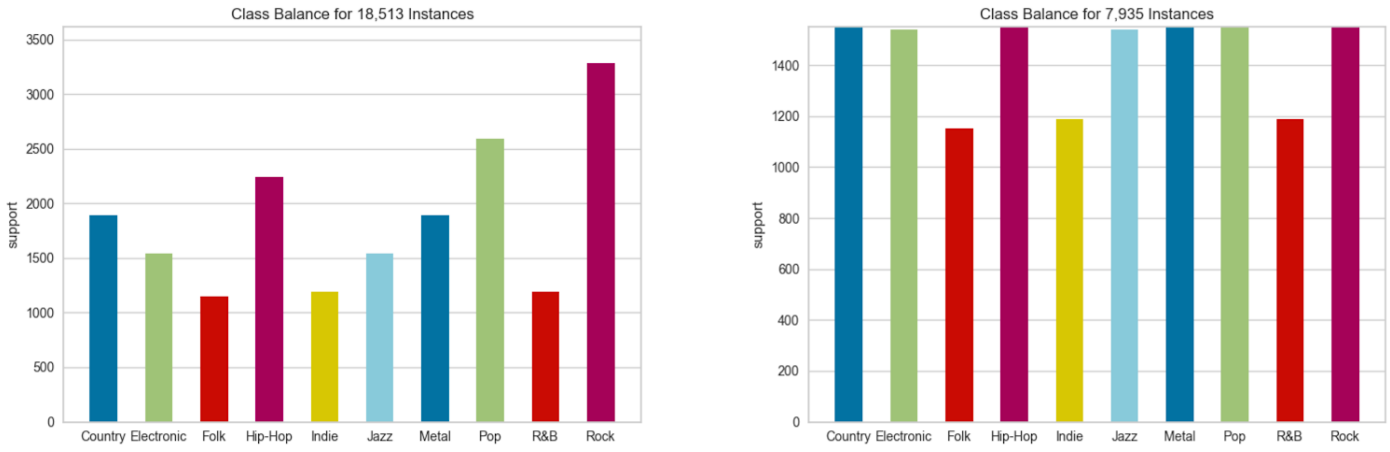
Classify song using the lyrics

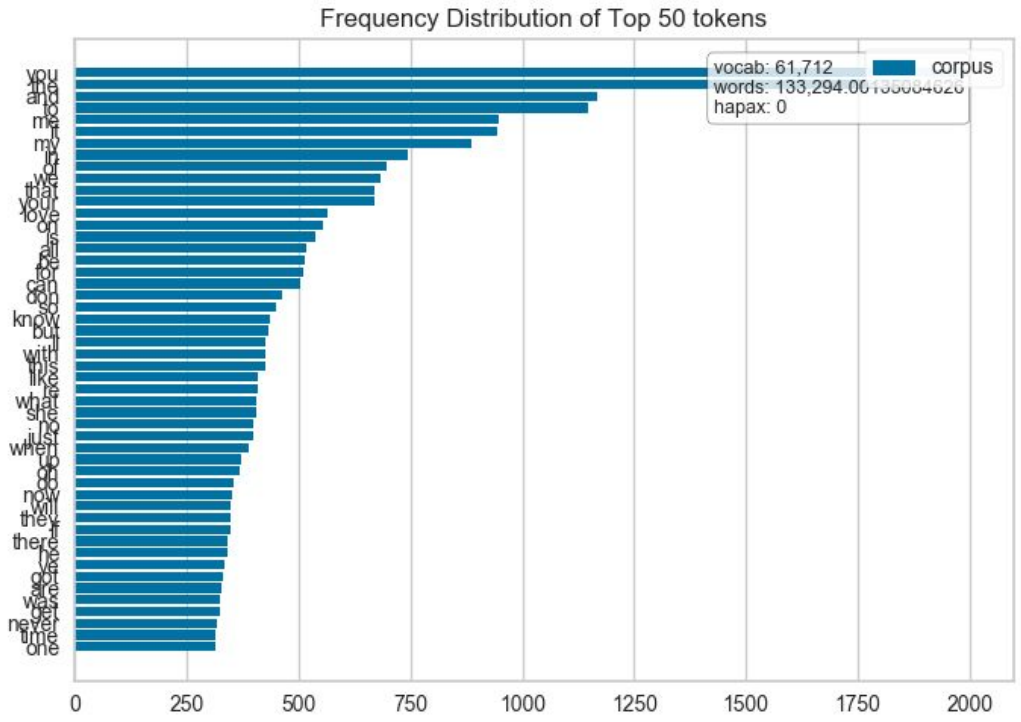
The dataset

First of all I analyzed the dataset. There are 10 classes: *Country, Electronic, Folk, Hip-Hop, Indie, Jazz, Metal, Pop, R&B and Rock*. The training dataset is not that balance, having around 3300 samples for Rock class and only around 1100 for R&B and Folk.

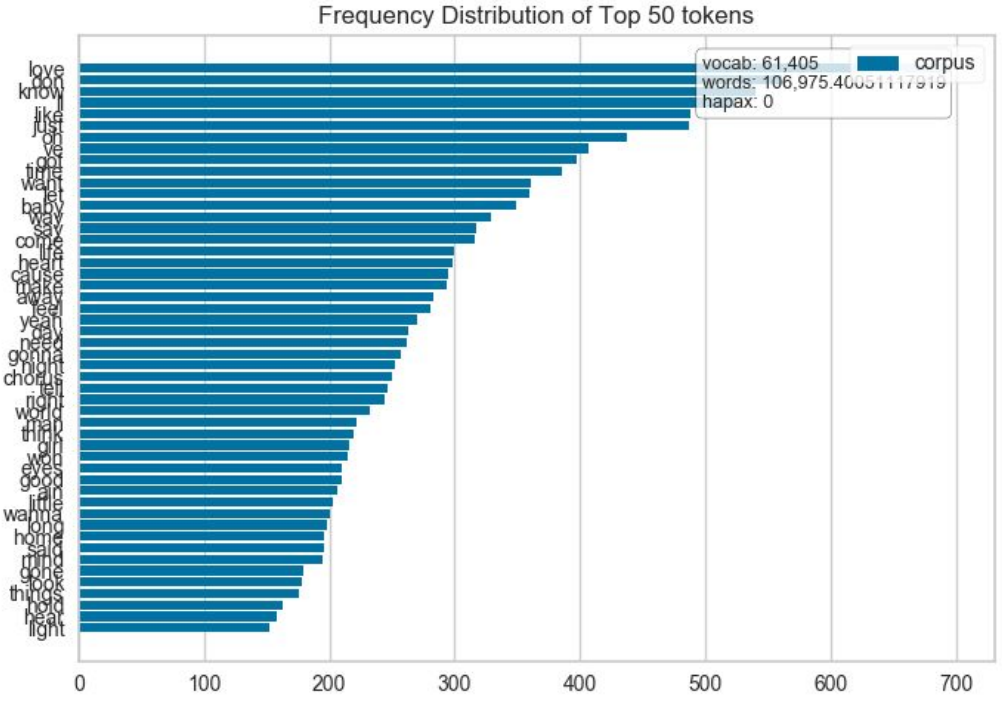


The class balance for Training(left) and Test(right) dataset

The next step is to try and see the term frequency when using stopwords and when we remove them. We can see that after we remove the stopwords, the word love advances from 13th place to the first. The stopwords should have a great importance when trying to do the prediction, but we will the both methods and compare the results.

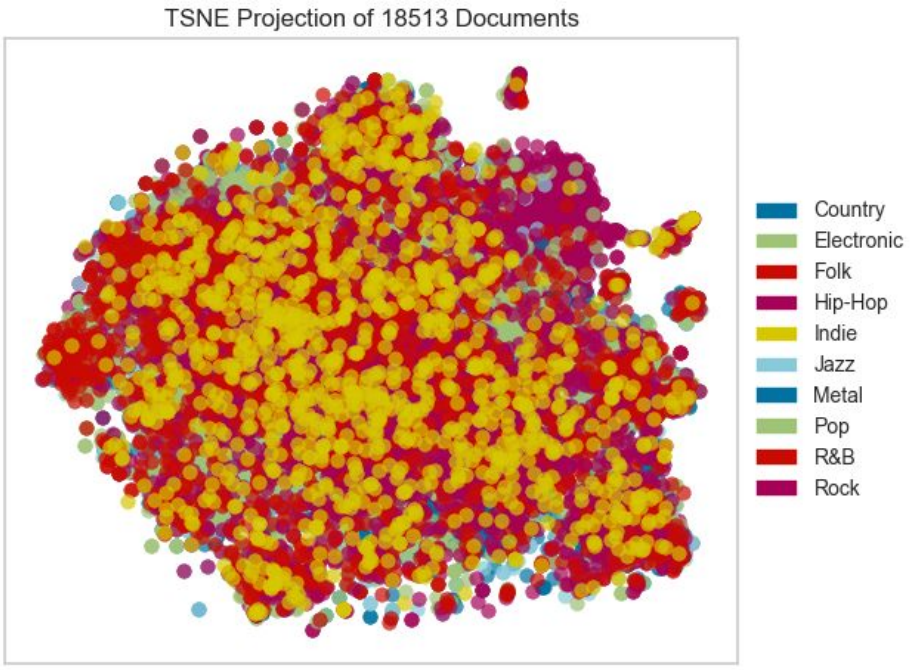


Term frequency from TfidfVectorizer with stopwords



Term frequency from TfidfVectorizer with stopwords removed

Also I tried to visualize how the genres are separated using T-SNE. The result was kind of surprising. The data doesn't look separated, so I don't expect the results to be that great.



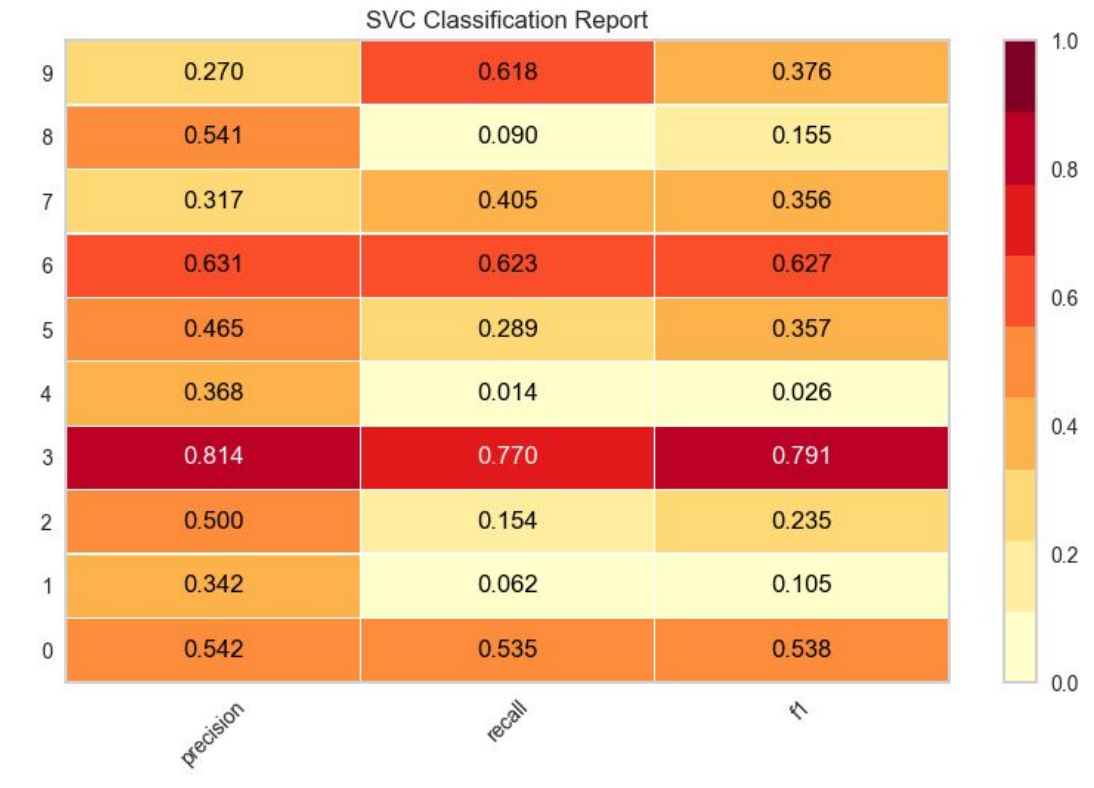
TSNE Projection using TfId

Results

TfIdfvectorizer preprocessing

I tested more TfId params for preprocessing and trained SVC and Random Forest Classifier. The results were similar overall but we can see a small improvement for models that kept the stopwords. The best results were obtained using using (3, 3) ngrams with an accuracy of 42.33%.

Model	Tfld params	Model Params	Score
SVC	max_df: 0.5	C: 0.5, kernel: "linear"	41.43%
Random Forest	max_df: 0.5	n_estimator: 1000	39.16%
SVC	max_df: 0.5, stop_words: "english"	C: 0.5, kernel: "linear"	40.45%
Random Forest	max_df: 0.5, stop_words: "english"	n_estimator: 1000	40.74%
SVC	max_df: 0.5, stop_words: "english", analyzer: "char", ngrams_range: (2,2)	C: 0.5, kernel: "linear"	37.40%
Random Forest	max_df: 0.5, stop_words: "english", analyzer: "char", ngrams_range: (2,2)	n_estimator: 1000	36.69%
SVC	max_df: 0.5, stop_words: "english", analyzer: "char", ngrams_range: (3,3)	C: 0.5, kernel: "linear"	42.33%
random Forest	max_df: 0.5, stop_words: "english", analyzer: "char", ngrams_range: (3,3)	n_estimator: 1000	38.51%

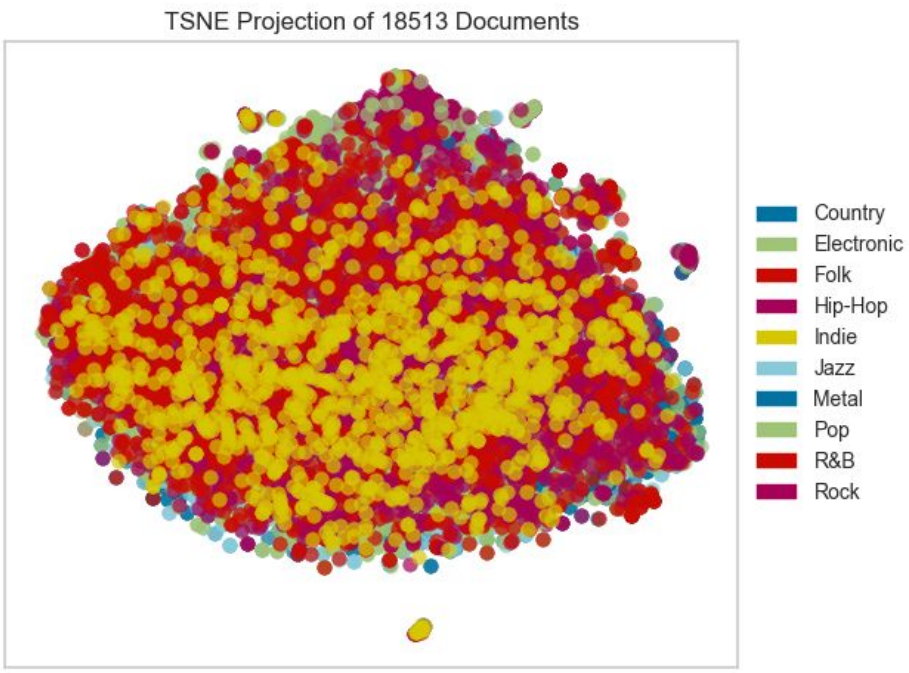


Classification report of the SVC model with 42.33%

Also tried a neural network with 3 layers, but the results were not great. The accuracy on the test dataset was 38%, but the network overfitted on the train dataset.

Word 2 vec

I try to extract new features using Word2vec model. I trained the gensim word2vec model on the training corpus. From the TSNE we can see that the data is not separated, but we can see it's a little bit more grouped.



Word 2 vec TSNE

The result weren't that different comparing to the results of tfid. Neural network achieved 41.33%, but the score for random forest remains the same.

Model	**Model Params**	**Score**
Random Forest	n_estimator: 1000	37.41%
Neural network	3 layers	41.33%

Conclusion

The results were not great, the highest accuracy obtained was 42.33% using the (3, 3) ngrams and the word 2 vec model didn’t improved the accuracy by a lot.