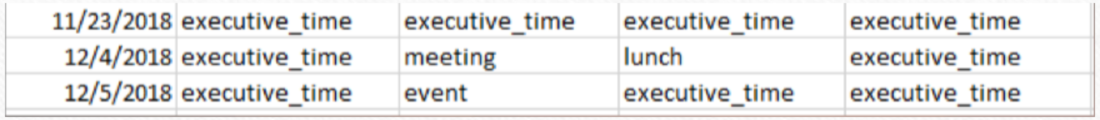
1. The output is same if the input is same in each top_category column and vice versa.
2. Data does not have equal distribution of number of classes so it becomes a class imbalance problem.

# Result of Validation

1. After doing the prediction , I found that our hypotheses is wrong as per the example below.

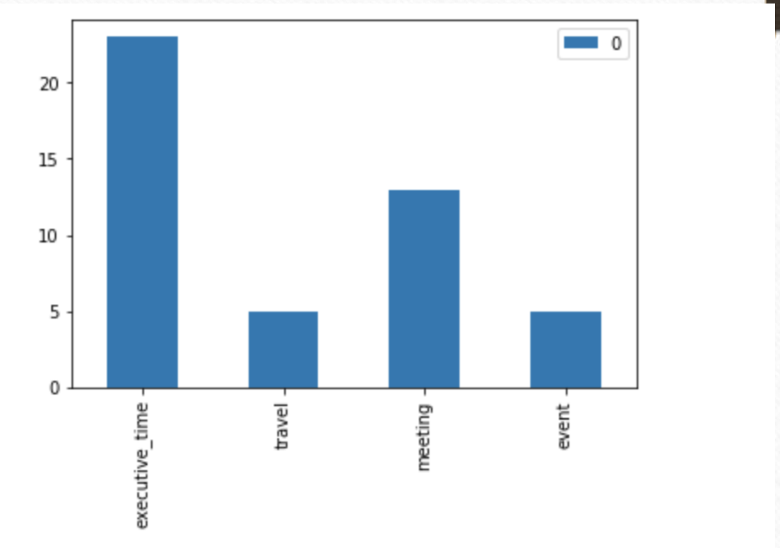


| | top_category_0901 | top_category_1101 | top_category_1301 | top_category_1501 |
|---|---|---|---|---|
| 11/30/2018 | executive_time | event | event | |
| 12/3/2018 | executive_time | meeting | lunch | event |
| 12/4/2018 | executive_time | meeting | lunch | executive_time |



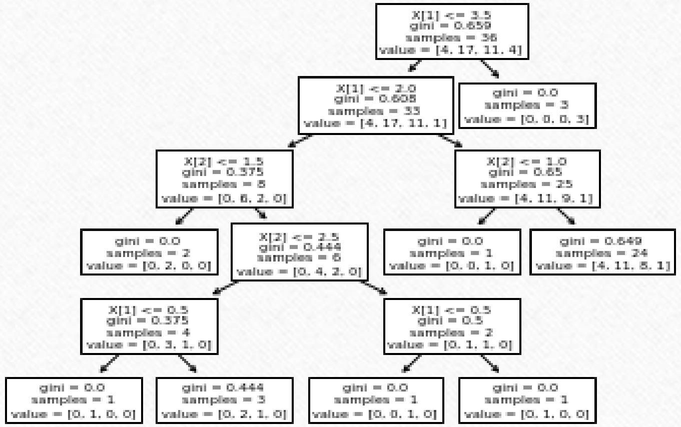
| | | | | |
|---|---|---|---|---|
| 11/23/2018 | executive_time | executive_time | executive_time | executive_time |
| 12/4/2018 | executive_time | meeting | lunch | executive_time |
| 12/5/2018 | executive_time | event | executive_time | executive_time |

2. Our hypotheses is proved right by the bar graph on the side as there are only few data points in travel and event top_category confirming it is a class imbalance problem.



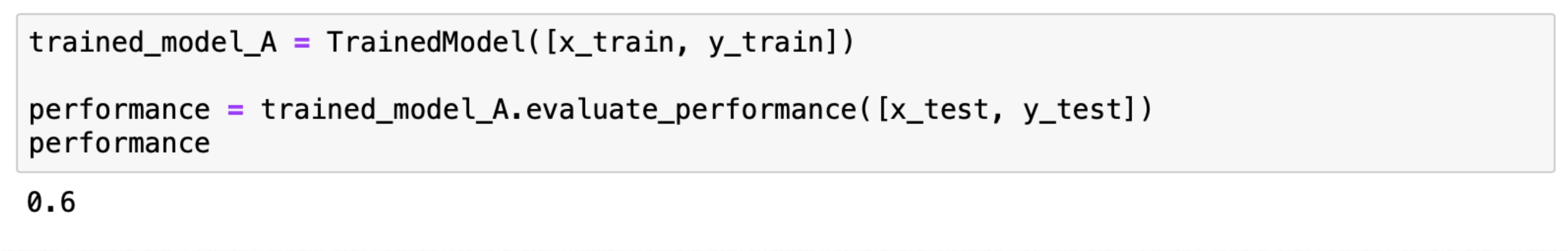
# Machine Learning Model : Decision Trees

- We have been given multiple classes in the form of top_category and we must predict the top_category at 15:01 based on previous activities. The machine learning model, I have used for this prediction is decision trees which is good for multi-class classification problem and also, the data set is small with less number of classes. The visualization of the Decision Tress looks as below :



- Our model produces 60% F1 Score on the test data as shown in next slide. The performance metric used is F1 Score because it is a class imbalance problem. As accuracy is not a reliable performance metric incase of class imbalance problem, so I chose F1-score as metric which considers both recall and precision and to produce unbiased result.

# Performance Metric : F1 Score

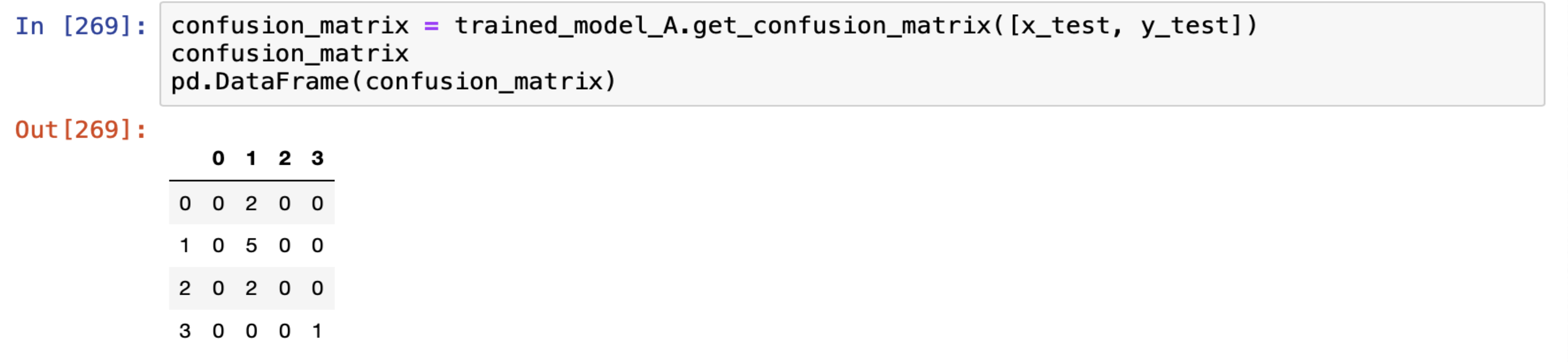


```
trained_model_A = TrainedModel([x_train, y_train])

performance = trained_model_A.evaluate_performance([x_test, y_test])
performance
```

```
0.6
```

# Confusion Matrix



```
In [269]: confusion_matrix = trained_model_A.get_confusion_matrix([x_test, y_test])
          confusion_matrix
          pd.DataFrame(confusion_matrix)
```

Out[269]:

|   | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0 | 2 | 0 | 0 |
| 1 | 0 | 5 | 0 | 0 |
| 2 | 0 | 2 | 0 | 0 |
| 3 | 0 | 0 | 0 | 1 |

THANK YOU ☺