Table of *contents*

01

Classifications with artificial data

02

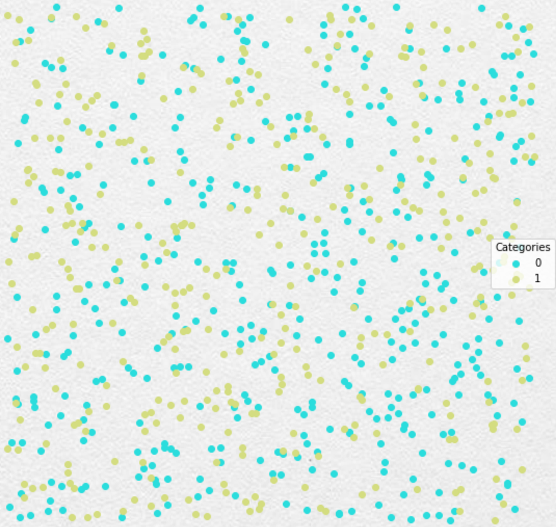
Classifications with real data

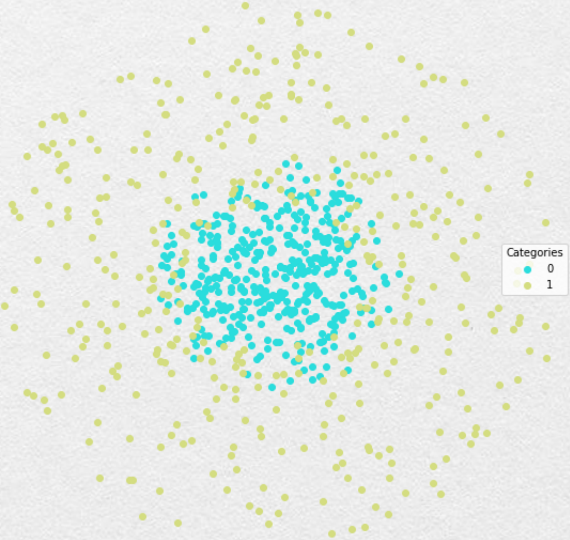
03

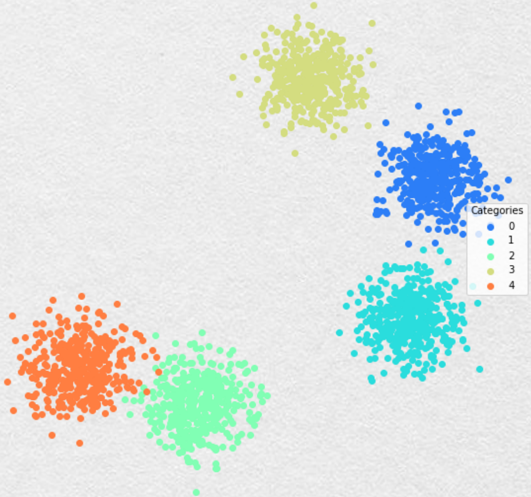
Explore BERT

Artificial datasets

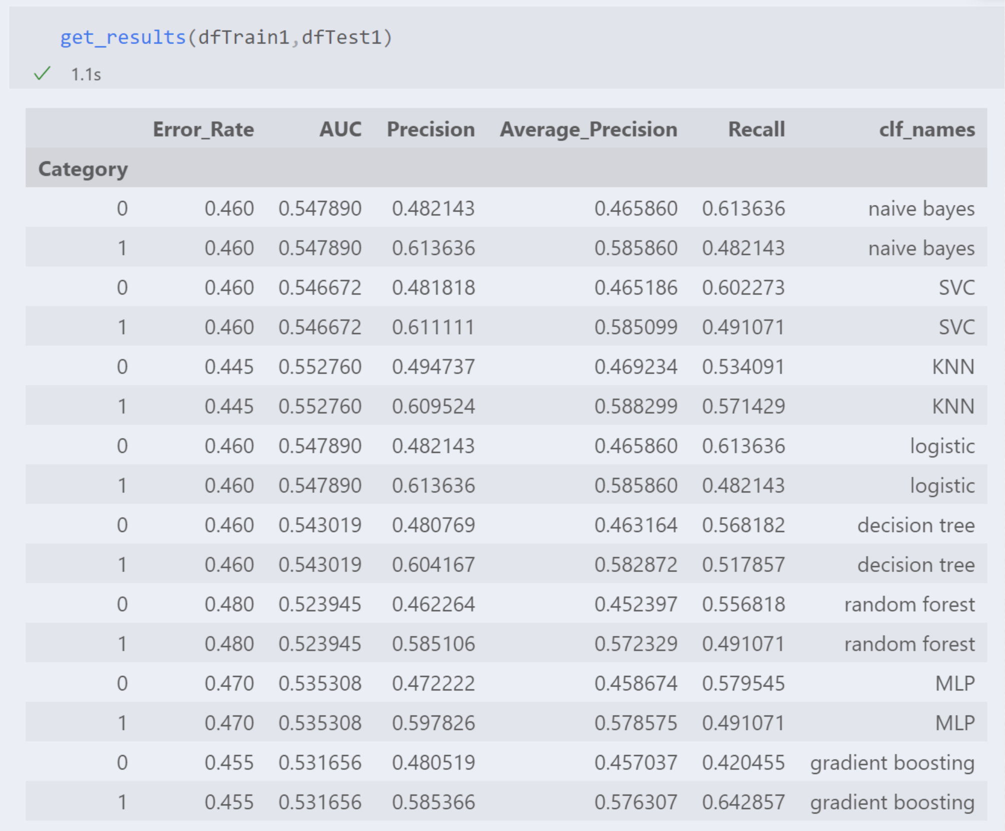
- 10 datasets from 5 different random functions.
- 8 classifiers: Naïve Bayes, Support Vector Machine, K Nearest Neighbor, Logistic Regression, Decision Tree, Random Forest, Neural Network, and Gradient Boosting







Classifying complete mess

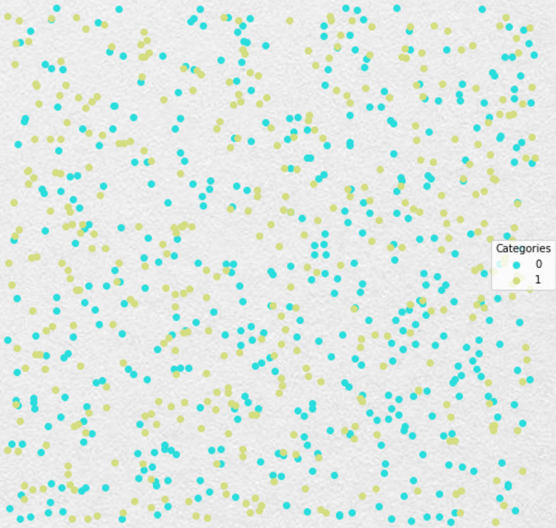


```
get_results(dfTrain1,dfTest1)
```

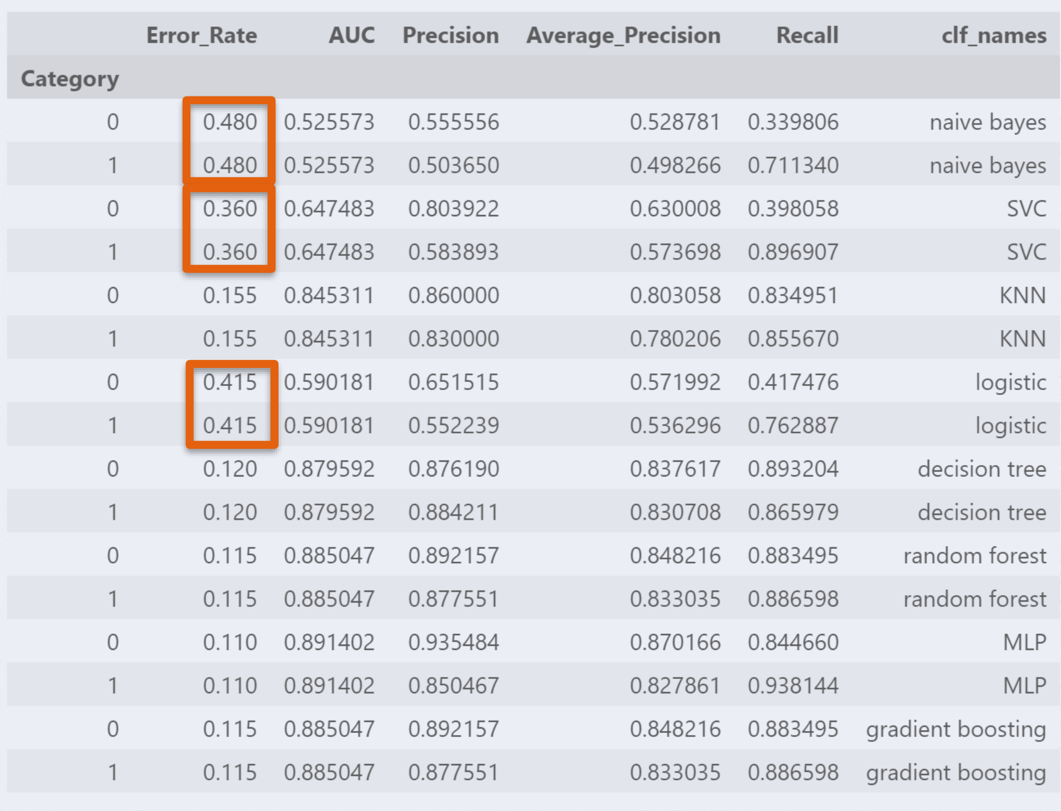
✓ 1.1s

	Error_Rate	AUC	Precision	Average_Precision	Recall	clf_names
Category						
0	0.460	0.547890	0.482143	0.465860	0.613636	naive bayes
1	0.460	0.547890	0.613636	0.585860	0.482143	naive bayes
0	0.460	0.546672	0.481818	0.465186	0.602273	SVC
1	0.460	0.546672	0.611111	0.585099	0.491071	SVC
0	0.445	0.552760	0.494737	0.469234	0.534091	KNN
1	0.445	0.552760	0.609524	0.588299	0.571429	KNN
0	0.460	0.547890	0.482143	0.465860	0.613636	logistic
1	0.460	0.547890	0.613636	0.585860	0.482143	logistic
0	0.460	0.543019	0.480769	0.463164	0.568182	decision tree
1	0.460	0.543019	0.604167	0.582872	0.517857	decision tree
0	0.480	0.523945	0.462264	0.452397	0.556818	random forest
1	0.480	0.523945	0.585106	0.572329	0.491071	random forest
0	0.470	0.535308	0.472222	0.458674	0.579545	MLP
1	0.470	0.535308	0.597826	0.578575	0.491071	MLP
0	0.455	0.531656	0.480519	0.457037	0.420455	gradient boosting
1	0.455	0.531656	0.585366	0.576307	0.642857	gradient boosting

- Each two rows is the performance of the classifier shown on the right column.

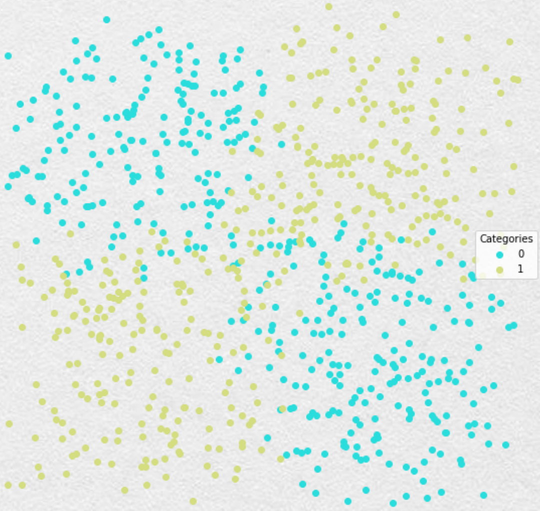


Medium difficulty

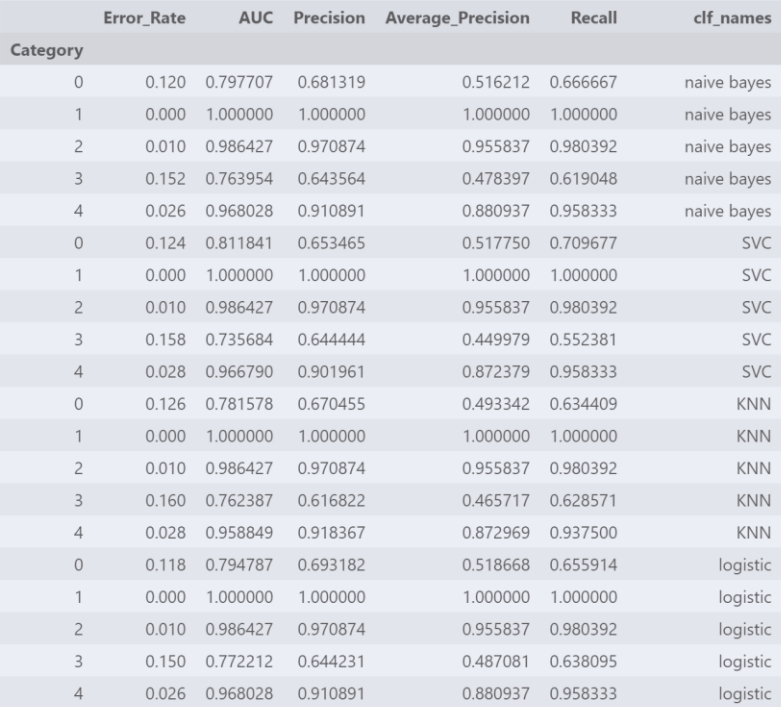


	Error_Rate	AUC	Precision	Average_Precision	Recall	clf_names
Category						
0	0.480	0.525573	0.555556	0.528781	0.339806	naive bayes
1	0.480	0.525573	0.503650	0.498266	0.711340	naive bayes
0	0.360	0.647483	0.803922	0.630008	0.398058	SVC
1	0.360	0.647483	0.583893	0.573698	0.896907	SVC
0	0.155	0.845311	0.860000	0.803058	0.834951	KNN
1	0.155	0.845311	0.830000	0.780206	0.855670	KNN
0	0.415	0.590181	0.651515	0.571992	0.417476	logistic
1	0.415	0.590181	0.552239	0.536296	0.762887	logistic
0	0.120	0.879592	0.876190	0.837617	0.893204	decision tree
1	0.120	0.879592	0.884211	0.830708	0.865979	decision tree
0	0.115	0.885047	0.892157	0.848216	0.883495	random forest
1	0.115	0.885047	0.877551	0.833035	0.886598	random forest
0	0.110	0.891402	0.935484	0.870166	0.844660	MLP
1	0.110	0.891402	0.850467	0.827861	0.938144	MLP
0	0.115	0.885047	0.892157	0.848216	0.883495	gradient boosting
1	0.115	0.885047	0.877551	0.833035	0.886598	gradient boosting

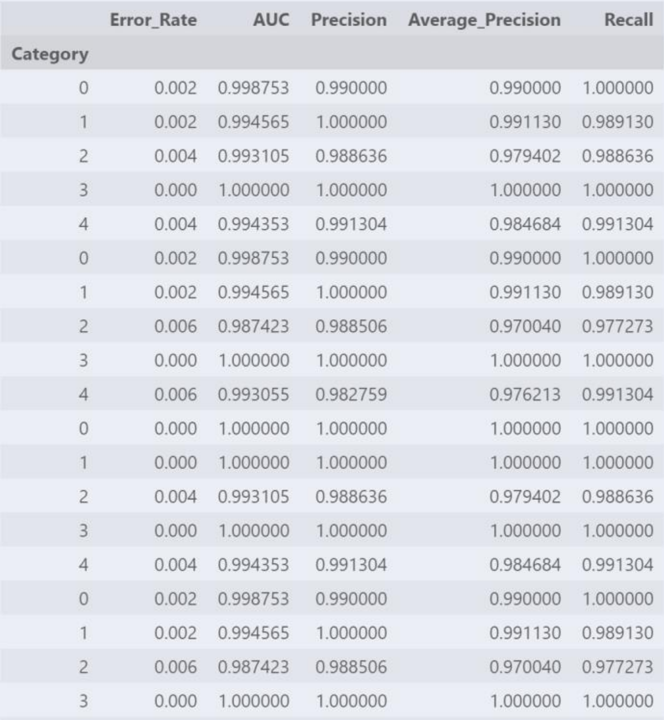
- Each two rows is the performance of the classifier shown on the right column.



Another finding...Lifelong learner



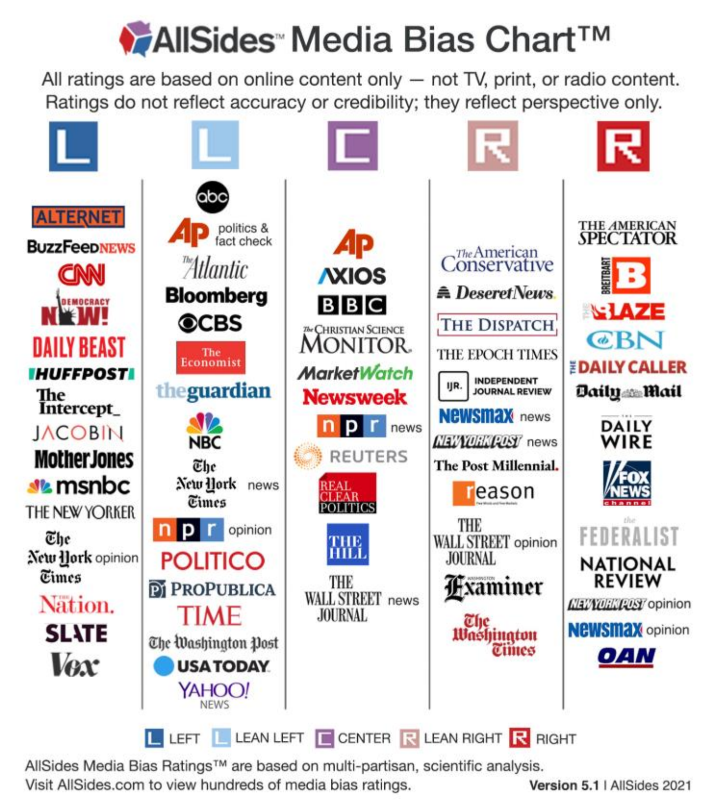
	Error_Rate	AUC	Precision	Average_Precision	Recall	clf_names
Category						
0	0.120	0.797707	0.681319	0.516212	0.666667	naive bayes
1	0.000	1.000000	1.000000	1.000000	1.000000	naive bayes
2	0.010	0.986427	0.970874	0.955837	0.980392	naive bayes
3	0.152	0.763954	0.643564	0.478397	0.619048	naive bayes
4	0.026	0.968028	0.910891	0.880937	0.958333	naive bayes
0	0.124	0.811841	0.653465	0.517750	0.709677	SVC
1	0.000	1.000000	1.000000	1.000000	1.000000	SVC
2	0.010	0.986427	0.970874	0.955837	0.980392	SVC
3	0.158	0.735684	0.644444	0.449979	0.552381	SVC
4	0.028	0.966790	0.901961	0.872379	0.958333	SVC
0	0.126	0.781578	0.670455	0.493342	0.634409	KNN
1	0.000	1.000000	1.000000	1.000000	1.000000	KNN
2	0.010	0.986427	0.970874	0.955837	0.980392	KNN
3	0.160	0.762387	0.616822	0.465717	0.628571	KNN
4	0.028	0.958849	0.918367	0.872969	0.937500	KNN
0	0.118	0.794787	0.693182	0.518668	0.655914	logistic
1	0.000	1.000000	1.000000	1.000000	1.000000	logistic
2	0.010	0.986427	0.970874	0.955837	0.980392	logistic
3	0.150	0.772212	0.644231	0.487081	0.638095	logistic
4	0.026	0.968028	0.910891	0.880937	0.958333	logistic



	Error_Rate	AUC	Precision	Average_Precision	Recall
Category					
0	0.002	0.998753	0.990000	0.990000	1.000000
1	0.002	0.994565	1.000000	0.991130	0.989130
2	0.004	0.993105	0.988636	0.979402	0.988636
3	0.000	1.000000	1.000000	1.000000	1.000000
4	0.004	0.994353	0.991304	0.984684	0.991304
0	0.002	0.998753	0.990000	0.990000	1.000000
1	0.002	0.994565	1.000000	0.991130	0.989130
2	0.006	0.987423	0.988506	0.970040	0.977273
3	0.000	1.000000	1.000000	1.000000	1.000000
4	0.006	0.993055	0.982759	0.976213	0.991304
0	0.000	1.000000	1.000000	1.000000	1.000000
1	0.000	1.000000	1.000000	1.000000	1.000000
2	0.004	0.993105	0.988636	0.979402	0.988636
3	0.000	1.000000	1.000000	1.000000	1.000000
4	0.004	0.994353	0.991304	0.984684	0.991304
0	0.002	0.998753	0.990000	0.990000	1.000000
1	0.002	0.994565	1.000000	0.991130	0.989130
2	0.006	0.987423	0.988506	0.970040	0.977273
3	0.000	1.000000	1.000000	1.000000	1.000000

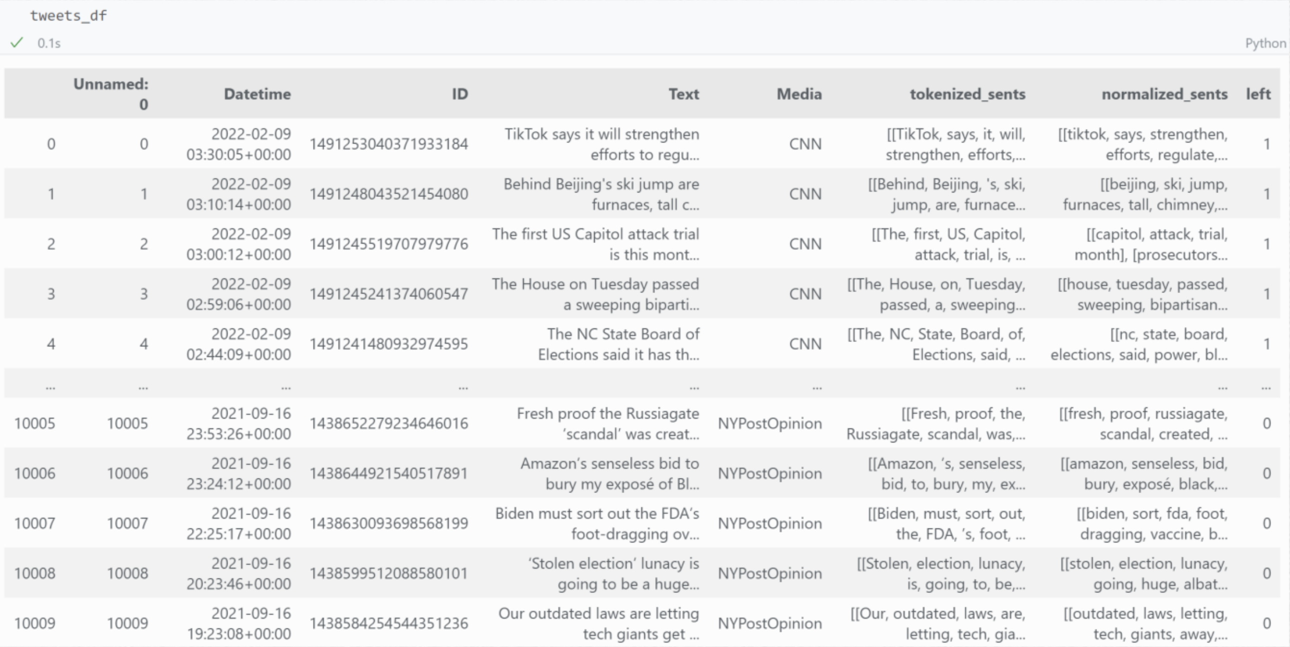
The text

- Tweets from 10 official accounts of news media, with 5 of them being left-skewed and 5 being right-skewed
- 5 Left skewed: CNN, Democracy Now, Daily Beast, Huffpost, and Jacobin
- 5 Right skewed: The American Spectator, Breitbart, Fox News, National Review, and New York Post Opinion.



The data

- Data: 1,000 tweets from each of the 10 media.
- Task: Distinguishing between left- and right- skewed media

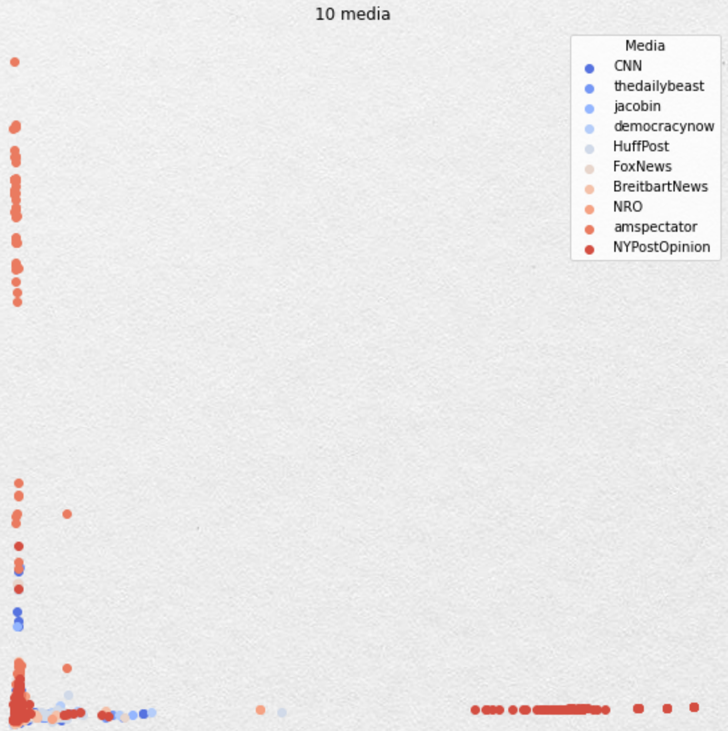


```
tweets_df
```

	Unnamed: 0	Datetime	ID	Text	Media	tokenized_sents	normalized_sents	left
0	0	2022-02-09 03:30:05+00:00	1491253040371933184	TikTok says it will strengthen efforts to regu...	CNN	[[TikTok, says, it, will, strengthen, efforts,...	[[tiktok, says, strengthen, efforts, regulate,...	1
1	1	2022-02-09 03:10:14+00:00	1491248043521454080	Behind Beijing's ski jump are furnaces, tall c...	CNN	[[Behind, Beijing, 's, ski, jump, are, furnace...	[[beijing, ski, jump, furnaces, tall, chimney,...	1
2	2	2022-02-09 03:00:12+00:00	1491245519707979776	The first US Capitol attack trial is this mont...	CNN	[[The, first, US, Capitol, attack, trial, is, ...	[[capitol, attack, trial, month], [prosecutors...	1
3	3	2022-02-09 02:59:06+00:00	1491245241374060547	The House on Tuesday passed a sweeping biparti...	CNN	[[The, House, on, Tuesday, passed, a, sweeping...	[[house, tuesday, passed, sweeping, bipartisan...	1
4	4	2022-02-09 02:44:09+00:00	1491241480932974595	The NC State Board of Elections said it has th...	CNN	[[The, NC, State, Board, of, Elections, said, ...	[[nc, state, board, elections, said, power, bl...	1
...	...	...	...	...	...	...	...	...
10005	10005	2021-09-16 23:53:26+00:00	1438652279234646016	Fresh proof the Russiagate ‘scandal’ was creat...	NYPostOpinion	[[Fresh, proof, the, Russiagate, scandal, was,...	[[fresh, proof, russiagate, scandal, created, ...	0
10006	10006	2021-09-16 23:24:12+00:00	1438644921540517891	Amazon’s senseless bid to bury my exposé of Bl...	NYPostOpinion	[[Amazon, ’s, senseless, bid, to, bury, my, ex...	[[amazon, senseless, bid, bury, exposé, black,...	0
10007	10007	2021-09-16 22:25:17+00:00	1438630093698568199	Biden must sort out the FDA’s foot-dragging ov...	NYPostOpinion	[[Biden, must, sort, out, the, FDA, ’s, foot, ...	[[biden, sort, fda, foot, dragging, vaccine, b...	0
10008	10008	2021-09-16 20:23:46+00:00	1438599512088580101	‘Stolen election’ lunacy is going to be a huge...	NYPostOpinion	[[Stolen, election, lunacy, is, going, to, be,...	[[stolen, election, lunacy, going, huge, albat...	0
10009	10009	2021-09-16 19:23:08+00:00	1438584254544351236	Our outdated laws are letting tech giants get ...	NYPostOpinion	[[Our, outdated, laws, are, letting, tech, gia...	[[outdated, laws, letting, tech, giants, away,...	0

What PCA tells

- TF-IDF + 2 dimentional PCA: failed to tell the two parties apart.



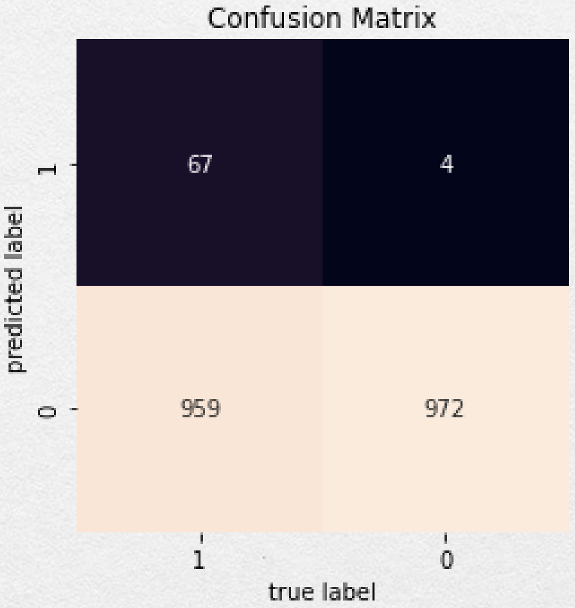
Logistic regression

	1	2	3	4
	10 PCA components	**100 PCA components**	**200 PCA components**	**with L1 regularization**
Train Accuracy	60.4%	73.3%	75.5%	93.5%
Test Accuracy	60.7%	71.9%	73.1%	83.8%

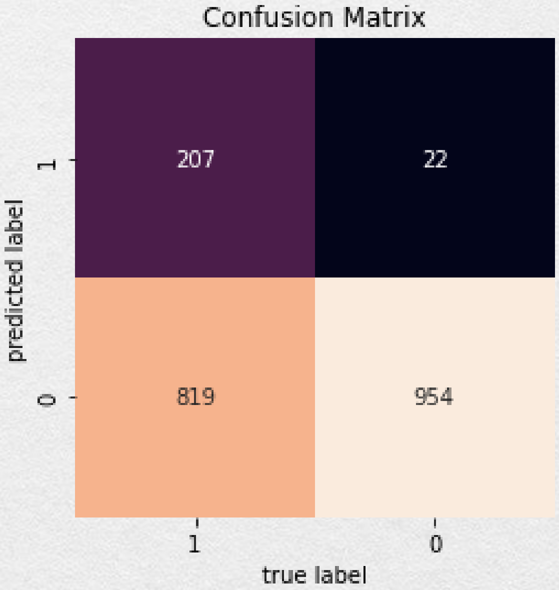
- The accuracy increases as more PCA components are added
- The logistic regression with L1 regularization did an excellent job, though it shows a bit of overfitting.

Decision Tree & Random Forest

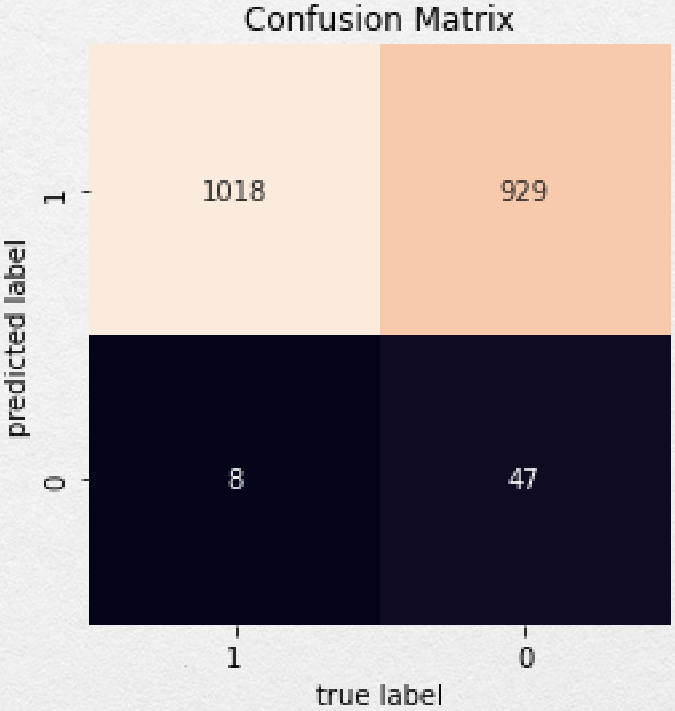
Decision Tree



Random Forest



K Nearest Neighbor



Test Accuracy

Logistic regression with 10 components: 61%
Decision tree: 52%
Random forest: 58%
KNN: 53%

Neural Network

	1 Logistic regression with L1 regularization	2 Neural network
Test Accuracy	83.8%	82.4%
Precision for the left	79.8%	81.6%
Precision for the right	88.6%	83.1%

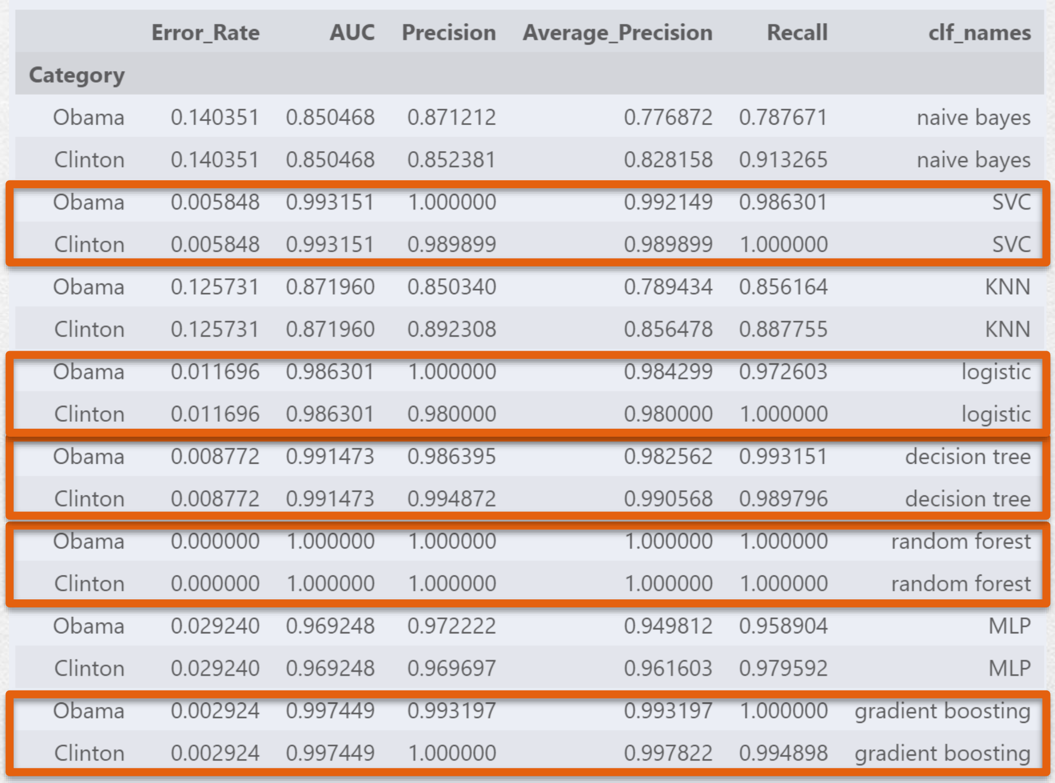
Neural Network

	1	2
	Logistic regression with L1 regularization	**Neural network**
Test Accuracy	83.8%	82.4%
Precision for the left	79.8%	81.6%
Precision for the right	88.6%	83.1%

Whoa!

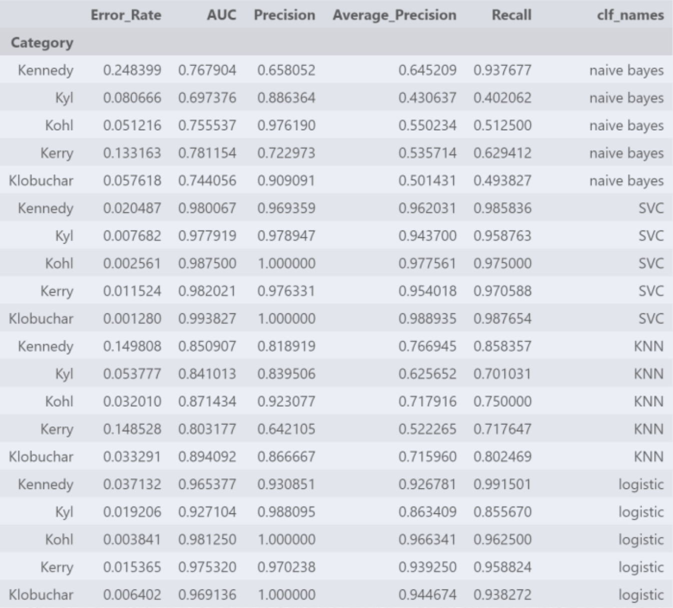
A classifier as simple as logistic regression can do a very good job on certain task.

Senate press releases – small

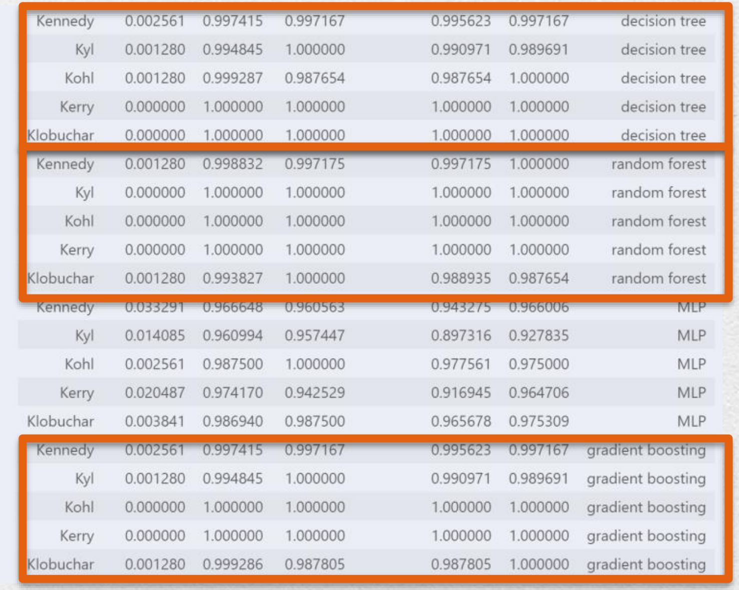


	Error_Rate	AUC	Precision	Average_Precision	Recall	clf_names
Category						
Obama	0.140351	0.850468	0.871212	0.776872	0.787671	naive bayes
Clinton	0.140351	0.850468	0.852381	0.828158	0.913265	naive bayes
Obama	0.005848	0.993151	1.000000	0.992149	0.986301	SVC
Clinton	0.005848	0.993151	0.989899	0.989899	1.000000	SVC
Obama	0.125731	0.871960	0.850340	0.789434	0.856164	KNN
Clinton	0.125731	0.871960	0.892308	0.856478	0.887755	KNN
Obama	0.011696	0.986301	1.000000	0.984299	0.972603	logistic
Clinton	0.011696	0.986301	0.980000	0.980000	1.000000	logistic
Obama	0.008772	0.991473	0.986395	0.982562	0.993151	decision tree
Clinton	0.008772	0.991473	0.994872	0.990568	0.989796	decision tree
Obama	0.000000	1.000000	1.000000	1.000000	1.000000	random forest
Clinton	0.000000	1.000000	1.000000	1.000000	1.000000	random forest
Obama	0.029240	0.969248	0.972222	0.949812	0.958904	MLP
Clinton	0.029240	0.969248	0.969697	0.961603	0.979592	MLP
Obama	0.002924	0.997449	0.993197	0.993197	1.000000	gradient boosting
Clinton	0.002924	0.997449	1.000000	0.997822	0.994898	gradient boosting

Senate press releases – large



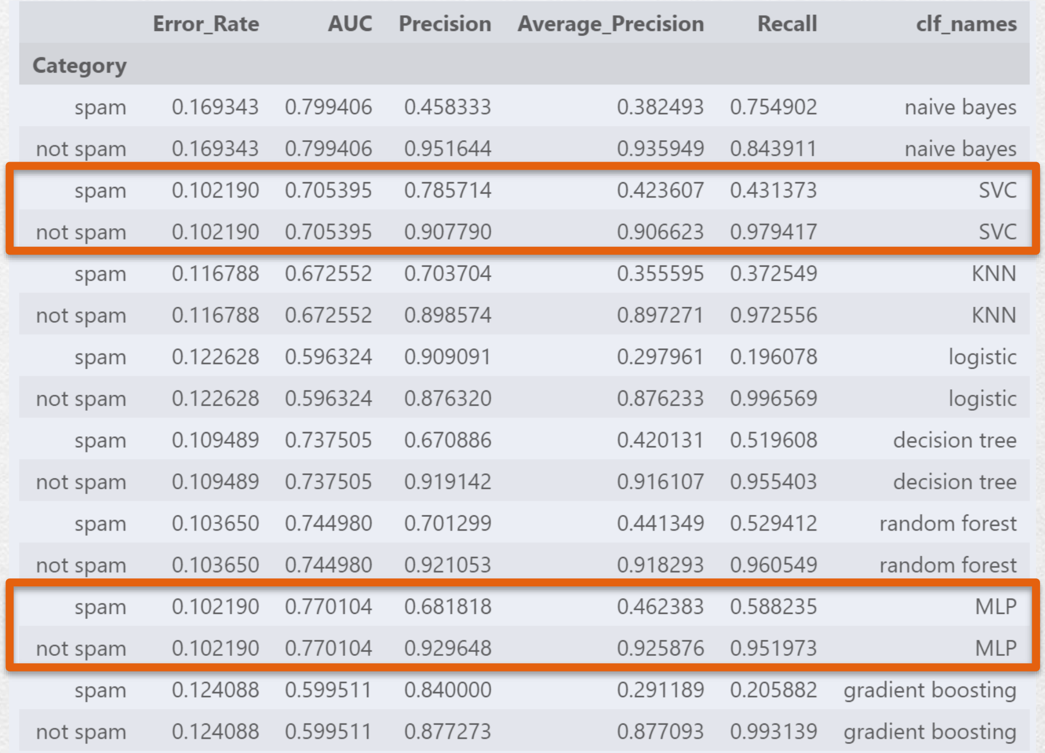
	Error_Rate	AUC	Precision	Average_Precision	Recall	clf_names
Category						
Kennedy	0.248399	0.767904	0.658052	0.645209	0.937677	naive bayes
Kyl	0.080666	0.697376	0.886364	0.430637	0.402062	naive bayes
Kohl	0.051216	0.755537	0.976190	0.550234	0.512500	naive bayes
Kerry	0.133163	0.781154	0.722973	0.535714	0.629412	naive bayes
Klobuchar	0.057618	0.744056	0.909091	0.501431	0.493827	naive bayes
Kennedy	0.020487	0.980067	0.969359	0.962031	0.985836	SVC
Kyl	0.007682	0.977919	0.978947	0.943700	0.958763	SVC
Kohl	0.002561	0.987500	1.000000	0.977561	0.975000	SVC
Kerry	0.011524	0.982021	0.976331	0.954018	0.970588	SVC
Klobuchar	0.001280	0.993827	1.000000	0.988935	0.987654	SVC
Kennedy	0.149808	0.850907	0.818919	0.766945	0.858357	KNN
Kyl	0.053777	0.841013	0.839506	0.625652	0.701031	KNN
Kohl	0.032010	0.871434	0.923077	0.717916	0.750000	KNN
Kerry	0.148528	0.803177	0.642105	0.522265	0.717647	KNN
Klobuchar	0.033291	0.894092	0.866667	0.715960	0.802469	KNN
Kennedy	0.037132	0.965377	0.930851	0.926781	0.991501	logistic
Kyl	0.019206	0.927104	0.988095	0.863409	0.855670	logistic
Kohl	0.003841	0.981250	1.000000	0.966341	0.962500	logistic
Kerry	0.015365	0.975320	0.970238	0.939250	0.958824	logistic
Klobuchar	0.006402	0.969136	1.000000	0.944674	0.938272	logistic



Kennedy	0.002561	0.997415	0.997167	0.995623	0.997167	decision tree
Kyl	0.001280	0.994845	1.000000	0.990971	0.989691	decision tree
Kohl	0.001280	0.999287	0.987654	0.987654	1.000000	decision tree
Kerry	0.000000	1.000000	1.000000	1.000000	1.000000	decision tree
Klobuchar	0.000000	1.000000	1.000000	1.000000	1.000000	decision tree
Kennedy	0.001280	0.998832	0.997175	0.997175	1.000000	random forest
Kyl	0.000000	1.000000	1.000000	1.000000	1.000000	random forest
Kohl	0.000000	1.000000	1.000000	1.000000	1.000000	random forest
Kerry	0.000000	1.000000	1.000000	1.000000	1.000000	random forest
Klobuchar	0.001280	0.993827	1.000000	0.988935	0.987654	random forest
Kennedy	0.033291	0.966648	0.960563	0.943275	0.966006	MLP
Kyl	0.014085	0.960994	0.957447	0.897316	0.927835	MLP
Kohl	0.002561	0.987500	1.000000	0.977561	0.975000	MLP
Kerry	0.020487	0.974170	0.942529	0.916945	0.964706	MLP
Klobuchar	0.003841	0.986940	0.987500	0.965678	0.975309	MLP
Kennedy	0.002561	0.997415	0.997167	0.995623	0.997167	gradient boosting
Kyl	0.001280	0.994845	1.000000	0.990971	0.989691	gradient boosting
Kohl	0.000000	1.000000	1.000000	1.000000	1.000000	gradient boosting
Kerry	0.000000	1.000000	1.000000	1.000000	1.000000	gradient boosting
Klobuchar	0.001280	0.999286	0.987805	0.987805	1.000000	gradient boosting

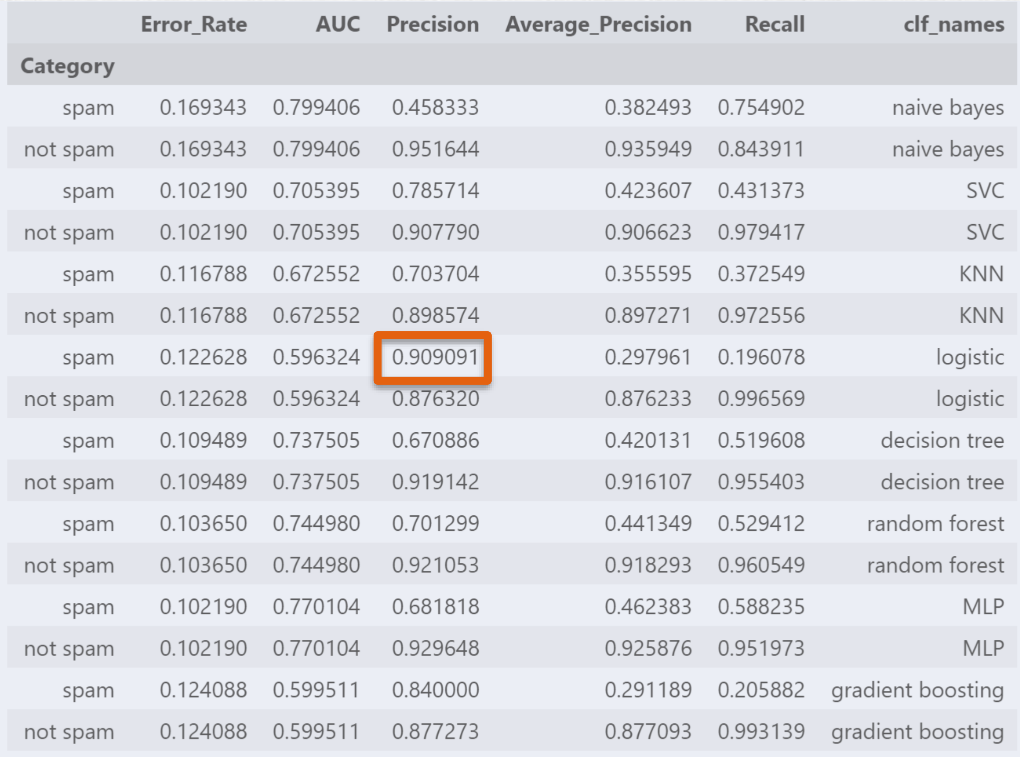
- Different classifiers seem to be good at identifying different senators

Spam



	Error_Rate	AUC	Precision	Average_Precision	Recall	clf_names
Category						
spam	0.169343	0.799406	0.458333	0.382493	0.754902	naive bayes
not spam	0.169343	0.799406	0.951644	0.935949	0.843911	naive bayes
spam	0.102190	0.705395	0.785714	0.423607	0.431373	SVC
not spam	0.102190	0.705395	0.907790	0.906623	0.979417	SVC
spam	0.116788	0.672552	0.703704	0.355595	0.372549	KNN
not spam	0.116788	0.672552	0.898574	0.897271	0.972556	KNN
spam	0.122628	0.596324	0.909091	0.297961	0.196078	logistic
not spam	0.122628	0.596324	0.876320	0.876233	0.996569	logistic
spam	0.109489	0.737505	0.670886	0.420131	0.519608	decision tree
not spam	0.109489	0.737505	0.919142	0.916107	0.955403	decision tree
spam	0.103650	0.744980	0.701299	0.441349	0.529412	random forest
not spam	0.103650	0.744980	0.921053	0.918293	0.960549	random forest
spam	0.102190	0.770104	0.681818	0.462383	0.588235	MLP
not spam	0.102190	0.770104	0.929648	0.925876	0.951973	MLP
spam	0.124088	0.599511	0.840000	0.291189	0.205882	gradient boosting
not spam	0.124088	0.599511	0.877273	0.877093	0.993139	gradient boosting

Spam

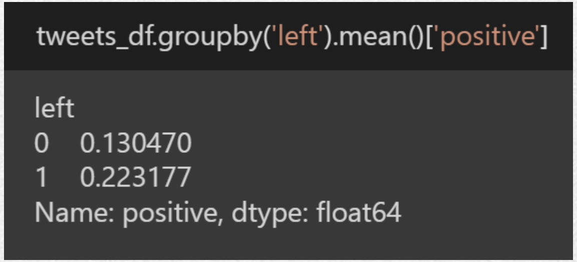


	Error_Rate	AUC	Precision	Average_Precision	Recall	clf_names
Category						
spam	0.169343	0.799406	0.458333	0.382493	0.754902	naive bayes
not spam	0.169343	0.799406	0.951644	0.935949	0.843911	naive bayes
spam	0.102190	0.705395	0.785714	0.423607	0.431373	SVC
not spam	0.102190	0.705395	0.907790	0.906623	0.979417	SVC
spam	0.116788	0.672552	0.703704	0.355595	0.372549	KNN
not spam	0.116788	0.672552	0.898574	0.897271	0.972556	KNN
spam	0.122628	0.596324	0.909091	0.297961	0.196078	logistic
not spam	0.122628	0.596324	0.876320	0.876233	0.996569	logistic
spam	0.109489	0.737505	0.670886	0.420131	0.519608	decision tree
not spam	0.109489	0.737505	0.919142	0.916107	0.955403	decision tree
spam	0.103650	0.744980	0.701299	0.441349	0.529412	random forest
not spam	0.103650	0.744980	0.921053	0.918293	0.960549	random forest
spam	0.102190	0.770104	0.681818	0.462383	0.588235	MLP
not spam	0.102190	0.770104	0.929648	0.925876	0.951973	MLP
spam	0.124088	0.599511	0.840000	0.291189	0.205882	gradient boosting
not spam	0.124088	0.599511	0.877273	0.877093	0.993139	gradient boosting

- BERT
- Sentiment analysis pipeline
- Applied on my media tweets dataset

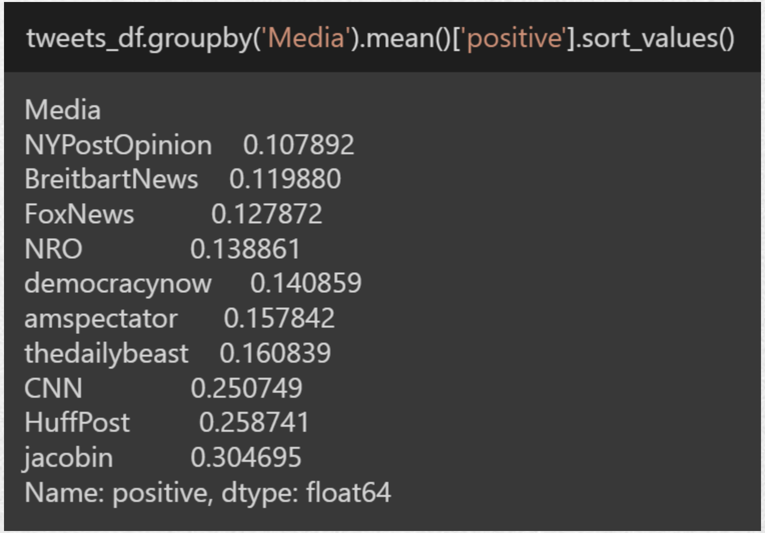
BERT - The “sentiment-analysis” pipeline

- Calculated the average proportion of tweets with positive sentiment by skewness group, and by media



```
tweets_df.groupby('left').mean()['positive']

left
0    0.130470
1    0.223177
Name: positive, dtype: float64
```



```
tweets_df.groupby('Media').mean()['positive'].sort_values()

Media
NYPostOpinion    0.107892
BreitbartNews    0.119880
FoxNews          0.127872
NRO              0.138861
democracynow     0.140859
amspectator      0.157842
thedailybeast    0.160839
CNN              0.250749
HuffPost         0.258741
jacobin          0.304695
Name: positive, dtype: float64
```

Takeaways

1. Different classifiers are good at different tasks.
2. The simple logistic regression classifier performs well on various tasks
3. Training for multiple times seems to boost the performance
4. But…How to open the black box?