Notes on N26 Dataset

(More detailed step-by-step comments inline within script - “N26.py”)

0. Instruction on running script:

Language version used: Python 2.7.9

Libraries used: pandas, numpy, sklearn, matplotlib

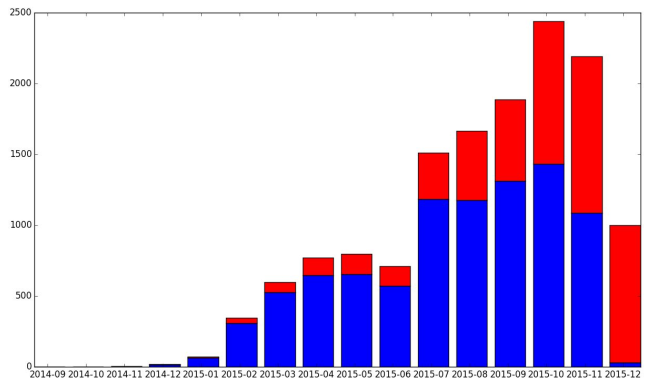
Place original dataset, new test.csv, and rf_model.pkl file in working directory, then:

a) For new dataset: replace csv_path with your test file, set use_trained_model = True
b) To review results for existing N26_TRAIN.csv dataset: run script in current state

1. Comments on grouping

No geographical mapping of user nationalities attempted given time constraints - for a real use-case, one could use methods from geopy and basemap to see global distribution of users. (This would be more useful if we had current residence of users, not just birthplace, of course).

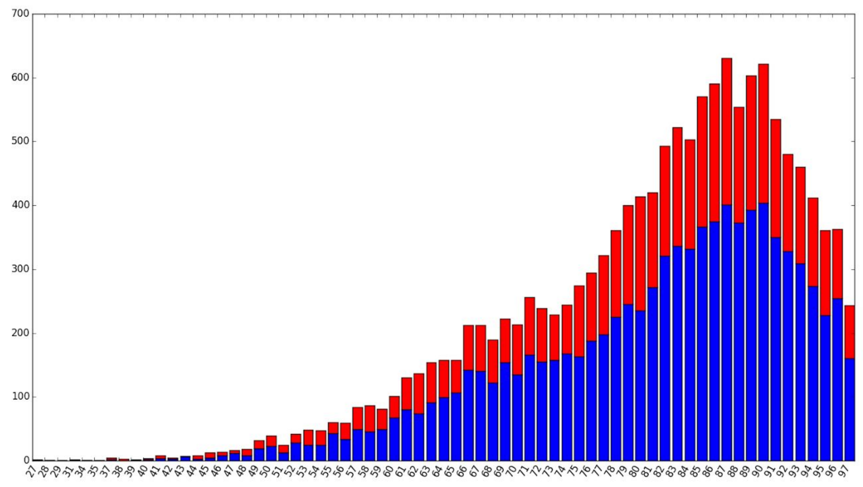
I was mainly interested in visual distribution of main variables within data (in each case showing proportion of incomplete signups for each group in red).



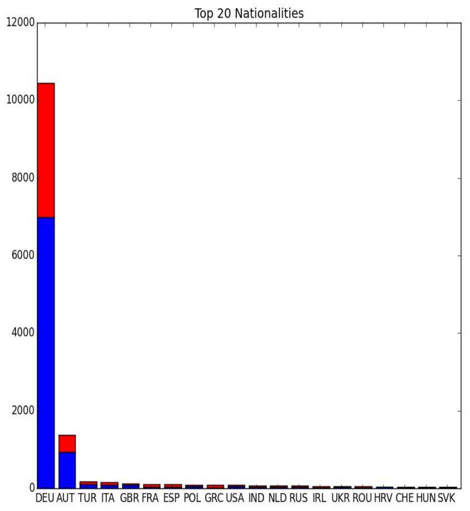
Month joined / user distribution:

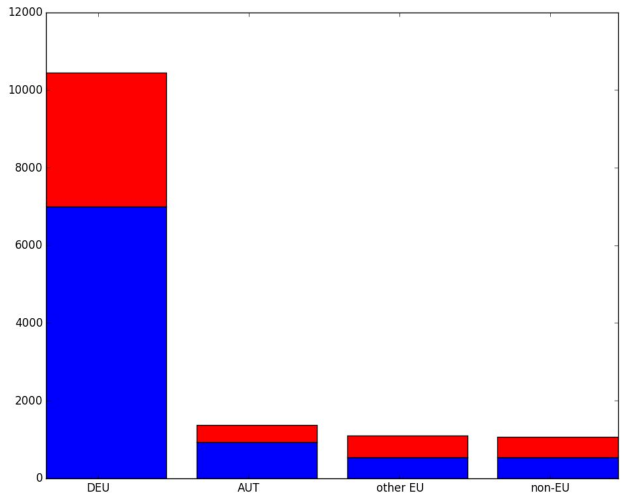
Note the high proportion of incomplete signups for December 2015. This can be explained by early cut-off of dataset (I am guessing dataset was created middle of December as only users up to 7th of December are represented)

Strong growth trend visible, with higher proportion of successful signups among first groups of users (which makes sense given that beta-testers were probably given more attention, and also because early users will have had more time to complete sign-up.)



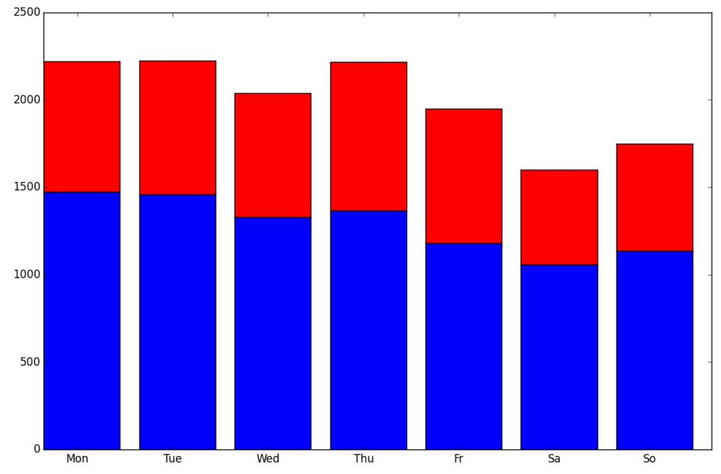
Age distribution





Split by nationality:

As expected, highest proportion German/Austrian and a long tail of other countries - roughly sortable into 4 large groups (DEU/AUT/ non-german EU/ non-EU)



User distribution by weekday on which they signed up:
Split implies fewer initial signups during weekend - though possibly skewed somewhat by time cut-off at end of dataset.

Part II: Comments on classifier choice

I decided to try out a decision tree-based classifier first as it naturally handles the categorical variables with very little pre-processing, and because this low-dimensional dataset already lends itself to rule-based classification by inspection. For instance, a very simple rule-based classification approach (e.g. 1. If passport type is null, then signup cannot be complete
2. If user near very end of dataset, less likely to have had time to complete signup)
would give reasonable results. Just testing for missing passport data by itself already gave ~65% accuracy in prediction.

The decision tree technique seemed to overfit the training data (100% fit on training but only ~80% on test data), so I moved to a random forest learner as next step. After eliminating several variables that I assume would only add noise average accuracy on multiple tests rose to about 96%. (Features deemed unnecessary for classification include last name and birthplace - while it would be possible to extract limited information from these, it would require processing against a larger reference database. Other variables like 'id' are already excluded, as well.)

For comparison, I also tried a SVM classifier but found similar performance at higher training times.