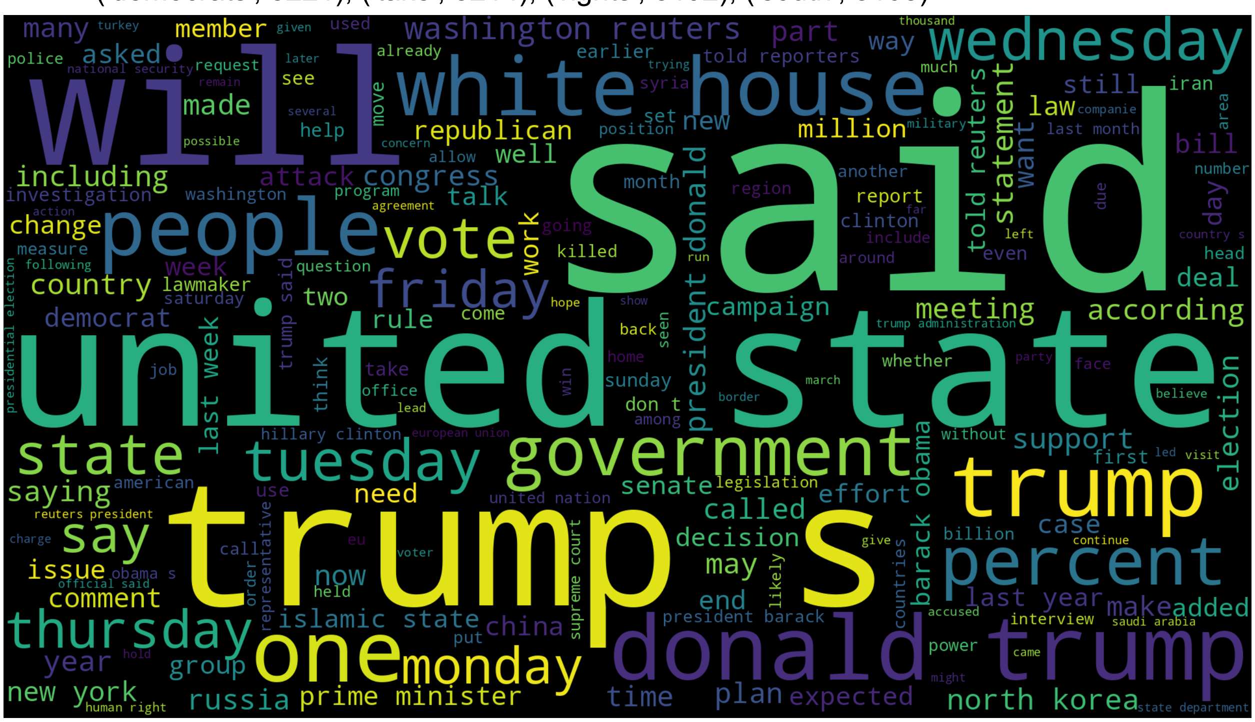
Task 1:

What are the most commonly used words (top 100) in the collection, the most commonly used words (top 100) in the real news and most commonly used words (top 100) in the fake news?

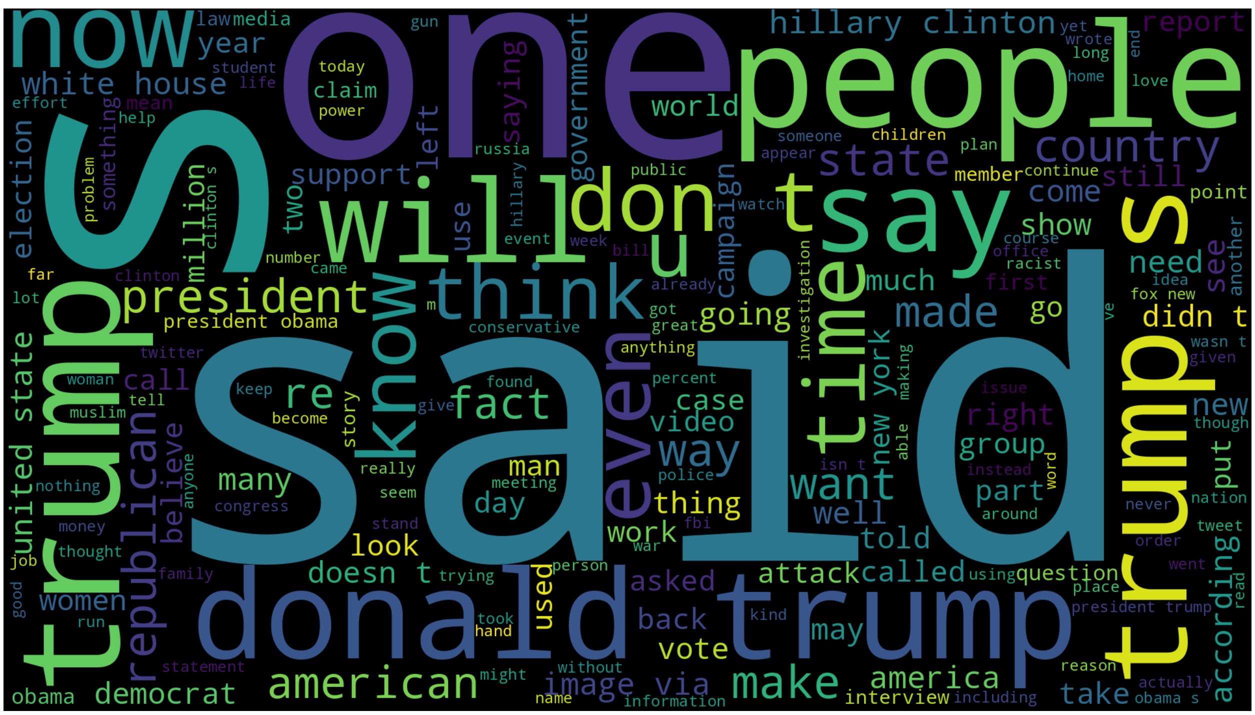
True:

- ('said', 99024), ('trump', 54238), ('would', 31524), ('reuters', 28412), ('president', 26386), ('state', 19726), ('government', 18285), ('new', 16743), ('house', 16513), ('states', 16506), ('also', 15946), ('united', 15575), ('republican', 15346), ('people', 15126), ('told', 14244), ('could', 13709), ('one', 12670), ('last', 12612), ('party', 12434), ('washington', 12238), ('two', 11619), ('election', 11474), ('year', 10968), ('former', 10601), ('campaign', 10554), ('donald', 10447), ('security', 10071), ('percent', 9936), ('north', 9868), ('clinton', 9453), ('white', 9443), ('court', 9404), ('senate', 9204), ('obama', 9197), ('country', 8685), ('minister', 8660), ('china', 8561), ('first', 8546), ('officials', 8474), ('since', 8332), ('tuesday', 8263), ('democratic', 8237), ('week', 8217), ('foreign', 8196), ('administration', 8191), ('national', 8184), ('including', 8119), ('presidential', 8011), ('wednesday', 8008), ('military', 7996), ('russia', 7821), ('may', 7811), ('law', 7796), ('tax', 7776), ('years', 7742), ('political', 7698), ('thursday', 7661), ('statement', 7562), ('friday', 7331), ('korea', 7235), ('support', 7135), ('monday', 7092), ('group', 7010), ('bill', 6898), ('vote', 6877), ('time', 6791), ('republicans', 6749), ('many', 6721), ('say', 6681), ('office', 6522), ('congress', 6473), ('federal', 6446), ('million', 6394), ('committee', 6385), ('made', 6201), ('department', 6163), ('according', 6142), ('official', 6101), ('police', 6054), ('called', 6047), ('saying', 5982), ('news', 5973), ('deal', 5818), ('trade', 5812), ('policy', 5678), ('leader', 5648), ('russian', 5551), ('iran', 5479), ('make', 5389), ('next', 5386), ('secretary', 5379), ('city', 5365), ('public', 5356), ('month', 5274), ('whether', 5270), ('countries', 5239), ('democrats', 5221), ('take', 5214), ('rights', 5192), ('south', 5108)



Fake:

- ('trump', 74038), ('said', 31125), ('people', 25997), ('president', 25591), ('would', 23457), ('one', 22881), ('clinton', 18074), ('obama', 17865), ('like', 17643), ('donald', 16188), ('also', 15243), ('new', 14173), ('us', 13914), ('even', 13666), ('hillary', 13549), ('news', 13406), ('time', 12782), ('white', 12762), ('state', 12532), ('via', 11348), ('media', 11050), ('get', 10705), ('america', 10613), ('campaign', 10561), ('house', 10559), ('know', 10287), ('could', 10223), ('first', 10016), ('american', 9937), ('going', 9745), ('many', 9695), ('image', 9622), ('states', 9522), ('make', 9143), ('told', 9103), ('republican', 8928), ('right', 8896), ('country', 8684), ('made', 8666), ('government', 8600), ('police', 8564), ('say', 8551), ('way', 8457), ('back', 8397), ('think', 8358), ('two', 8308), ('years', 8259), ('video', 8068), ('election', 8019), ('united', 7974), ('last', 7771), ('may', 7628), ('political', 7546), ('party', 7459), ('black', 7454), ('see', 7305), ('want', 7285), ('women', 7165), ('national', 7075), ('well', 7013), ('former', 7010), ('says', 6968), ('world', 6962), ('republicans', 6944), ('much', 6931), ('law', 6880), ('take', 6698), ('go', 6665), ('year', 6579), ('public', 6535), ('man', 6326), ('according', 6278), ('never', 6184), ('another', 6180), ('really', 6179), ('bill', 6145), ('americans', 6120), ('day', 6095), ('fact', 6012), ('every', 6002), ('since', 5974), ('show', 5888), ('support', 5846), ('2016', 5733), ('still', 5723), ('presidential', 5689), ('security', 5603), ('fbi', 5568), ('called', 5547), ('story', 5546), ('washington', 5527), ('group', 5503), ('administration', 5502), ('vote', 5476), ('department', 5438), ('office', 5332), ('good', 5313), ('federal', 5312), ('russia', 5311), ('saying', 5109)



By reading the preprocessed textual data, can you easily tell the difference between the real news and fake news? What does the strongest feature set (for machine learning) look like?

There are some words that show up on the fake dataset a lot more than the true dataset. For example "people" is the third most common word in the fake dataset, but the 14 most common words in the true dataset. If it has the words clinton, obama, hillary it is most likely fake news. Comparing the true vs fake news it is pretty easy to get a good feel for which words are most likely to be in true news or in fake news.

Task 2:

Please report the performance of different algorithms in the following table:

ML Model	Filter	Precision	Recall	Accuracy
SVC	Unfiltered	0.9944060158802738	0.9944409942517028	0.9944320712694877
LinearSVC	Unfiltered	0.9965379423818262	0.996546798690362	0.9965478841870824
SGDClassifier	Unfiltered	0.9942358279572523	0.9939588229156833	0.994097995545657
Perceptron	Unfiltered	0.9952172524226757	0.9951904589197971	0.9952115812917595
PassiveAggressiveClassifier	Unfiltered	0.9946127560763967	0.994681955712774	0.9946547884187082
MLPClassifier	Unfiltered	0.9948605990655202	0.9948782026759142	0.9948775055679288
MultinomialNB	Unfiltered	0.9559787252174787	0.9561614094009747	0.9561247216035634
RandomForestClassifier	Unfiltered	0.9876150750551755	0.987623772214411	0.9876391982182628
KNeighborsClassifier	Unfiltered	0.899235056898619	0.8995476383292677	0.8983296213808464
DecisionTreeClassifier	Unfiltered	0.9963165709177896	0.9961093418522733	0.9962138084632517
AdaBoostClassifier	Unfiltered	0.996011925962202	0.9959578093870636	0.9959910913140312
GradientBoostingClassifier	Unfiltered	0.9959431131775552	0.9960293525837527	0.9959910913140312
BaggingClassifier	Unfiltered	0.9976995811895611	0.9976174625019252	0.9976614699331848

SVC	Stop words removed	0.9939466856276766	0.9937894046513014	0.9938752783964365
LinearSVC	Stop words removed	0.9962978418463472	0.9963416088276356	0.9963251670378619
SGDClassifier	Stop words removed	0.9951484940096116	0.9950386780407101	0.9951002227171493
Perceptron	Stop words removed	0.994035900974811	0.9941466238069923	0.994097995545657
PassiveAggressiveClassifier	Stop words removed	0.9948781178056754	0.994860316876742	0.9948775055679288
MLPClassifier	Stop words removed	0.9946127560763967	0.994681955712774	0.9946547884187082
MultinomialNB	Stop words removed	0.9551148627813753	0.955224392255449	0.9552338530066815
RandomForestClassifier	Stop words removed	0.9899872866021229	0.9901759763907452	0.9900890868596882
KNeighborsClassifier	Stop words removed	0.8235852762322429	0.8233280503982074	0.821826280623608
DecisionTreeClassifier	Stop words removed	0.9972631910790997	0.9971623682785415	0.9972160356347439
AdaBoostClassifier	Stop words removed	0.9950836667922602	0.995101278337813	0.9951002227171493
GradientBoostingClassifier	Stop words removed	0.9959354011518847	0.9960382954833389	0.9959910913140312
BaggingClassifier	Stop words removed	0.9971089546686001	0.9970910734957297	0.9971046770601336
LogisticRegression	Stop words removed	0.9968682086061814	0.9968858836330032	0.9968819599109131
SVC	TF-IDF	0.992652322621699	0.9925842902516626	0.9926503340757238
LinearSVC	TF-IDF	0.9943536885065074	0.9941658959162964	0.9942835931700075
SGDClassifier	TF-IDF	0.98951577935	0.98961050715	0.98960653303

		97193	95371	63771
Perceptron	TF-IDF	0.9911291700440843	0.9912801901460517	0.9912397921306607
PassiveAggressiveClassifier	TF-IDF	0.993559409972769	0.9933212680539414	0.9934669636228656
MLPClassifier	TF-IDF	0.991788959214907	0.9918081669729926	0.991833704528582
MultinomialNB	TF-IDF	0.9400419928334367	0.9391462978265337	0.9398663697104677
RandomForestClassifier	TF-IDF	0.986490901349717	0.9866770319518974	0.9866369710467706
KNeighborsClassifier	TF-IDF	0.7923585538784901	0.6967757522822886	0.7147735708982925
DecisionTreeClassifier	TF-IDF	0.9969984312698743	0.9970371656850665	0.9970304380103935
AdaBoostClassifier	TF-IDF	0.9966630013176441	0.9967780704561451	0.9967334818114328
GradientBoostingClassifier	TF-IDF	0.9959014398353867	0.9962097857447187	0.9960653303637713
BaggingClassifier	TF-IDF	0.9972593445647596	0.9973745518667709	0.9973273942093541
LogisticRegression	TF-IDF	0.9858243780667695	0.9858433528667342	0.985894580549369

Please provide error analysis for the best performed results (top 2) by using "Test Confusion Matrix". Please explain this outcome.

The best model is the BaggingClassifier unfiltered with an accuracy of 0.997661.
Confusion matrix:

Confusion matrix	True Positive	True Negative
Predicted Positive	4293	17
Predicted Negative	9	4661

The second best result BaggingClassifier TF-IDF with an accuracy of 0.997327.
Confusion matrix:

Confusion matrix	True Positive	True Negative
Predicted Positive	6286	7
Predicted Negative	22	7155

The third best DecisionTreeClassifier Stop words removed with an accuracy of 0.997216.
Confusion matrix:

Confusion matrix	True Positive	True Negative
Predicted Positive	4290	20
Predicted Negative	8	4662

It looks like a bagging classifier is the best model for this dataset. It got the top two spots. The second best model is the DecisionTreeClassifier.

- True Positive
    - Data that is labeled positive in the dataset
- True Negative
    - Data that is labeled negative in the dataset
- Predicted Positive
    - Data that the model predicts to be positive
- Predicted Negative
    - Data that the model predicts to be negative

Part 3:

Did you witness the performance improvement (compared with the result from task 2)? Why?

The models seemed to perform worse after applying POS tagging which is quite interesting. I was not expecting the result to be the models getting worse after removing the tags. I was expecting that the models would get better because there would be less unneeded words. It seems though that the model likes all of the data it can get and is able to filter though the data it doesn't need.

Please report the performance of different algorithms in the following table:

ML Model	Feature	Filter	Precision	Recall	Accuracy
SVC	Nouns	Stop Words	0.9428602423488577	0.9417228141142735	0.9425389755011135
LinearSVC	Nouns	Stop Words	0.9482611284667446	0.9466864184526107	0.9476614699331849
SGDClassifier	Nouns	Stop Words	0.9474452386251494	0.9454859940784496	0.9466221232368226
Perceptron	Nouns	Stop Words	0.9460519775065843	0.946146817544728	0.946325167037862
PassiveAggressiveClassifier	Nouns	Stop Words	0.942821052232125	0.9423244322616204	0.9428359317000743
MLPClassifier	Nouns	Stop Words	0.953798793683873	0.9532798849973212	0.9537490720118782
MultinomialNB	Nouns	Stop Words	0.9177836130165402	0.9187189461293814	0.9184112843355605
RandomForestClassifier	Nouns	Stop Words	0.9481744968603016	0.9495223288742104	0.9487750556792873
KNeighborsClassifier	Nouns	Stop Words	0.7425763595964823	0.743300615936801	0.7410541945063103
DecisionTreeClassifier	Nouns	Stop Words	0.9004717255838333	0.8987514430742962	0.9000742390497402
AdaBoostClassifier	Nouns	Stop Words	0.872274038248214	0.8715251110813781	0.8725315515961396
GradientBoostingClassifier	Nouns	Stop Words	0.8953099366720174	0.8969408872973172	0.8956941351150706
BaggingClassifier	Nouns	Stop Words	0.9197294653600421	0.921146829168809	0.9203414996288047
LogisticRegression	Nouns	Stop Words	0.9539879799547938	0.9527087883653622	0.9535263548626578

SVC	Nouns and Verbs	Stop Words	0.9471685654758555	0.9462050486549709	0.9469190794357832
LinearSVC	Nouns and Verbs	Stop Words	0.9496826064319757	0.9474856016937592	0.9487008166295472
SGDClassifier	Nouns and Verbs	Stop Words	0.9516027855788249	0.9488507448301458	0.9502598366740905
Perceptron	Nouns and Verbs	Stop Words	0.9492830997305404	0.9476214484530352	0.948626577579807
PassiveAggressiveClassifier	Nouns and Verbs	Stop Words	0.9456413695372794	0.9438249350529386	0.9449146250927988
MLPClassifier	Nouns and Verbs	Stop Words	0.9520553026360234	0.9506320522053638	0.9515219005196733
MLPClassifier	Nouns and Verbs	Stop Words	0.9551069009674772	0.9539604694012012	0.9547141796585004
MultinomialNB	Nouns and Verbs	Stop Words	0.9208257911533219	0.9218297383888772	0.9214550853749072
RandomForestClassifier	Nouns and Verbs	Stop Words	0.9541075511901049	0.9550358075938725	0.9546399406087602
KNeighborsClassifier	Nouns and Verbs	Stop Words	0.7343593628261785	0.7350383299087315	0.7352635486265776
DecisionTreeClassifier	Nouns and Verbs	Stop Words	0.8967814690287522	0.8945893467932957	0.8961395694135115
AdaBoostClassifier	Nouns and Verbs	Stop Words	0.8774737248265938	0.8748099213678528	0.8766889383815887
GradientBoostingClassifier	Nouns and Verbs	Stop Words	0.8797731878647508	0.8796597979123637	0.8757980697847068
BaggingClassifier	Nouns and Verbs	Stop Words	0.917621776504298	0.918873203218759	0.9182628062360801
LogisticRegression	Nouns and Verbs	Stop Words	0.9543824492470058	0.9528101392806234	0.9537490720118782
SVC	Nouns and	Stop Words	0.95380903	0.95327009	0.95374907

	Adjectives		43103969	86282234	20118782
LinearSVC	Nouns and Adjectives	Stop Words	0.9529939094967315	0.951497415214009	0.9524127691165553
SGDClassifier	Nouns and Adjectives	Stop Words	0.9567087094863007	0.9555017671813493	0.9562731997030438
Perceptron	Nouns and Adjectives	Stop Words	0.9547754231640677	0.9532966458149843	0.9541945063103192
PassiveAggressiveClassifier	Nouns and Adjectives	Stop Words	0.9561218845795015	0.9546387289888925	0.9555308092056422
MLPClassifier	Nouns and Adjectives	Stop Words	0.9606148126454077	0.9597640519695168	0.9603563474387528
MultinomialNB	Nouns and Adjectives	Stop Words	0.9276892176890567	0.9285816245509977	0.9282850779510022
RandomForestClassifier	Nouns and Adjectives	Stop Words	0.9596451915917832	0.9607892959086046	0.9602078693392725
KNeighbors Classifier	Nouns and Adjectives	Stop Words	0.7577254168348495	0.7587673634155544	0.7582034149962881
DecisionTreeClassifier	Nouns and Adjectives	Stop Words	0.9058462440317684	0.9046822484408842	0.9057164068299925
AdaBoostClassifier	Nouns and Adjectives	Stop Words	0.8856267000561289	0.8844324019949934	0.8855976243504083
GradientBoostingClassifier	Nouns and Adjectives	Stop Words	0.8894425760405869	0.8898754609252534	0.88634001484781
BaggingClassifier	Nouns and Adjectives	Stop Words	0.931696138334696	0.9331656971113007	0.932293986636971
LogisticRegression	Nouns and Adjectives	Stop Words	0.9612755251315364	0.9600450447528224	0.9608017817371938
SVC	Adjectives	Stop Words	0.9073247891195408	0.9073168076394611	0.907720861172977
LinearSVC	Adjectives	Stop Words	0.9129620316472402	0.9113460572811238	0.9125463994060876

SGDClassifier	Adjectives	Stop Words	0.8895663827543937	0.891107945621708	0.8900519673348181
Perceptron	Adjectives	Stop Words	0.9014985360256321	0.8967215021430045	0.8991833704528582
PassiveAggressiveClassifier	Adjectives	Stop Words	0.8773985134583124	0.8789474188794691	0.8778025241276912
MLPClassifier	Adjectives	Stop Words	0.9207568743859067	0.9200675166475105	0.9207869339272458
MultinomialNB	Adjectives	Stop Words	0.9018669449012877	0.9023606648540333	0.9024498886414254
RandomForestClassifier	Adjectives	Stop Words	0.9226020855895565	0.9238638418482015	0.9232368225686711
KNeighborsClassifier	Adjectives	Stop Words	0.7440223806008006	0.7420387721330528	0.7373422420193022
DecisionTreeClassifier	Adjectives	Stop Words	0.853576504834174	0.8531015295275679	0.8540460282108389
AdaBoostClassifier	Adjectives	Stop Words	0.8141429636059903	0.8106173735373612	0.8051967334818114
GradientBoostingClassifier	Adjectives	Stop Words	0.828809236725293	0.8247716958544387	0.8191536748329621
BaggingClassifier	Adjectives	Stop Words	0.8774362093097248	0.8790688296372704	0.8774313288789903
LogisticRegression	Adjectives	Stop Words	0.9140860194907874	0.913012253043356	0.9139569413511507
SVC	Nouns	TF-IDF	0.9706849862680036	0.9707125302566524	0.9708240534521159
LinearSVC	Nouns	TF-IDF	0.966164495275594	0.9655382754305388	0.9659985152190051
SGDClassifier	Nouns	TF-IDF	0.9554627243336813	0.9553542963411312	0.9556050482553823
Perceptron	Nouns	TF-IDF	0.95862918	0.95800280	0.95850037

			83303295	44368387	11952487
PassiveAggressiveClassifier	Nouns	TF-IDF	0.9643830935870881	0.963600873254099	0.9641425389755011
MLPClassifier	Nouns	TF-IDF	0.9639824207243082	0.9628478099379072	0.9635486265775798
Multinomial NB	Nouns	TF-IDF	0.928453895516794	0.9292489353614971	0.9290274684484039
Random Forest Accuracy	Nouns	TF-IDF	0.963937324587038	0.9644056803363128	0.9642910170749814
K-Nearest Neighbor Accuracy	Nouns	TF-IDF	0.7611874455100262	0.6326615396867933	0.6551596139569413
DecisionTreeClassifier	Nouns	TF-IDF	0.9212049695920663	0.9205148777054799	0.9212323682256867
AdaBoostClassifier	Nouns	TF-IDF	0.9068435110914641	0.9054888932349421	0.9066072754268746
GradientBoostingClassifier	Nouns	TF-IDF	0.9200521533949324	0.9217461224999675	0.9204899777282851
BaggingClassifier	Nouns	TF-IDF	0.9394798730126666	0.9408526132738457	0.9400890868596882
LogisticRegression	Nouns	TF-IDF	0.9548438311292194	0.9543237274659164	0.9547884187082405
SVC	Nouns and Verbs	TF-IDF	0.9737289216065776	0.9737841770397566	0.9738678544914625
LinearSVC	Nouns and Verbs	TF-IDF	0.9697090352791329	0.9687417946442922	0.9693392724573126
SGDClassifier	Nouns and Verbs	TF-IDF	0.9599038946271332	0.9592654342498697	0.9597624350408315
Perceptron	Nouns and Verbs	TF-IDF	0.9631053181643054	0.9630780331638793	0.9632516703786191

PassiveAggressiveClassifier	Nouns and Verbs	TF-IDF	0.9677380650734395	0.9669609743734183	0.9674832962138085
MLPClassifier	Nouns and Verbs	TF-IDF	0.9668405373817373	0.9652861546501825	0.9661469933184855
MultinomialNB	Nouns and Verbs	TF-IDF	0.928469255560925	0.9292000035160077	0.9290274684484039
RandomForestClassifier	Nouns and Verbs	TF-IDF	0.9680854440297596	0.9682557752142755	0.9682999257609503
KNeighborsClassifier	Nouns and Verbs	TF-IDF	0.7474192040208054	0.5937363673055476	0.6192279138827023
DecisionTreeClassifier	Nouns and Verbs	TF-IDF	0.9283450786676629	0.9277020137402836	0.9283593170007424
AdaBoostClassifier	Nouns and Verbs	TF-IDF	0.9109337651341205	0.9100384699512305	0.910913140311804
GradientBoostingClassifier	Nouns and Verbs	TF-IDF	0.9066878258315979	0.9067116247739586	0.9028210838901262
BaggingClassifier	Nouns and Verbs	TF-IDF	0.9434419643374867	0.9444764592544633	0.9440237564959169
LogisticRegression	Nouns and Verbs	TF-IDF	0.958435551821613	0.958040787505136	0.9584261321455085
SVC	Nouns and Adjectives	TF-IDF	0.9779783121138792	0.9781967445886748	0.978173719376392
LinearSVC	Nouns and Adjectives	TF-IDF	0.9751747330050227	0.9743081795380706	0.9748329621380847
SGDClassifier	Nouns and Adjectives	TF-IDF	0.9693398859826262	0.9690647448245219	0.9693392724573126
Perceptron	Nouns and Adjectives	TF-IDF	0.9716056790856069	0.9715778932652975	0.9717149220489978
PassiveAggressiveClassi	Nouns and Adjectives	TF-IDF	0.9775162812333865	0.9772957233279208	0.9775055679287306

fier					
MLPClassifier	Nouns and Adjectives	TF-IDF	0.9739317386367752	0.9728878054113794	0.9734966592427617
MultinomialNB	Nouns and Adjectives	TF-IDF	0.9314168838182727	0.9322802742601156	0.9319970304380104
RandomForestClassifier	Nouns and Adjectives	TF-IDF	0.969657166656187	0.9701712798363311	0.970007423904974
KNeighborsClassifier	Nouns and Adjectives	TF-IDF	0.7651891712938224	0.6010921964312035	0.6265033407572383
DecisionTreeClassifier	Nouns and Adjectives	TF-IDF	0.9288246014665003	0.9279720577463971	0.9287305122494433
AdaBoostClassifier	Nouns and Adjectives	TF-IDF	0.9169356254767159	0.9156975817992665	0.9167037861915367
GradientBoostingClassifier	Nouns and Adjectives	TF-IDF	0.9333402775413553	0.9351352703155668	0.9337045285820341
BaggingClassifier	Nouns and Adjectives	TF-IDF	0.949448895964714	0.9509415405928073	0.9500371195248701
LogisticRegression	Nouns and Adjectives	TF-IDF	0.9674197843782819	0.9671066075017922	0.9674090571640683
SVC	Adjectives	TF-IDF	0.9358432811989412	0.9361142836241653	0.9362286562731997
LinearSVC	Adjectives	TF-IDF	0.9250343020803268	0.923868679679984	0.9247958426132146
SGDClassifier	Adjectives	TF-IDF	0.9131300742981119	0.9124721650311289	0.913214550853749
Perceptron	Adjectives	TF-IDF	0.9107721482321678	0.9105398995914102	0.9110616184112843
PassiveAggressiveClassifier	Adjectives	TF-IDF	0.8929473334572957	0.8941583701541781	0.893615441722346
MLPClassifier	Adjectives	TF-IDF	0.9142248155692692	0.9131711597651103	0.9141054194506311

Multinomial NB	Adjectives	TF-IDF	0.9067774924873386	0.9076546809255097	0.9074239049740164
RandomForestClassifier	Adjectives	TF-IDF	0.9285958303904891	0.9289161988121695	0.9290274684484039
KNeighbors Classifier	Adjectives	TF-IDF	0.7537198608266289	0.6418676678756966	0.6631031922791388
DecisionTreeClassifier	Adjectives	TF-IDF	0.849820342770101	0.8485358672973664	0.8499628804751299
AdaBoostClassifier	Adjectives	TF-IDF	0.8195299992380158	0.8150277712578369	0.8181143281365999
GradientBoostingClassifier	Adjectives	TF-IDF	0.8567623586856752	0.8570793343081472	0.8574610244988864
BaggingClassifier	Adjectives	TF-IDF	0.8829172123405951	0.8845506244334507	0.883147735708983
LogisticRegression	Adjectives	TF-IDF	0.9125056231715822	0.9119050427277967	0.9126206384558277

Part 4:

Please tell me what you plan to do in the future if you want to further enhance the performance of the machine learning models, e.g., enhancing learning models? Or investigating novel features?

BERT! Ever since BERT came out a couple of years ago it has taken over. Everyone seems to be using it these days. In the future I think we could get better results if BERT was fine-tuned along with adding a classifier in at the end. It would probably make sense to use a couple of fully connected layers and a sigmoid function as the classifier. Using this model should give us a better understanding of the context of the words in the sentence.

Another thing that could be done is to add a satire label to the model. Sometimes people are making fun of people who believe the fake information. It is hard to classify this as true or fake information because it really isn’t either. Hence why there should be a satire label. Given only true and fake labels it would be hard for a model to be able to classify satire text snippets correctly

It would be interesting to do a combination of fact checking and classification. For each of the statements in the text snippets fact checking could be done to make sure that the information is true. From there it would be simple to classify it as true vs false just based on if the whole statement is true. If the fact check is unsure the text could be classified based on the wording using a classifier. If the fact checker determines that something is false the statement could then be passed into a context classifier to determine if it is satire or not.

Try an LSTM. It would be interesting seeing the results of an LSTM classifying the text snippets. Using an LSTM would allow for more contextual understanding.

It would also be interesting to use a genetic optimizer. I am personally obsessed with this type of machine learning. It is the part of machine learning I would like to do a lot of research in the future.

This isn't strictly about machine learning, but it would be interesting to get a crowd sourced dataset of people on the internet voting whether or not a statement is false. Then given that dataset it would be interesting to compare it to the classification of several different machine learning models. It would be most interesting to look at the ones that mess up both humans and computers and figure out why. There may be a way to phrase or put a statement that makes it look believable even if the information is not.

It would be interesting to fact check inside the same statement. The model would be checking for consistency in the statement. For example it would check to make sure that the ball that was thrown through the window doesn't go from green to red in the same statement. If facts are changing throughout the statement then it would seem that the statement is trying to be deceptive. This approach would probably work less for written text and more for speech since written text can be proof read. It could be applied to such things as verbal questions and answers and videos. I'm not sure how to go about building a model to do such a thing, but it would be interesting to work on.

Try a standard RNN. It would be interesting almost just as a baseline to compare the other models.