Fake News Research Project Final Report

Marian Longa, 08/09/2017

Supervisors: Axel Oehmichen, Miguel Molina-Solana

Data Science Institute, Imperial College London

Introduction

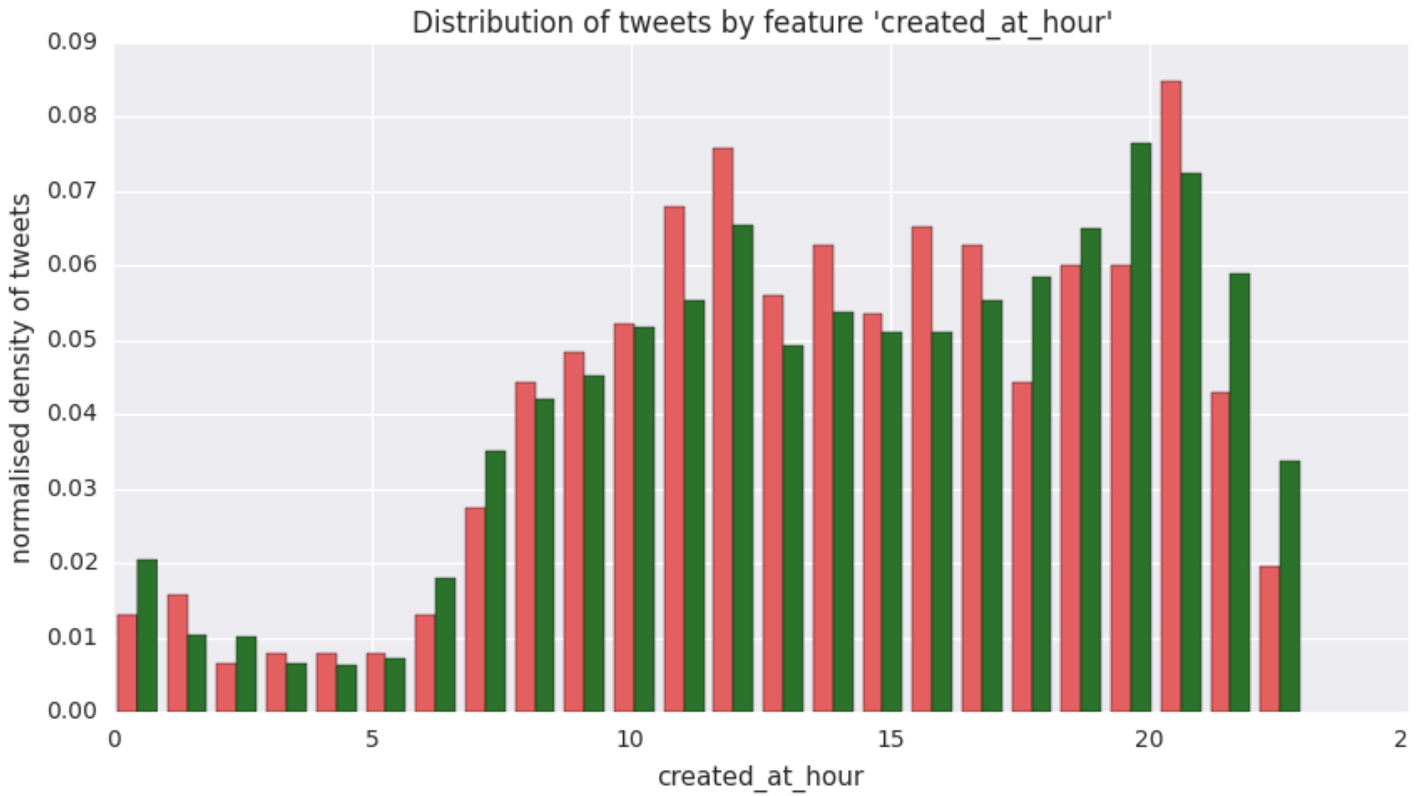
- **Problem**: given the metadata about tweets related to the 2016 US election, implement a classifier to best categorize the tweets as “fake news” and “other type of news”.
- **Solution**:
  1. Go through the list of tweets and manually label each one as “fake news” or “other type of news” (also label each “fake” tweet as one of 5 “fake news” subcategories)
  2. Use the tweet metadata to engineer base and derived features
  3. Calculate which features best separate the ”fake” and “other” news classes
  4. Use subsets of those features to create different feature sets and test the classification performance of each feature set using logistic regression
  5. Out of these, choose the feature set which obtains the highest classification score in logistic regression
  6. Use this best feature set to train and test different types of classifiers, varying their hyperparameters and noting the corresponding classification scores
  7. When the hyperparameters for each classifier model are optimized, compare the models and select the model with highest classification score

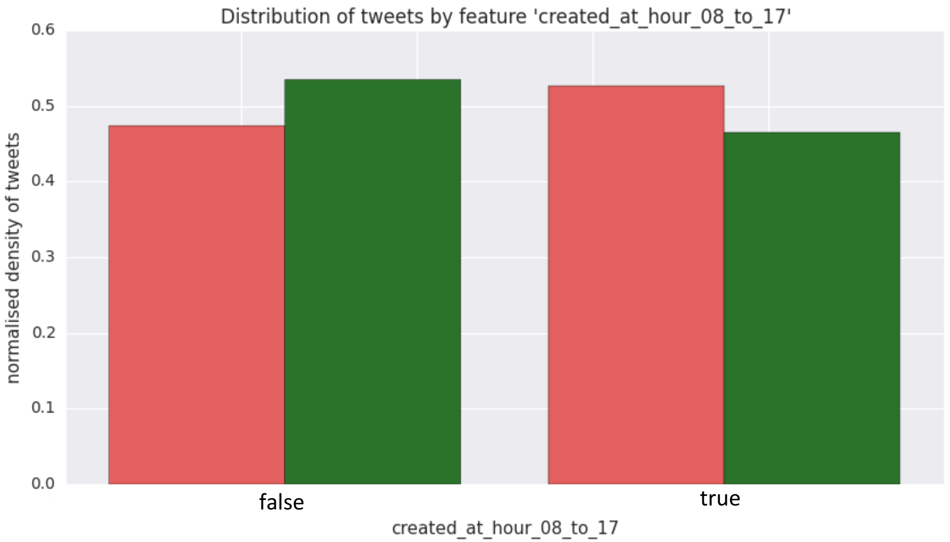
Feature engineering and selection

Feature selection method

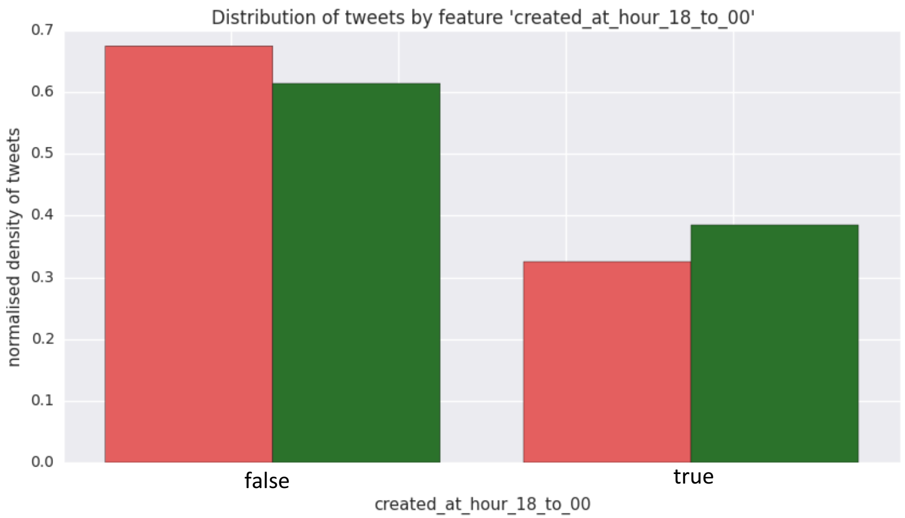
- Download 23 tweet fields from tweet database:
  tweet_id, created_at, retweet_count, text, user_screen_name, user_verified, user_friends_count, user_followers_count, user_favourites_count, tweet_source, geo_coordinates, num_hashtags, num_mentions, num_urls, num_media, user_default_profile_image, user_description, user_listed_count, user_name, user_profile_use_background_image, user_default_profile, user_statuses_count, user_created_at (green denotes newly added features w.r.t. previous paper)
- Define 85 base + derived features (check character types, calculate per-unit-time quantities, determine trends from histograms)
- For each feature calculate:
  - the difference in mean $\Delta\mu$ between 'fake' and 'other' classes after scaling the feature data to $\mu=0$, $\sigma=1$
  - the p-value corresponding to a t-test performed on the unscaled 'fake' and 'other' classes
- Eliminate features with high p-value

‘created_at_hour’ related features



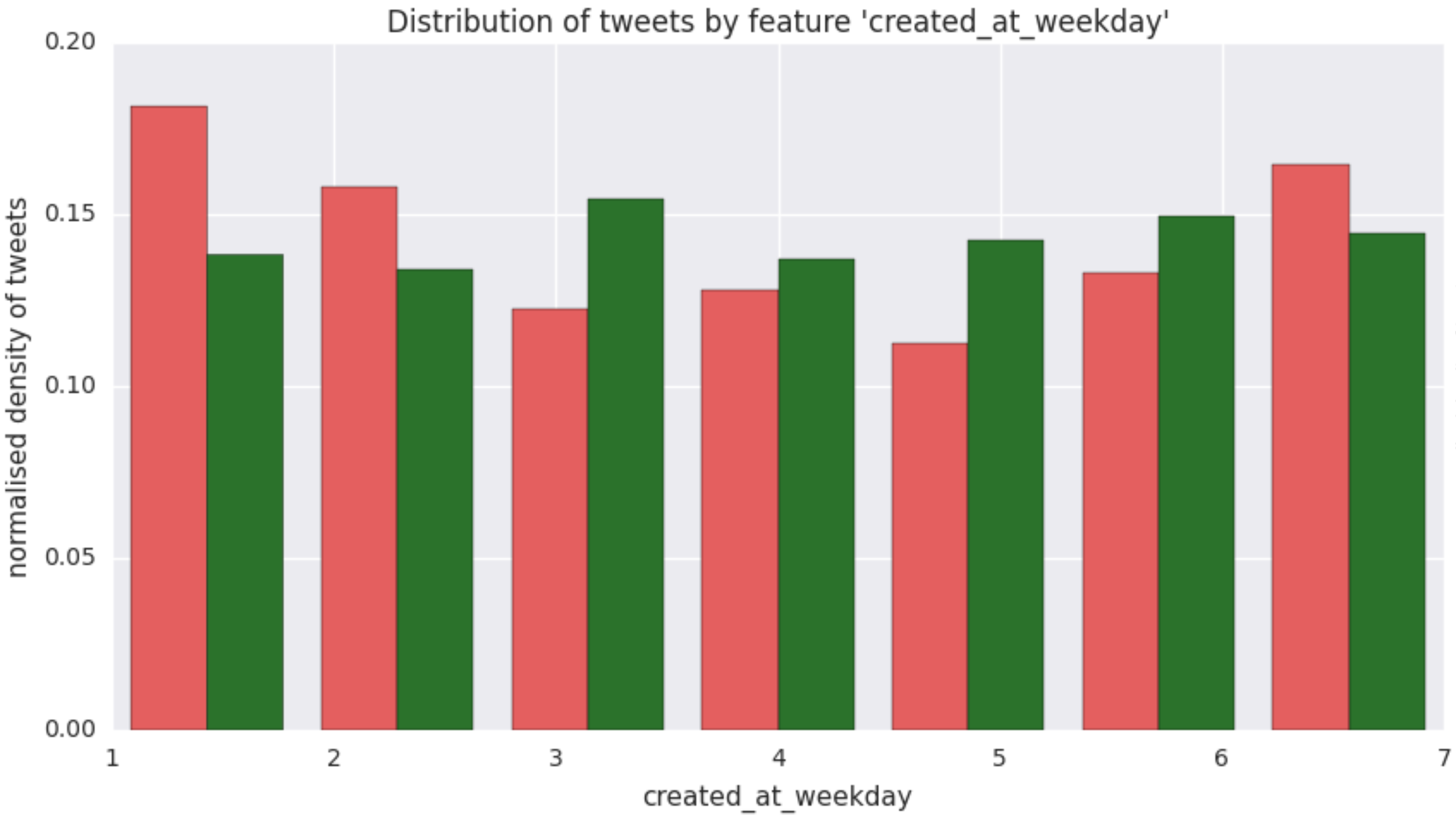


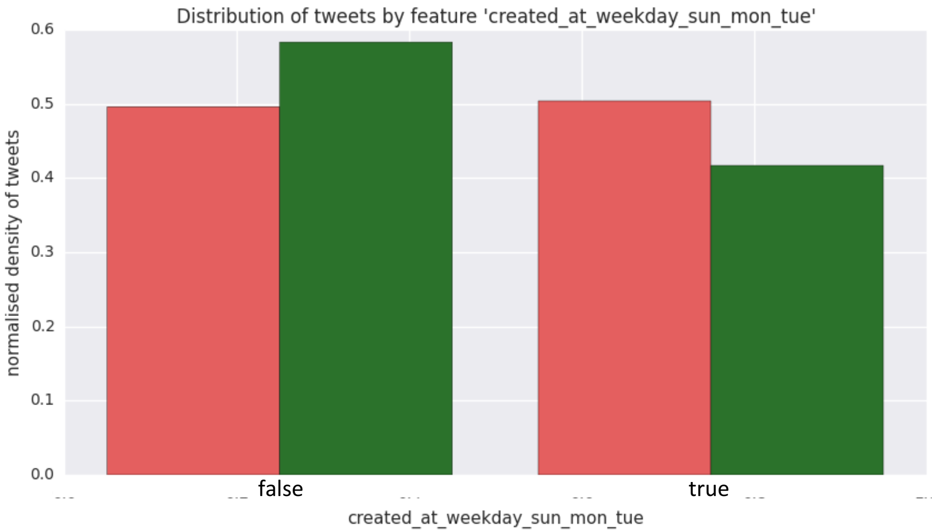
$\Delta\mu = 0.122$, $p = 0.00166$



$\Delta\mu = -0.125$, $p = 0.00124$

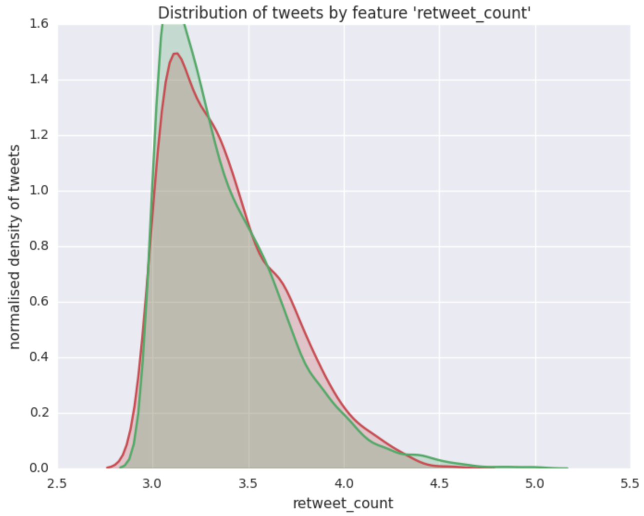
‘created_at_weekday’ related features



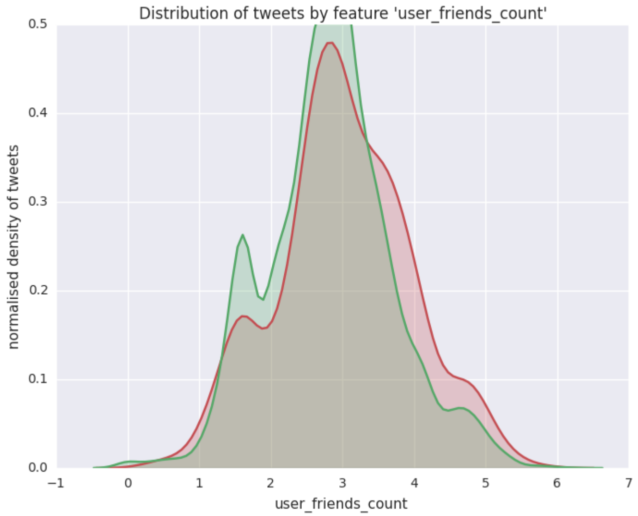


$\Delta\mu = 0.176$, $p = 0.00000532$

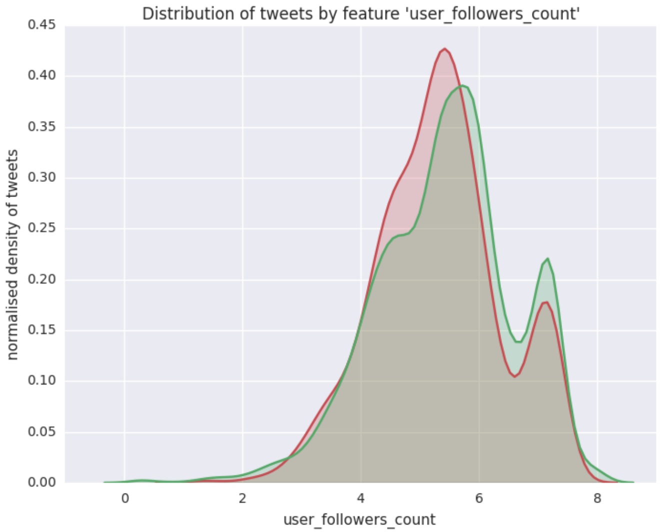
per-unit-time related features (log10)



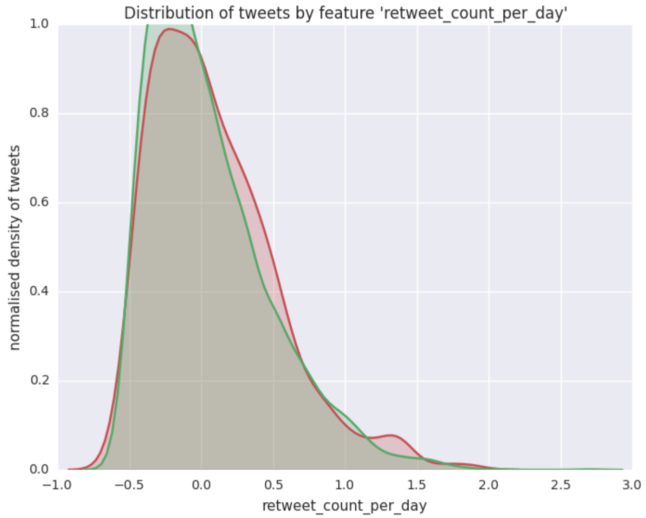
$\Delta\mu = -0.0247$, $p = 0.523$



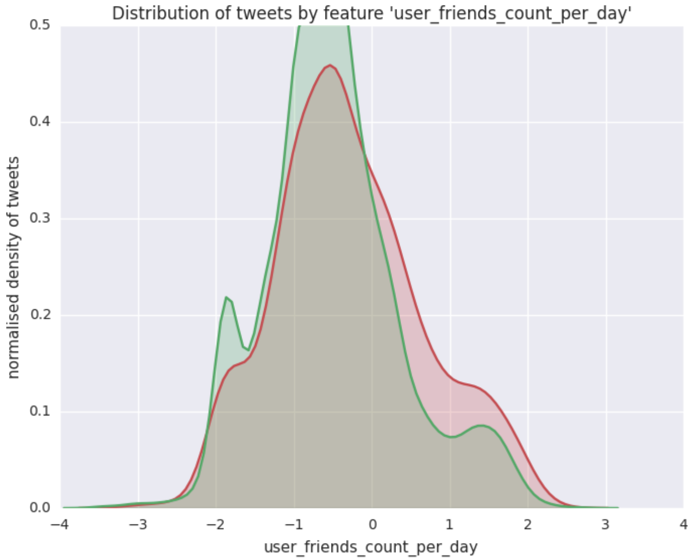
$\Delta\mu = 0.0830$, $p = 0.0320$



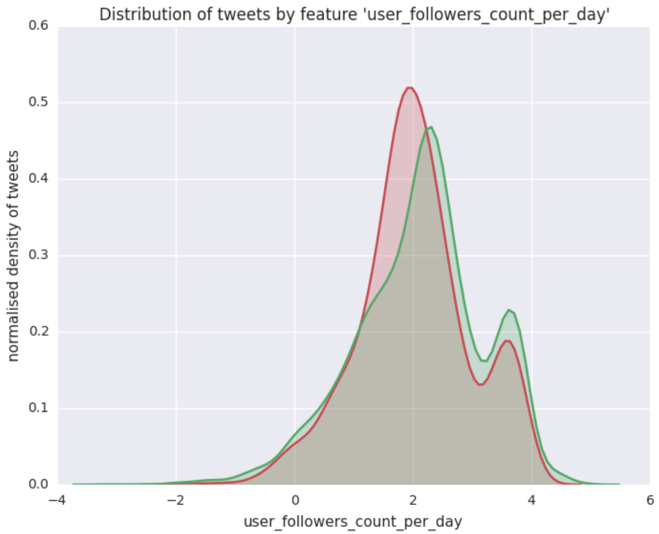
$\Delta\mu = -0.126$, $p = 0.00111$



$\Delta\mu = 0.0268$, $p = 0.489$

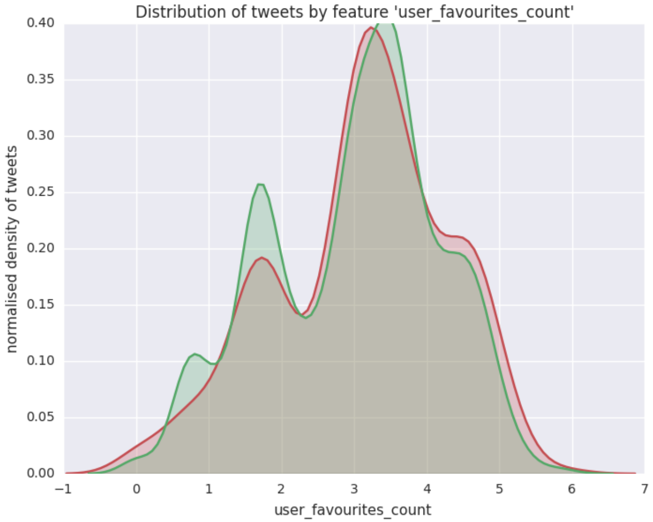


$\Delta\mu = 0.129$, $p = 0.000829$

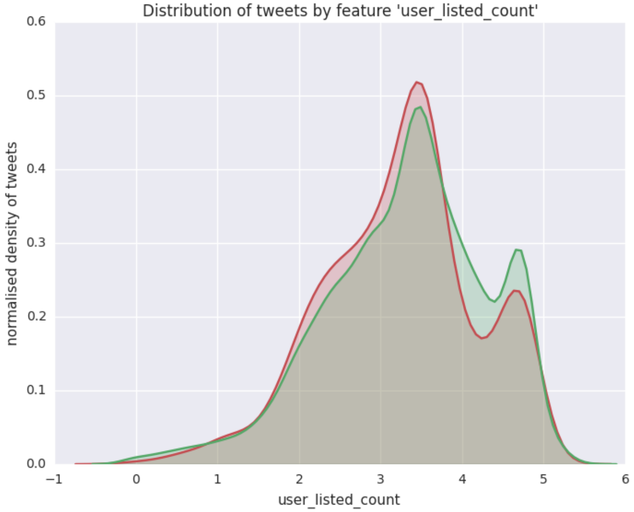


$\Delta\mu = -0.143$, $p = 0.000233$

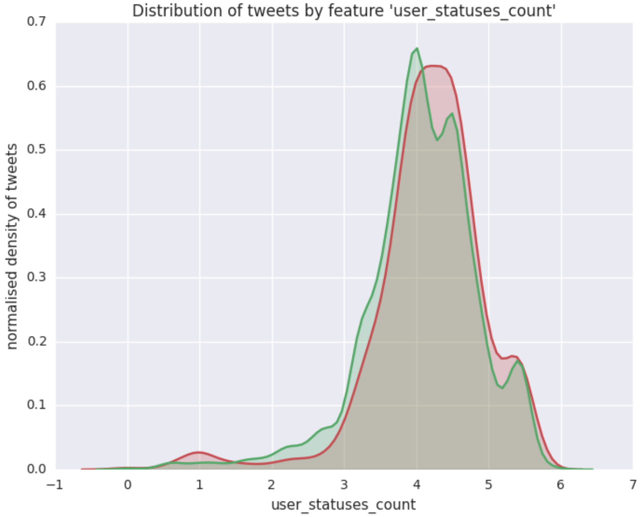
per-unit-time related features (log10)



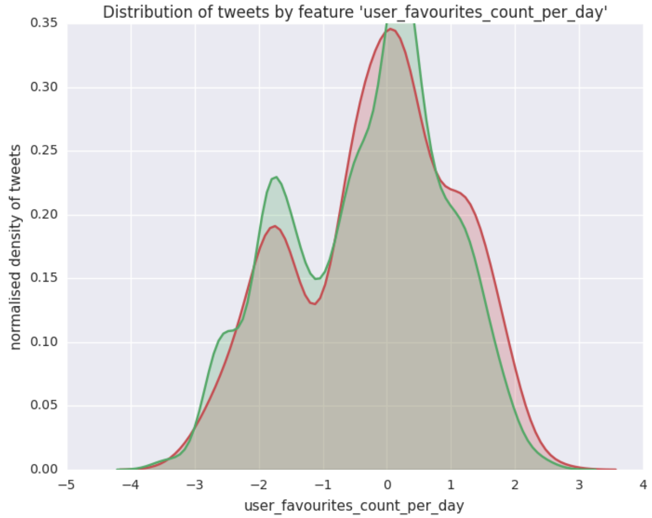
Δμ = 0.0680, p = 0.0790



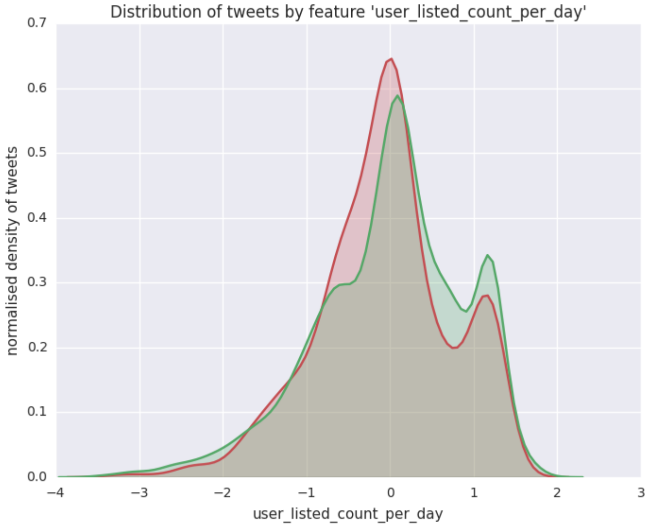
Δμ = -0.0991, p = 0.0105



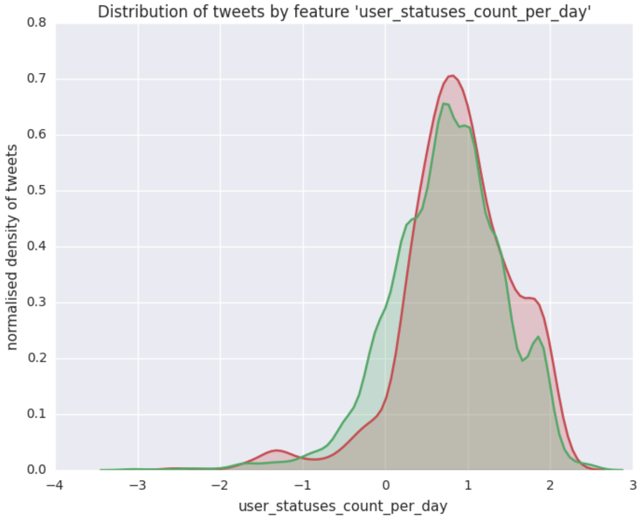
Δμ = 0.0767, p = 0.0476



Δμ = 0.0990, p = 0.0106



Δμ = -0.132, p = 0.000680



Δμ = 0.115, p = 0.00291

text-related features

FEATURE	DIFF MEAN	P VALUE
text_num_caps_digits	0.328508469805197	0.00000000000000001817
text_num_caps_digits_exclam	0.320376289091854	0.00000000000000011032
text_num_caps	0.284341784083254	0.00000000000018982640
text_num_caps_exclam	0.276391941495619	0.00000000000087281045
text_num_digits	0.272337246767137	0.00000000000186958317
text_num_swears	-0.115639271995914	0.00282836836204724000
text_num_nonstandard	0.063133372168354	0.10317168793059400000
text_num_nonstandard_extended	0.048553312639814	0.21011337294653800000
text_num_exclam	-0.009094815310897	0.81441095787411000000

FEATURE	DIFF MEAN	P VALUE
user_screen_name_has_caps_digits	0.262285988688136	0.00000000001178100595
user_screen_name_num_caps_digits	0.225172500229789	0.00000000588121100628
user_screen_name_has_caps_digits_underscores	0.220518463593140	0.00000001201798678693
user_screen_name_num_caps_digits_underscores	0.216318353886877	0.00000002262544180715
user_screen_name_has_caps	0.206996667603478	0.00000008838704498429
user_screen_name_num_caps	0.177737513482832	0.00000439979639098320
user_screen_name_num_caps_underscores	0.168496131447383	0.00001346469570402120
user_screen_name_has_caps_underscores	0.161481492528789	0.00003033387736915430
user_screen_name_has_digits	0.155118614476325	0.00006165950921821620
user_screen_name_num_digits	0.133110918511019	0.00058745594493451000
user_screen_name_num_digits_underscores	0.110560873515159	0.00430932407291492000
user_screen_name_num_weird_chars	0.110560873515159	0.00430932407291492000
user_screen_name_has_digits_underscores	0.060639335522454	0.11751832452650200000
user_screen_name_has_weird_chars	0.060639335522454	0.11751832452650200000
user_screen_name_has_underscores	-0.028897507148646	0.45574730213312300000
user_screen_name_num_underscores	-0.027552309134123	0.47699401643818400000

FEATURE	DIFF MEAN	P VALUE
user_description_num_exclam	0.150950653420117	0.00009675917875963300
user_description_num_caps_exclam	0.121901143581656	0.00164608597263901000
user_description_num_non_a_to_z	0.114943253249033	0.00299923689022872000
user_description_num_caps	0.114512461659785	0.00310965737703582000
user_description_num_non_a_to_z_non_digits	0.114507752043833	0.00311088476024148000
user_description_num_caps_with_num_nonstandard	0.114368994414807	0.00314724544039675000
user_description_num_nonstandard	0.084096404735919	0.02993281139725200000
user_description_num_nonstandard_extended	0.054021652644027	0.16318955311498100000
user_description_num_digits	0.039756151198316	0.30481632729323100000

FEATURE	DIFF MEAN	P VALUE
user_name_has_weird_chars	0.094534681286187	0.01466390719685070000
user_name_has_underscores	0.089929887442158	0.02025241432449590000
user_name_has_digits_underscores	0.085898104988500	0.02658832483519980000
user_name_has_nonprintable_chars	0.068094032521295	0.07878999542386130000
user_name_has_caps_digits	0.053377805119681	0.16826434210917700000
user_name_has_caps	0.045822150476645	0.23690878845441800000
user_name_num_digits_underscores	0.043565275499711	0.26080589854854900000
user_name_num_weird_chars	0.042632887635448	0.27114939598233300000
user_name_num_nonprintable_chars	0.041769560980110	0.28097418974511000000
user_name_has_caps_underscores	0.040632640844963	0.29427702016733700000
user_name_has_caps_digits_underscores	0.039421456149658	0.30890613237491100000
user_name_num_digits	0.036001457629479	0.35276549231467300000
user_name_num_underscores	0.033341590172670	0.38946950124127400000
user_name_has_digits	0.022438616439895	0.56248580648111200000
user_name_num_caps_digits_underscores	0.021013635357653	0.58756178677826800000
user_name_num_caps_underscores	0.015661773926583	0.68603906159413600000
user_name_num_caps_digits	0.009008668192290	0.81613728320399300000
user_name_num_caps	0.002906504401600	0.94020082866916800000

All features

FEATURE	DIFF MEAN	P VALUE
user_verified	-0.334862652593959	0.0000000000000000430
text_num_caps_digits	0.328508469805197	0.0000000000000001817
text_num_caps_digits_exclam	0.320376289091854	0.0000000000000011032
text_num_caps	0.284341784083254	0.0000000000018982640
text_num_caps_exclam	0.276391941495619	0.0000000000087281045
text_num_digits	0.272337246767137	0.0000000000186958317
user_screen_name_has_caps_digits	0.262285988688136	0.0000000001178100595
user_screen_name_num_caps_digits	0.225172500229789	0.0000000588121100628
user_screen_name_has_caps_digits_underscores	0.220518463593140	0.0000001201798678693
user_screen_name_num_caps_digits_underscores	0.216318353886877	0.0000002262544180715
user_screen_name_has_caps	0.206996667603478	0.0000008838704498429
num_urls_is_nonzero	0.193751829686201	0.0000055546626709527
num_urls	0.192759060318776	0.0000063457724714638
user_screen_name_num_caps	0.177737513482832	0.0000439979639098320
created_at_weekday_sun_mon_tue	0.176202799872474	0.0000531821008238821
user_screen_name_num_caps_underscores	0.168496131447383	0.0001346469570402120
user_screen_name_has_caps_underscores	0.161481492528789	0.0003033387736915430
user_screen_name_has_digits	0.155118614476325	0.0006165950921821620
user_description_num_exclam	0.150950653420117	0.0009675917875963300
user_followers_count_per_day	-0.142513168885046	0.0023286248761311800
user_screen_name_num_digits	0.133110918511019	0.0058745594493451000
user_listed_count_per_day	-0.131571815531587	0.0067990384942370200
user_friends_count_per_day	0.129452328814406	0.0082945532326780700
user_followers_count	-0.126288845970021	0.0111016076037015000
num_media	-0.126018499508719	0.0113783067671653000
created_at_hour_18_to_00	-0.125111745368415	0.0123535800778445000
num_media_is_nonzero	-0.125054182010775	0.0124180263026520000
user_description_num_caps_exclam	0.121901143581656	0.0164608597263901000
created_at_hour_08_to_17	0.121817302581855	0.0165832748771026000
text_num_swears	-0.115639271995914	0.0282836836204724000
user_statuses_count_per_day	0.115292888911431	0.0291226034527856000
user_description_num_non_a_to_z	0.114943253249033	0.0299923689022872000
user_description_num_caps	0.114512461659785	0.0310965737703582000
user_description_num_non_a_to_z_non_digits	0.114507752043833	0.0311088476024148000
user_description_num_caps_with_num_nonstandard	0.114368994414807	0.0314724544039675000
user_profile_use_background_image	0.110840963665571	0.0421215183874502000
user_screen_name_num_digits_underscores	0.110560873515159	0.0430932407291492000
user_screen_name_num_weird_chars	0.110560873515159	0.0430932407291492000

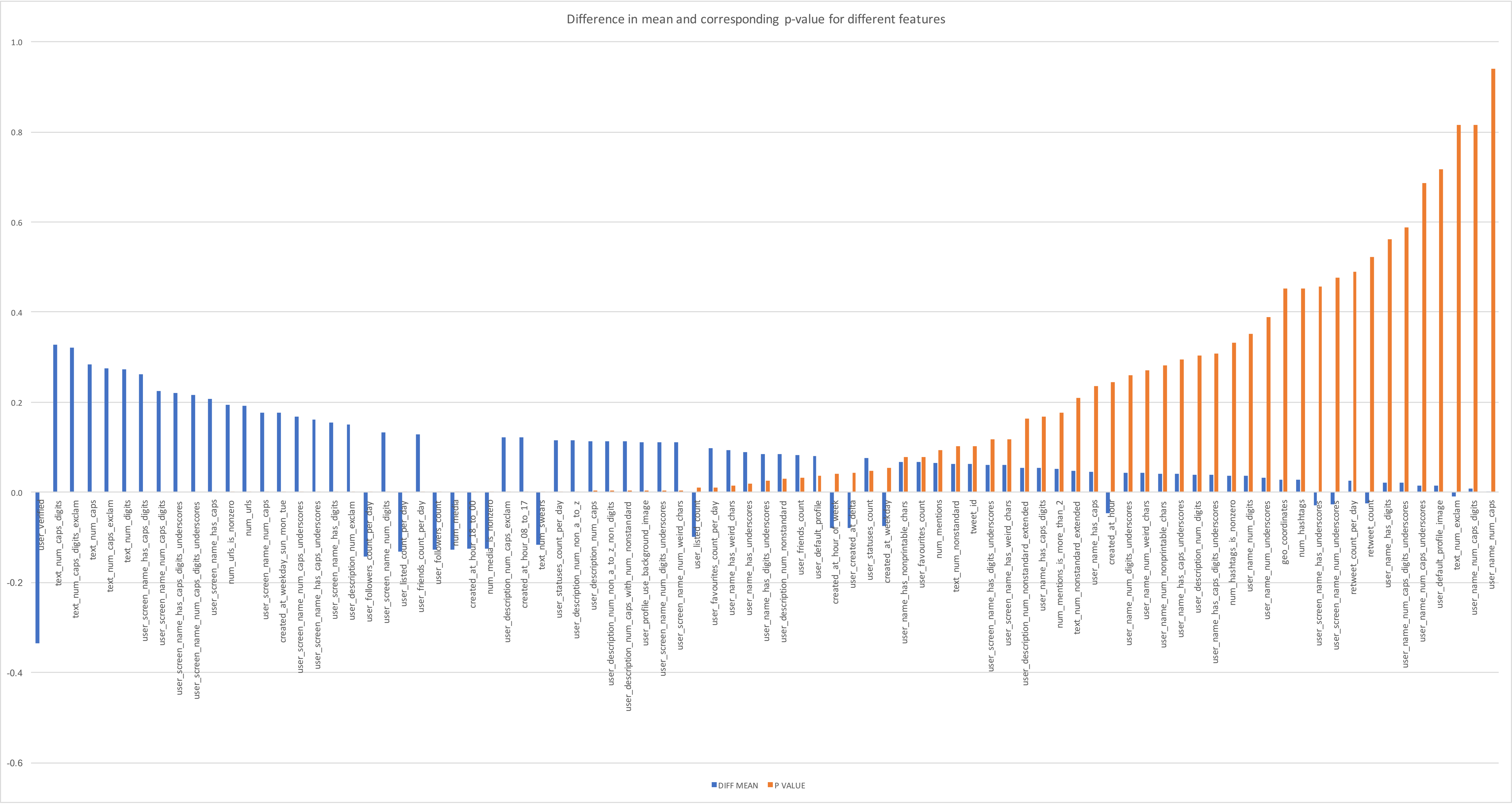
p < 0.01

user_listed_count	-0.099054697681362	0.0105479063059875000
user_favourites_count_per_day	0.098958024473082	0.0106238787884505000
user_name_has_weird_chars	0.094534681286187	0.0146639071968507000
user_name_has_underscores	0.089929887442158	0.0202524143244959000
user_name_has_digits_underscores	0.085898104988500	0.0265883248351998000
user_description_num_nonstandard	0.084096404735919	0.0299328113972520000
user_friends_count	0.083044103896057	0.0320487441994148000
user_default_profile	0.080376062248807	0.0379957617847991000
created_at_hour_of_week	-0.079448339664958	0.0402725193995434000
user_created_at_delta	-0.078195521824918	0.0435298236880818000
user_statuses_count	0.076734653944410	0.0476063268138459000

0.01 ≤ p < 0.05

created_at_weekday	-0.074366010607228	0.0548956729243725000
user_name_has_nonprintable_chars	0.068094032521295	0.0787899954238613000
user_favourites_count	0.068046610466555	0.0789986523545469000
num_mentions	0.064822633303216	0.0942715434459716000
text_num_nonstandard	0.063133372168354	0.1031716879305940000
tweet_id	0.063049950829170	0.1036279492557680000
user_screen_name_has_digits_underscores	0.060639335522454	0.1175183245265020000
user_screen_name_has_weird_chars	0.060639335522454	0.1175183245265020000
user_description_num_nonstandard_extended	0.054021652644027	0.1631895531149810000
user_name_has_caps_digits	0.053377805119681	0.1682643421091770000
num_mentions_is_more_than_2	0.052343384253203	0.1766642497996150000
text_num_nonstandard_extended	0.048553312639814	0.2101133729465380000
user_name_has_caps	0.045822150476645	0.2369087884544180000
created_at_hour	-0.045140846024975	0.2439538538775120000
user_name_num_digits_underscores	0.043565275499711	0.2608058985485490000
user_name_num_weird_chars	0.042632887635448	0.2711493959823330000
user_name_num_nonprintable_chars	0.041769560980110	0.2809741897451100000
user_name_has_caps_underscores	0.040632640844963	0.2942770201673370000
user_description_num_digits	0.039756151198316	0.3048163272932310000
user_name_has_caps_digits_underscores	0.039421456149658	0.3089061323749110000
num_hashtags_is_nonzero	0.037645605063081	0.3312101913614200000
user_name_num_digits	0.036001457629479	0.3527654923146730000
user_name_num_underscores	0.033341590172670	0.3894695012412740000
geo_coordinates	0.029147327335745	0.4518609713848420000
num_hashtags	0.029147327335745	0.4518609713848420000
user_screen_name_has_underscores	-0.028897507148646	0.4557473021331230000
user_screen_name_num_underscores	-0.027552309134123	0.4769940164381840000
retweet_count_per_day	0.026792257249070	0.4892342397219030000
retweet_count	-0.024747089874719	0.5229932886129190000
user_name_has_digits	0.022438616439895	0.5624858064811120000
user_name_num_caps_digits_underscores	0.021013635357653	0.5875617867782680000
user_name_num_caps_underscores	0.015661773926583	0.6860390615941360000
user_default_profile_image	0.014059915293107	0.7166862245812530000
text_num_exclam	-0.009094815310897	0.8144109578741100000
user_name_num_caps_digits	0.009008668192290	0.8161372832039930000
user_name_num_caps	0.002906504401600	0.9402008286691680000

p ≥ 0.05



Difference in mean and corresponding p-value for different features
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
user_verified
text_num_caps_digits
text_num_caps_digits_exclam
text_num_caps
text_num_caps_exclam
text_num_digits
user_screen_name_has_caps_digits
user_screen_name_num_caps_digits
user_screen_name_has_caps_digits_underscores
user_screen_name_num_caps_digits_underscores
user_screen_name_has_caps
num_urls_is_nonzero
num_urls
user_screen_name_num_caps
created_at_weekday_sun_mon_tue
user_screen_name_num_caps_underscores
user_screen_name_has_caps_underscores
user_screen_name_has_digits
user_description_num_exclam
user_followers_count_per_day
user_screen_name_num_digits
user_listed_count_per_day
user_friends_count_per_day
user_followers_count
num_media
created_at_hour_18_to_00
num_media_is_nonzero
user_description_num_caps_exclam
created_at_hour_08_to_17
text_num_swears
user_statuses_count_per_day
user_description_num_non_a_to_z
user_description_num_caps
user_description_num_non_a_to_z_non_digits
user_description_num_caps_with_num_nonstandard
user_profile_use_background_image
user_screen_name_num_digits_underscores
user_screen_name_num_weird_chars
user_listed_count
user_favourites_count_per_day
user_name_has_weird_chars
user_name_has_underscores
user_name_has_digits_underscores
user_description_num_nonstandard
user_friends_count
user_default_profile
created_at_hour_of_week
user_created_at_delta
user_statuses_count
created_at_weekday
user_name_has_nonprintable_chars
user_favourites_count
num_mentions
text_num_nonstandard
tweet_id
user_screen_name_has_digits_underscores
user_screen_name_has_weird_chars
user_description_num_nonstandard_extended
user_name_has_caps_digits
num_mentions_is_more_than_2
text_num_nonstandard_extended
user_name_has_caps
created_at_hour
user_name_num_digits_underscores
user_name_num_weird_chars
user_name_num_nonprintable_chars
user_name_has_caps_underscores
user_description_num_digits
user_name_has_caps_digits_underscores
num_hashtags_is_nonzero
user_name_num_digits
user_name_num_underscores
geo_coordinates
num_hashtags
user_screen_name_has_underscores
user_screen_name_num_underscores
retweet_count_per_day
retweet_count
user_name_has_digits
user_name_num_caps_digits_underscores
user_name_num_caps_underscores
user_default_profile_image
text_num_exclam
user_name_num_caps_digits
user_name_num_caps
DIFF MEAN
P VALUE

Model performance evaluation

Evaluation method

- Use the **same feature set** for evaluation of all models → consistency in results
- Use **K-fold cross-validation** (k=5) → decrease variance of model scores
- **Upsample minority class** (‘fake news’) to 1:1 during training, while keeping original class proportions (~1:8) for testing → if there was no upsampling, the classifier would learn to classify all data as ‘other news’ to maximize accuracy
- Test ***logistic regression*, *SVM*, *KNN*, *random forest*** models with different hyperparameters and note the results → use grid search to loop through relevant ranges for hyperparameters
- For each model note the model parameters which **maximize the ROC AUC score** (maximizing accuracy causes all data to be classified as ‘other news’ due to imbalanced data set, therefore accuracy is not a good metric here)

Logistic Regression – testing method

- Use logistic regression model with *liblinear* solver and *l1* penalty
- Run logistic regression with different feature sets, note resulting performances
- Choose the feature set with high ROC AUC value and reasonable features included (don’t include all features since this may cause overfitting)

Logistic Regression – results

results are sorted by mean ROC AUC score

feature_set	mean_accuracy_score	mean_roc_auc_score	mean_precision_score	mean_recall_score	mean_f1_score	mean_cm_TN	mean_cm_FP	mean_cm_FN	mean_cm_TP
features_extended_some_multiple	0.622322627	0.656842461	0.195162758	0.608326967	0.295385728	640.4	385.2	60	93.2
features_extended_some_single	0.615196504	0.653507112	0.193286013	0.614854427	0.293944347	631	394.6	59	94.2
features_extended_some_multiple_without_text_num_swears	0.622322196	$\boxed{0.651971269}$	0.192110256	0.592640693	0.290029959	642.8	382.8	62.4	90.8
features_extended_some_multiple_without_biasing_features	0.624354655	0.651012711	0.192117706	0.587394958	0.289382631	646	379.6	63.2	90
features_extended_all_reduced	0.631823648	0.650918979	0.196633277	0.592717087	0.295247988	654	371.6	62.4	90.8
features_extended_all	0.633348927	0.648506836	0.197754208	0.594015788	0.296644288	655.6	370	62.2	91
features_extended_some_single_without_biasing_features	0.614515947	0.648401415	0.191212703	0.604371446	0.290262392	631.8	393.8	60.6	92.6
features_extended_few_multiple	0.614175526	0.639212461	0.184805381	0.576911977	0.279832426	635.6	390	64.8	88.4
features_extended_few_single	0.603828348	0.638716343	0.177606619	0.565164248	0.270145401	625.2	400.4	66.6	86.6
features_basic_some	0.624547881	0.596672056	0.171323113	0.480256345	0.249697647	662.6	363	79.6	73.6
features_basic_all	0.622330391	0.592582537	0.169846611	0.48424582	0.250343994	659.4	366.2	79	74.2
features_basic_few	0.615383831	0.58801292	0.170008974	0.486800781	0.247907816	650.8	374.8	78.6	74.6

features_extended_some_multiple has the highest ROC AUC score but text_num_swears wasn't a good feature → choose features_extended_some_multiple_without_text_num_swears instead

for details on which features are included in which feature set, please see the source of *models.py* file

Logistic regression – chosen feature set

- *features_extended_some_multiple_without_text_num_swears* feature set contains the following features:

user_verified	user_followers_count	num_media
text_num_caps	user_statuses_count_per_day	created_at_hour_18_to_00
text_num_digits	user_description_num_caps	user_profile_use_background_image
user_screen_name_has_caps	user_favourites_count_per_day	created_at_weekday
user_screen_name_has_digits	user_name_has_weird_chars	user_listed_count
num_urls_is_nonzero	user_default_profile	created_at_hour
user_description_num_exclam	created_at_weekday_sun_mon_tue	user_friends_count
user_followers_count_per_day	created_at_hour_08_to_17	user_created_at_delta
user_listed_count_per_day	user_friends_count_per_day	user_statuses_count

- The same feature set is used for training SVM, KNN, Random Forests

SVM – testing method

- Determine performances of SVM model with different hyperparameters using grid search:
  - Kernel $\in$ {linear, polynomial, RBF, sigmoid}
    - Polynomial: degree $\in$ {2, 3, 4, 5}
    - RBF: gamma $\in$ {?}
  - Maximum number of iterations $\in$ {1, 5} * 10^{1 ,2, 3, 4, 5, 6}
  - C $\in$ {1, 5} * 10^{-15, -14, …, 14, 15}

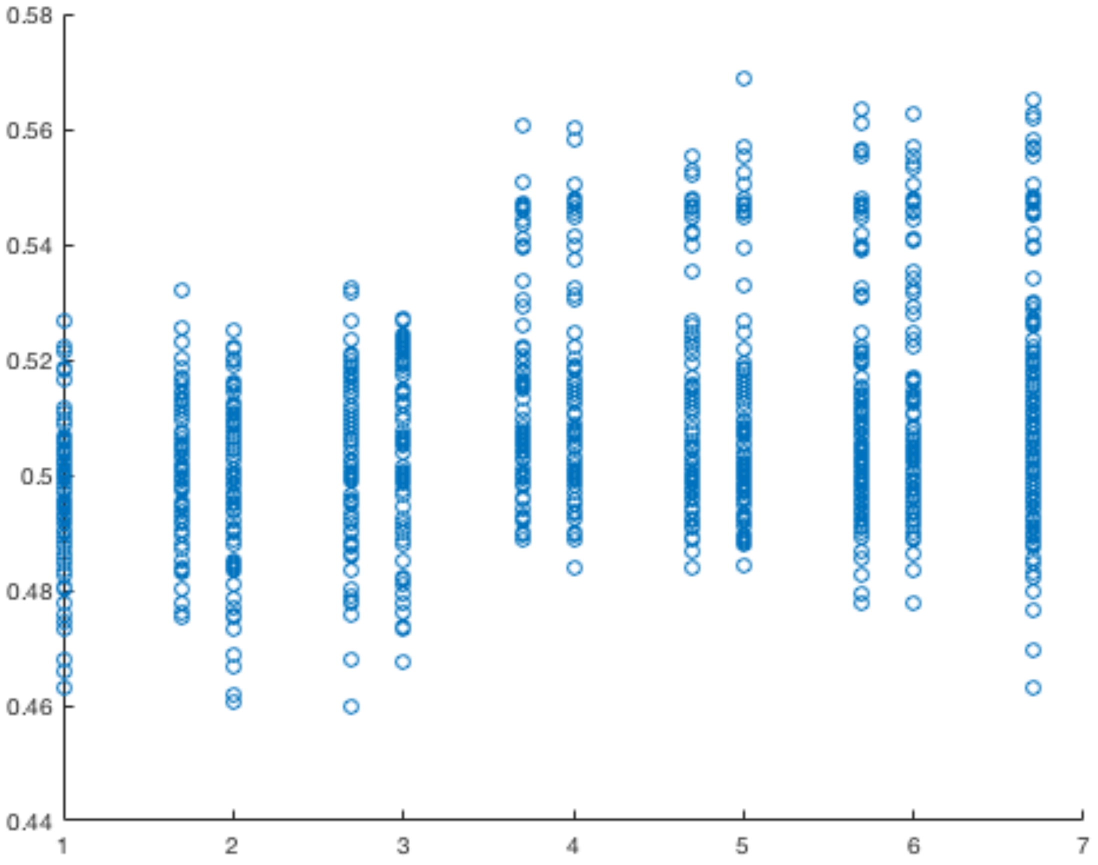
SVM – results

results are sorted by mean ROC AUC score

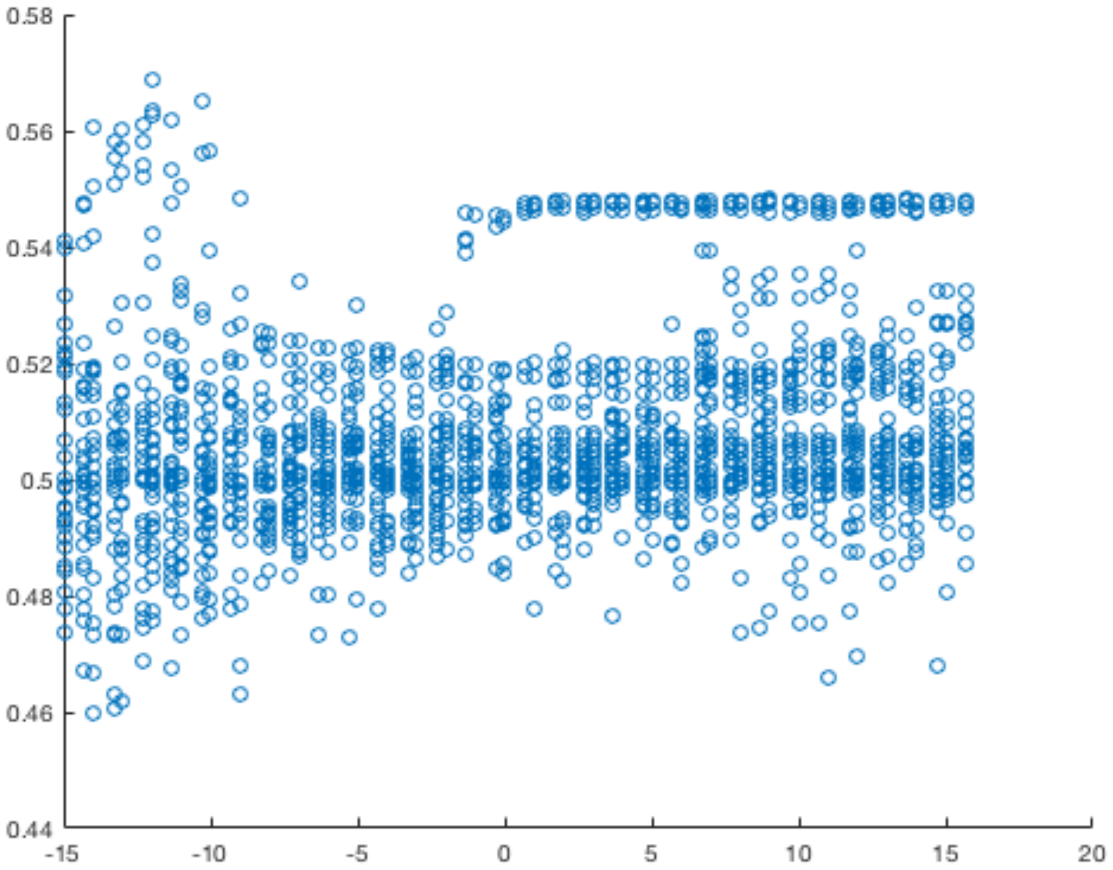
kernel	max_iter	poly_degree	C	mean_accuracy_score	mean_roc_auc_score	mean_precision_score	mean_recall_score	mean_f1_score	mean_cm_TN	mean_cm_FP	mean_cm_FN	mean_cm_TP
linear	100000	0	1.00E-12	0.467594389	0.568646156	0.110392233	0.566114931	0.166113391	464.2	561.4	66.4	86.8
linear	5000000	0	5.00E-11	0.538071355	0.565076114	0.139573531	0.484475002	0.189029666	560	465.6	79	74.2
linear	500000	0	1.00E-12	0.393067042	0.563481675	0.137912132	0.678533232	0.213361743	359.2	666.4	49.2	104
linear	1000000	0	1.00E-12	0.393067042	0.562555564	0.137912132	0.678533232	0.213361743	359.2	666.4	49.2	104
linear	5000000	0	1.00E-12	0.393067042	0.562555564	0.137912132	0.678533232	0.213361743	359.2	666.4	49.2	104
linear	5000000	0	5.00E-12	0.48993346	0.561898114	0.142832409	0.567625838	0.20084771	490.4	535.2	66.2	87
linear	500000	0	5.00E-13	0.225155471	0.561176437	0.134262577	0.909905781	0.233963356	126	899.6	13.8	139.4
linear	5000	0	1.00E-14	0.1409936	0.560792897	0.131291594	0.998692811	0.232072698	13.2	1012.4	0.2	153
linear	10000	0	1.00E-13	0.294977365	0.560208623	0.105879402	0.796078431	0.186894117	225.6	800	31.2	122
linear	5000000	0	5.00E-13	0.225155471	0.558144738	0.134262577	0.909905781	0.233963356	126	899.6	13.8	139.4
linear	10000	0	5.00E-14	0.352140305	0.558007325	0.105653283	0.71517698	0.184102455	305.4	720.2	43.6	109.6

showing first 11 out of 5208 results

SVM – graphs



ROC AUC score vs log10(maximum number of iterations)



ROC AUC score vs log10(C)

KNN – testing method

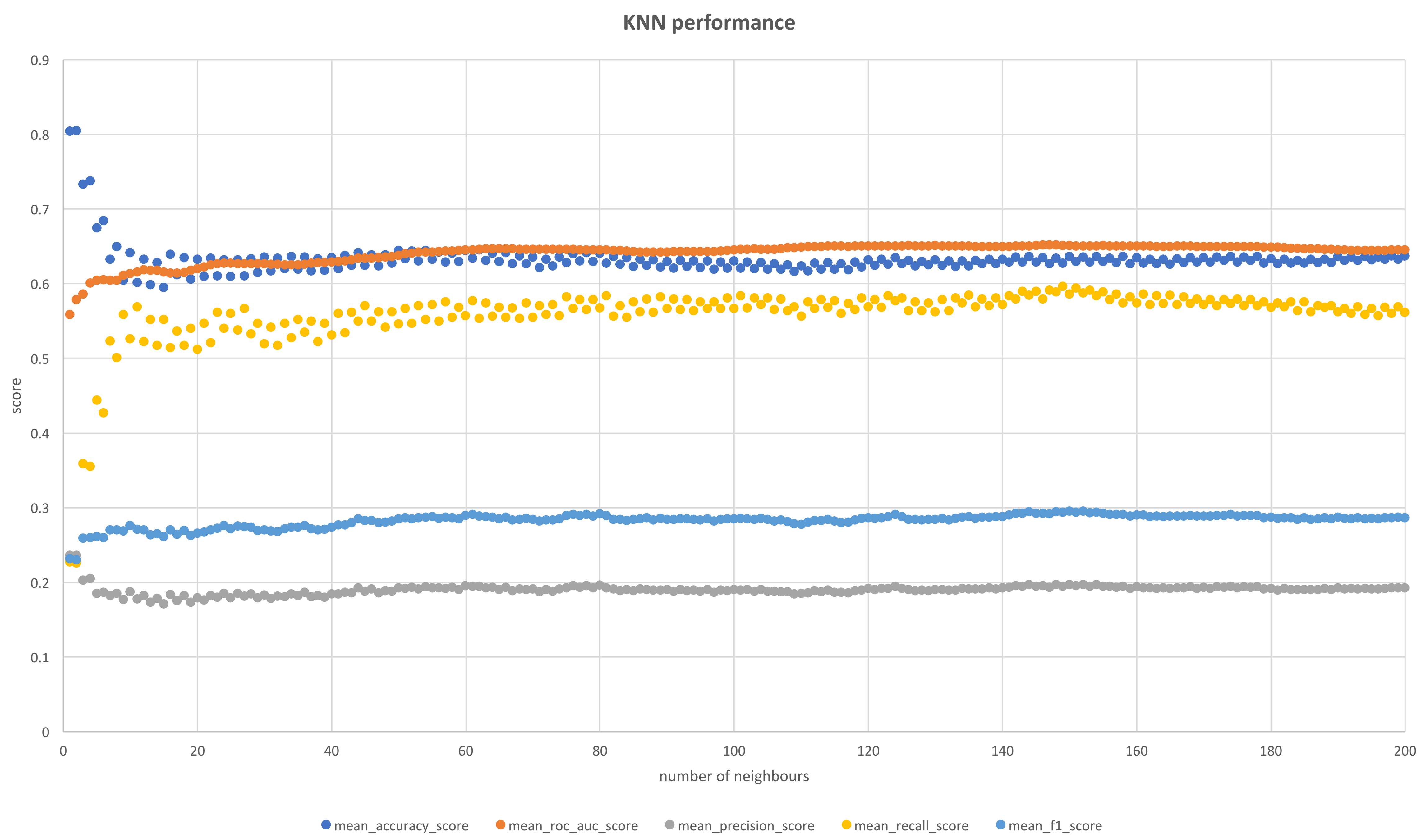
- Use K nearest neighbours model with $k \in \{1, 2, \ldots, 199, 200\}$
- Determine for which *k* the ROC AUC score is maximised

KNN – results

results are sorted by mean ROC AUC score

n_neighbors	mean_accuracy_score	mean_roc_auc_score	mean_precision_score	mean_recall_score	mean_f1_score	mean_cm_TN	mean_cm_FP	mean_cm_FN	mean_cm_TP
147	0.626736892	0.651695297	0.193487404	0.591460827	0.291542832	648.2	377.4	62.6	90.6
148	0.634371634	0.651618482	0.196732336	0.588846448	0.294884886	657.6	368	63	90.2
146	0.635220243	0.651606691	0.195335752	0.579704609	0.292156503	660	365.6	64.4	88.8
149	0.627415001	0.651155166	0.194895783	0.596672608	0.293778749	648.2	377.4	61.8	91.4
145	0.629111355	0.651019064	0.194449911	0.590136661	0.292473619	651.2	374.4	62.8	90.4
150	0.636408841	0.65096015	0.197294279	0.58622358	0.29519591	660.4	365.2	63.4	89.8
130	0.631992421	0.650926866	0.190302579	0.562728122	0.284347345	658.8	366.8	67	86.2
126	0.631484667	0.650878184	0.190220352	0.564043799	0.284417103	658	367.6	66.8	86.4
155	0.629619973	0.650802431	0.194451533	0.588837959	0.292308793	652	373.6	63	90.2
...									
10	0.641667536	0.613180275	0.187246933	0.526135303	0.276128915	675.8	349.8	72.6	80.6
9	0.604852777	0.610886	0.17684494	0.558764112	0.26855781	627.4	398.2	67.6	85.6
6	0.68459449	0.605680388	0.186846178	0.426924709	0.259864136	741.6	284	87.8	65.4
8	0.649475941	0.604930561	0.185388081	0.501358119	0.270540613	688.8	336.8	76.4	76.8
7	0.632847793	0.604576374	0.18197415	0.523546388	0.269947098	665.8	359.8	73	80.2
5	0.674753049	0.604214471	0.185392004	0.443909685	0.261487639	727.4	298.2	85.2	68
4	0.737359718	0.600826355	0.205092487	0.355122655	0.259885785	814.8	210.8	98.8	54.4
3	0.733117539	0.585738487	0.202611979	0.359018759	0.258905428	809.2	216.4	98.2	55
2	0.804720639	0.578425738	0.235930246	0.225914608	0.230637165	914	111.6	118.6	34.6
1	0.804381368	0.558911046	0.236117732	0.227221798	0.231429117	913.4	112.2	118.4	34.8

showing first 9 and last 10 results out of 200



KNN performance
score
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
20
40
60
80
100
120
140
160
180
200
number of neighbours
mean_accuracy_score
mean_roc_auc_score
mean_precision_score
mean_recall_score
mean_f1_score

Random Forest – testing method

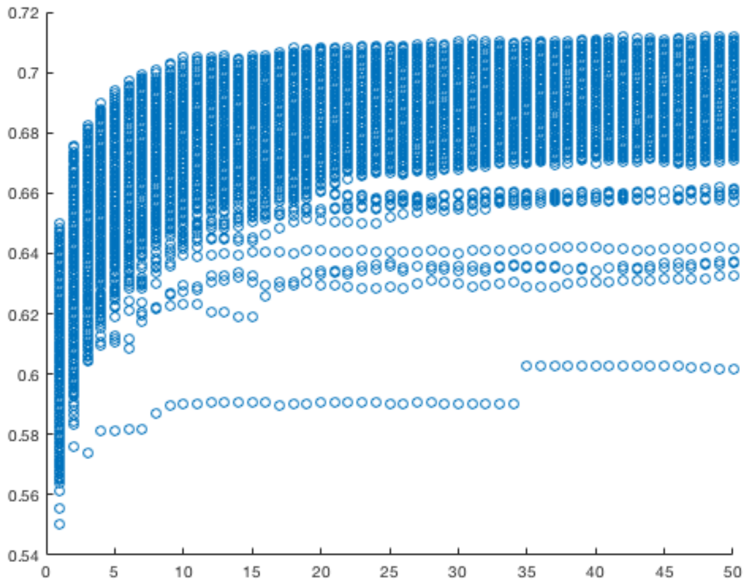
- Determine performances of Random Forest model with different hyperparameters using grid search:
  - Number of estimators (trees) $\in$ {1, 2, …, 50}
  - Maximum tree depth $\in$ {unlimited, 1, 2, 3, …, 50}
  - Minimum number of samples required in a leaf $\in$ {1, 2, …, 25}
  - Maximum number of features to use when looking for a split $\in \{\sqrt{n},\ \log_2 n,\ 0.5\,n,\ n\}$

Random Forest – results

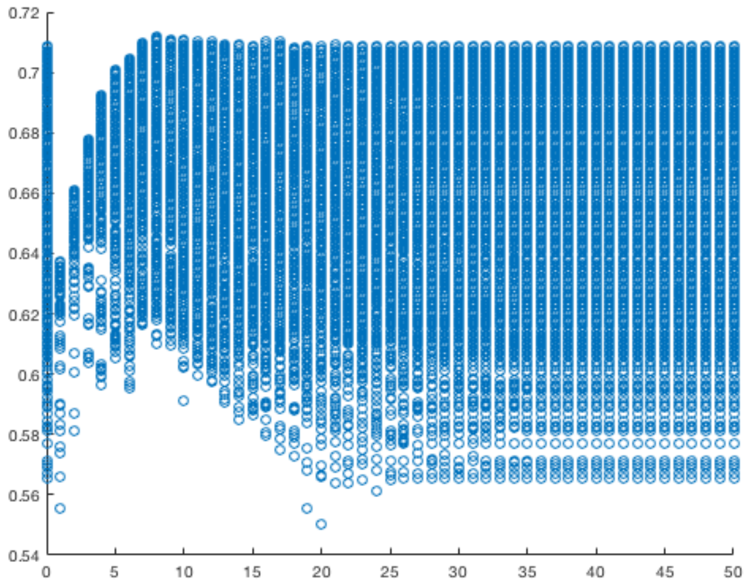
results are sorted by mean ROC AUC score

n_estimators	max_depth	min_samples_leaf	max_features	mean_accuracy_score	mean_roc_auc_score	mean_precision_score	mean_recall_score	mean_f1_score	mean_cm_TN	mean_cm_FP	mean_cm_FN	mean_cm_TP
49	8	12	sqrt	0.738544288	$\boxed{0.7119453}$	0.257725739	0.539156269	0.348556925	788	237.6	70.6	82.6
48	8	12	sqrt	0.739223548	0.711868295	0.259080159	0.541770648	0.350353495	788.4	237.2	70.2	83
50	8	12	sqrt	0.738885141	0.711830041	0.258341597	0.540471946	0.349410962	788.2	237.4	70.4	82.8
42	8	12	sqrt	0.738374796	0.71165972	0.25703447	0.5352347	0.34703612	788.4	237.2	71.2	82
41	8	12	sqrt	0.735999613	0.71155856	0.25447047	0.533935999	0.344456996	785.8	239.8	71.4	81.8
44	8	12	sqrt	0.738544144	0.711412076	0.257473007	0.53654189	0.347796489	788.4	237.2	71	82.2
47	8	12	sqrt	0.739222397	0.711326597	0.258337936	0.539156269	0.349101734	788.8	236.8	70.6	82.6
43	8	12	sqrt	0.740919182	0.711210194	0.260333402	0.539156269	0.350885531	790.8	234.8	70.6	82.6
46	8	12	sqrt	0.739223836	0.711100421	0.257837675	0.53654189	0.348090207	789.2	236.4	71	82.2
39	8	12	sqrt	0.736339316	0.711016406	0.255730053	0.539164757	0.346745453	785.4	240.2	70.6	82.6
50	10	15	log2	0.762299857	0.710971439	0.266111063	0.471267295	0.340122542	826.4	199.2	81	72.2
50	9	17	log2	0.751612397	0.710956638	0.269200335	0.531313131	0.35729071	804.6	221	71.8	81.4
49	9	17	log2	0.75110306	0.710885468	0.267104693	0.52479416	0.353986003	805	220.6	72.8	80.4
40	8	12	sqrt	0.737526619	0.710829466	0.257542747	0.541779136	0.348939529	786.4	239.2	70.2	83

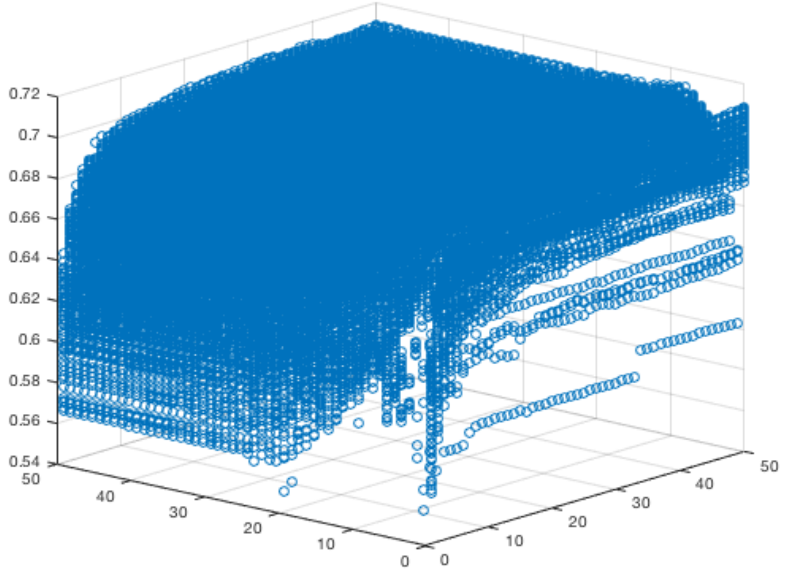
Random Forest – graphs



ROC AUC score vs number of estimators



ROC AUC score vs maximum tree depth (zero depth = unlimited depth)



ROC AUC score vs number of estimators and maximum tree depth

Best model

best results for each model are sorted by mean ROC AUC score

model	model-specific hyperparameters				mean_accuracy_score	mean_roc_auc_score	mean_precision_score	mean_recall_score	mean_f1_score	mean_cm_TN	mean_cm_FP	mean_cm_FN	mean_cm_TP
	n_estimators	max_depth	min_samples_leaf	max_features									
random forest	49	8	12	sqrt	0.738544288	$\boxed{0.711945300}$	0.257725739	0.539156269	0.348556925	788.0	237.6	70.6	82.6
	feature_set												
logistic regression	features_extended_some_multiple_without_text_num_swears				0.622322196	0.651971269	0.192110256	0.592640693	0.290029959	642.8	382.8	62.4	90.8
	n_neighbors												
KNN	147				0.626736892	0.651695297	0.193487404	0.591460827	0.291542832	648.2	377.4	62.6	90.6
	kernel	max_iter	poly_degree	C									
SVM	linear	100000	N/A	1.00E-12	0.467594389	0.568646156	0.110392233	0.566114931	0.166113391	464.2	561.4	66.4	86.8

- best model: **Random Forest** (n=49, depth=8, min_samples_leaf=12, max_features=sqrt)
- with ROC AUC = 71.2%, accuracy = 73.9%, precision = 25.8%, recall 53.9%

Future work

- Manually label more tweets and rerun the pipelines with more data
- Perform analysis using also the 5 subcategories of fake news
- Try new classification models
  - neural networks, naive Bayes?