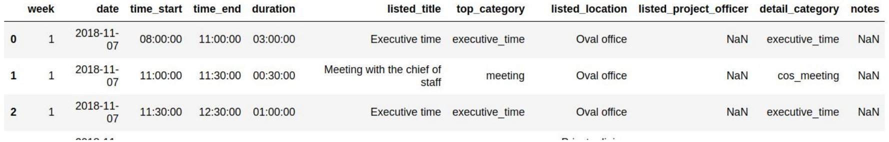
President Trump's "Executive Time"

Business problem

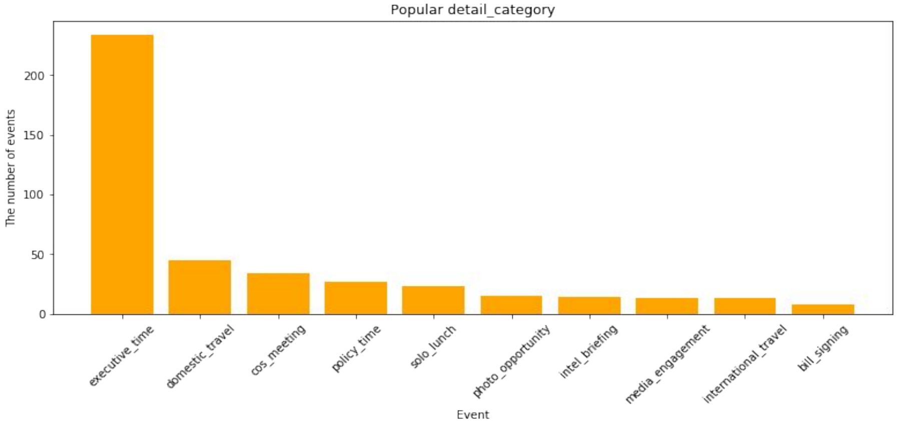
The business problem is to buy TV add at 15:01 if President Trump is going to watch TV at this time. To optimize the investments we only need to know is it executive_time or not. Trump will be watching TV only during his "executive time".

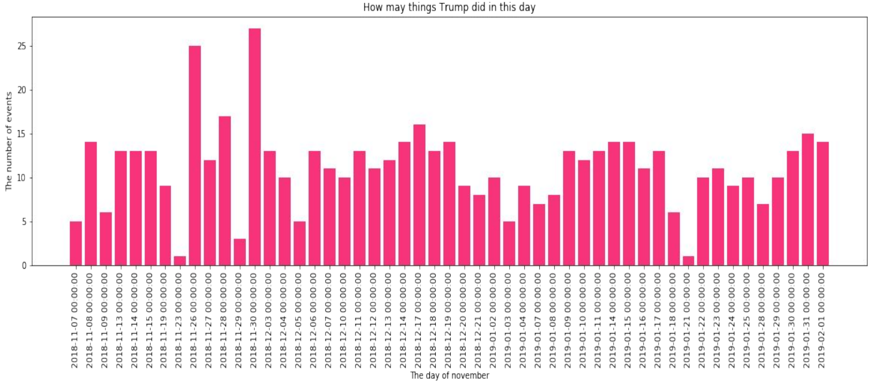
The goal is predict the `top_category` the of what he does at 15:01 on that day. But from business problem we are more interested to get good performance for class executive_time prediction only.

Raw dataset:

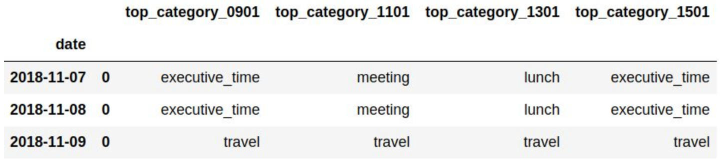


	week	date	time_start	time_end	duration	listed_title	top_category	listed_location	listed_project_officer	detail_category	notes
0	1	2018-11-07	08:00:00	11:00:00	03:00:00	Executive time	executive_time	Oval office	NaN	executive_time	NaN
1	1	2018-11-07	11:00:00	11:30:00	00:30:00	Meeting with the chief of staff	meeting	Oval office	NaN	cos_meeting	NaN
2	1	2018-11-07	11:30:00	12:30:00	01:00:00	Executive time	executive_time	Oval office	NaN	executive_time	NaN

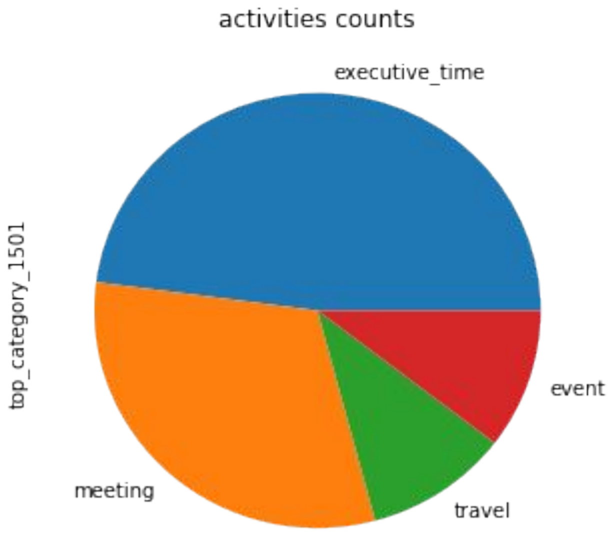




Preprocessed dataset



		top_category_0901	top_category_1101	top_category_1301	top_category_1501
date					
2018-11-07	**0**	executive_time	meeting	lunch	executive_time
2018-11-08	**0**	executive_time	meeting	lunch	executive_time
2018-11-09	**0**	travel	travel	travel	travel



executive_time	23
meeting	15
travel	5
event	5

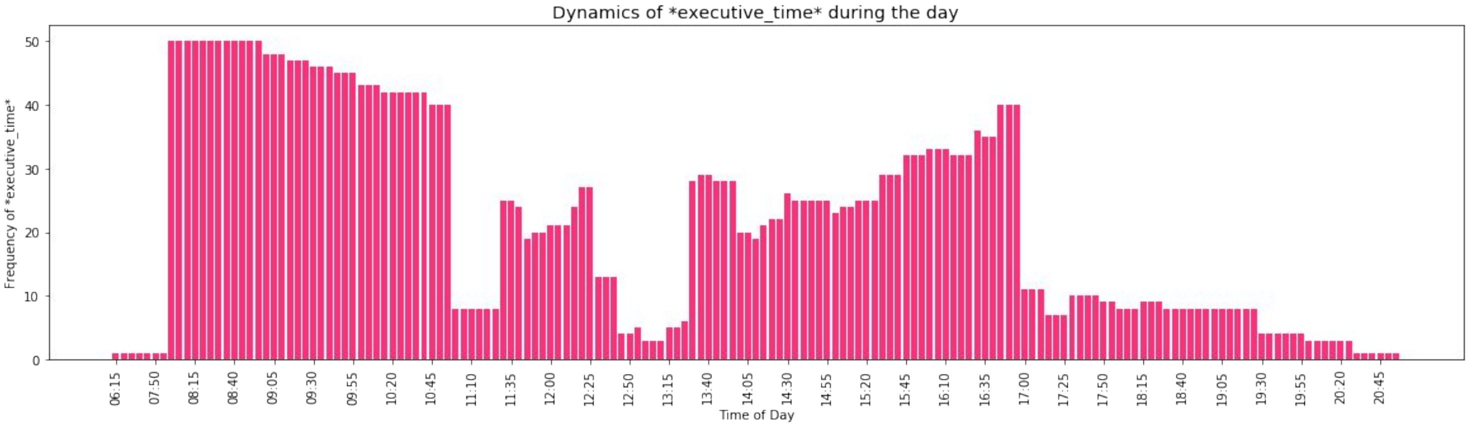
Hypotheses #1

Raw dataset is representative or not: if in each new month Trump does a new activity, we can’t predict the top_category, because it is new and our models doesn’t know anything about this new category.

I separated dataset by two parts without shuffle in time and the result is: President Trump has no new activities in the last month

Hypotheses #2

The most popular event is executive_time. And just in this period Trump can watch the TV. In business problem is predict the top_category at 15:01, and we will buy an ad if that top_category is executive_time.
'executive_time' at 15:01 happens in 9.87124463519 % of all executive time periods



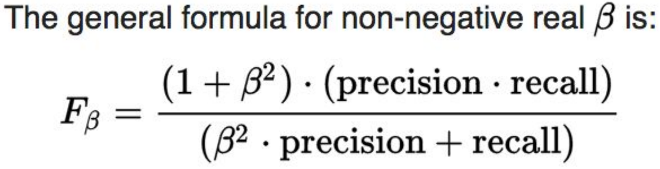
From the dynamics, we can suggest buying an ad not at 15:01, but at time between 16:35 and 17:00, or in the morning, because Thump does executive_time more often than at 15:01

The model and performance metric

The chosen model is RandomForestClassifier. We have a classification problem and small dataset. More complicated model needs more data to train on, while Random Forest is flexible and generalizable.

Performance metric. We predict class for top_category_1501, but we don’t care if we make a mistake between ‘travel’ and ‘meeting’, we really care if we make any mistake in class ‘executive_time’. From business problem, it is better to miss 1 chance to show an ad than to buy it wrong (as we know the ads are expensive). So we use weights for precision and recall.

Model does multi-class prediction, but performance is measured on binary prediction (1 = if executive_time, 0 = other class. And chosen metric is F_beta_score. The value of parameter beta should be fixed w.r.t. ad prices and other considerations:



The general formula for non-negative real β is:

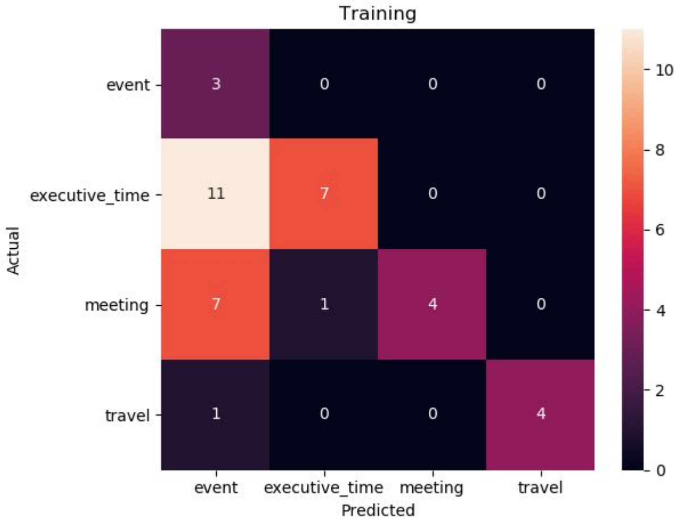
$$F_\beta = \frac{(1+\beta^2)\cdot(\text{precision}\cdot\text{recall})}{(\beta^2\cdot\text{precision}+\text{recall})}$$

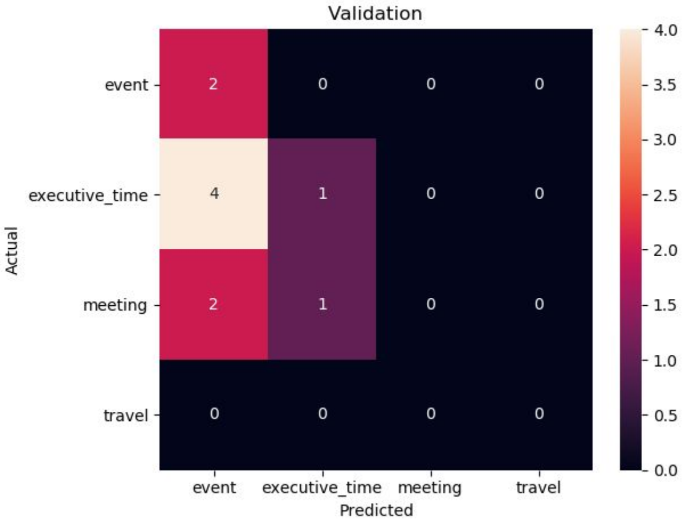
Performance:

Beta = 0.2

Training performance (dataset_A) = 0.677

Validation performance (dataset_B) = 0.473


Training
Actual
Predicted
event
executive_time
meeting
travel
event
executive_time
meeting
travel


Validation
Actual
Predicted
event
executive_time
meeting
travel
event
executive_time
meeting
travel