Project Report
Sentiment Analysis on Customer Reviews of a Photo Company

Gokay Bulut

Note:
Since the company owns all intellectual property rights, this is not the original report provided to the company. Rather, it is a report explaining my work.

Summary

In this project, I studied how customers expressed their feeling about the products / services of a Photo Company and the correlation between the reviews and the rating of the products / services. I employed 12 machine learning algorithms to predict sentiments of customers, namely, Logistic Regression, K-Nearest Neighbors, Linear Support Vector Machines (SVM), Kernel SVM, Naive Bayes, Decision Trees, Random Forest, AdaBoost, Gradient Boosting, Stochastic Gradient Boosting, Extreme Gradient Boosting (XGBoost) and CatBoost. I used these algorithms with 7 different bag of words methods, i.e. Count Vectorizer, TfIdf Vectorizer, Hashing Vectorizer, SMOTE, PCA with SMOTE, Truncated SVD with SMOTE and lastly, Word2Vec. Finally, I implemented deep learning models with Word2Vec, GloVe, FastText, Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) Long Short Term Memory (LSTM). Best average F1 score of 85% was achieved by Count Vectorizer with Naïve Bayes algorithm.

1. INTRODUCTION

1.a. General

Natural Language Processing (NLP) serves different purposes when dealing with text or unstructured text data. One of the subtopics of this research is called sentiment analysis or opinion mining, which is, given some text, we can computationally study peoples' opinions, appraisals, attitudes, and emotions toward entities, individuals, issues, events, topics and their attributes. Applications of this technique are diverse. For example, businesses always want to find public or consumer opinions and emotions about their products and services. Potential customers also want to know the opinions and emotions of existing users before they use a service or purchase a product. Further, researchers use this information to do an in-depth analysis of market trends and consumer opinions, which could potentially lead to a better prediction of the stock market. The average human reader will have difficulty identifying relevant sites and accurately summarizing the information and opinions contained in them. So, we need to use NLP to do the job.

1.b. Problem

A Photo Company provided its data and wanted to gain more information on how the reviews were actually worded. The company did not only want a basic classification of good and bad reviews but also how the customers felt about the products / services and how they expressed their feelings, which words they used the most. To excel in the service it provides, the company believes the word 'good' is not enough.

1.c. Data Set

My dataset comes from customer reviews of the photo company products / services which are related with photo shooting and selling related products. The data was obtained from the company. This dataset has 5157 data points in total. Translated to English, each record has the features below:

- name – name of the reviewer (dropped for privacy reasons)
- lastname – lastname of the reviewer (dropped for privacy reasons)
- sex - sex of the reviewer
- rating - rating given by the reviewer for the type of product / service
- product - type of product / service
- review - the text of the review
- review_date - date the review was provided
- reviewer_id - unique number of the reviewer

1.d. Methodology

My goal is to capture most used words by the customers and build a sentiment analysis model that predicts whether a user liked a product or not, based on the reviews of the customers. First, I extracted the features of my dataset and built several supervised models. These models not only include traditional algorithms such as Logistic Regression, K-Nearest Neighbors, Linear Support Vector Machines (SVM), Kernel SVM, Naive Bayes, Decision Trees, Random Forest, AdaBoost, Gradient Boosting, Stochastic Gradient Boosting, Extreme Gradient Boosting (XGBoost) and CatBoost; but also deep learning with Word2Vec, GloVe, FastText, Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) Long Short Term Memory (LSTM). Then, I compared the accuracy of these models and got a better understanding of the polarized attitudes towards the products / services.

2. DATA WRANGLING

2.1. Initial Understanding

The initial look of the data set after translating column names to English snake_case naming convention and dropping the ‘name’ and ‘lastname’ columns for privacy reasons:

	sex	rating	product	review	review_date	reviewer_id
0	k	5.0	hizmet	İstanbul'da en sevdiğim mekan, analog dostu. M...	03/08/2017	44720
1	k	5.0	Stüdyo	KESİNLİKLE ÇOK GÜZEL BİR STÜDYO	10/17/2017	46945
2	k	5.0	Hizmet	Burayı hep sevdim	01/03/2017	92805
3	e	5.0	Düğün	Düğün fotoğrafı için gitmiştik, çok güzel çeki...	01/19/2017	51670
4	k	2.0	Hizmet	O kadar iyi diil tabi. Isim var sadece	11/05/2017	79719

2.2. Information – info()

One of the basic and common ways to examine the data is using “info()” method. It is simple but tells a lot.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5157 entries, 0 to 5156
Data columns (total 8 columns):
name           5157 non-null object
lastname       5157 non-null object
sex            5157 non-null object
rating         4968 non-null float64
product        4989 non-null object
review         5016 non-null object
review_date    5157 non-null object
reviewer_id    5157 non-null int64
dtypes: float64(1), int64(1), object(6)
memory usage: 322.4+ KB
```

- What to learn from the information:
  - I have the shape, 5157 observations (records or rows) and 8 columns (or variables).
  - There is redundancy in columns. Since ‘reviewer_id’ provides enough information on the reviewers, I dropped the 'name' and 'lastname' columns in order to take care of the redundancy and keep privacy of the customers who provided the reviews to the company.
  - There are missing values in ‘rating’, ‘product’ and ‘review’ columns. Since they are not too many, I am going to drop them.
  - The ‘review_date’ is a variable related with date but data type is "object". I will convert to “datetime” for easier computing on dates.
  - By using the value_counts() method, I see that the ‘sex’ column has 2 and the ‘product’ column has 11 categories. First, I will translate them to English for better understanding. In order to facilitate efficient computing, I will change the data types of ‘sex’ and ‘product’ columns to “categorical”.

- Design of reshaping:
  - ‘name’: column dropped
  - ‘lastname’: column dropped
  - ‘sex’: column will be converted to “categorical”
  - ‘product’: column will be converted to “categorical”
  - ‘review_date’: column will be converted to “datetime”

- Issues fixed:
  - 2 redundant columns dropped
  - 2 column types converted to categorical
  - 1 column type converted to datetime
  - column names were changed to English snake_case naming convention

2.3. Statistics summary – describe( )

Numeric feature: ‘rating’

```
count    1064.000000
mean        4.178571
std         1.495075
min         1.000000
25%         4.000000
50%         5.000000
75%         5.000000
max         5.000000
Name: rating, dtype: float64
```

Non-numeric features

The total number of reviewers: 1040
The total number of reviews: 1109
The total number of reviewed products or service: 11
The total number of review days: 348
The total number of unique reviews: 1064

- Rating:
  - Mean of the ratings is more than 4 out of 5. It means that people are inclined towards giving high ratings. "std" value (1.495) and percentile values show that 2,3 and 4 star ratings are rare.
  - Small numbers of "ratings under 4" will decrease the predictability of these ratings. To overcome this problem I need to split the ratings in to two groups as "good" and "bad" ratings.

- Non-numeric variables statistics:
  - In total, 1040 reviewers wrote 1109 reviews on 11 products/services/topics in 348 unique days. Some reviewers wrote more than one review and some reviews are the same.

2.4. Text Preprocessing

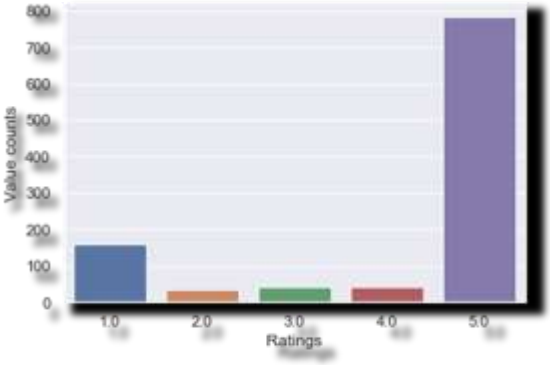
In the context of corpus normalization, I applied advanced text cleaning techniques such as:

- Lowercase the text
- Keep only words
- Html removal
- Whitespace and underscores removal
- Special characters removal
- Accent marks removal
- Lemmatization
- Stop words removal

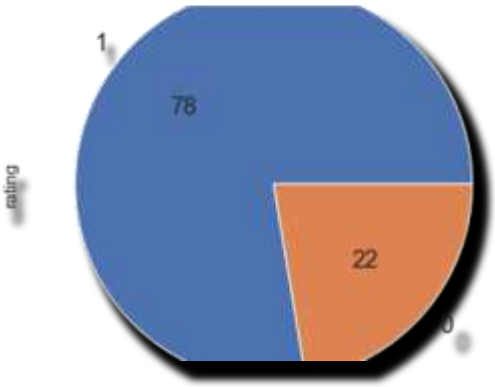
and saved the resulting column as ‘clean_text’.

3. EXPLORATORY DATA ANALYSIS

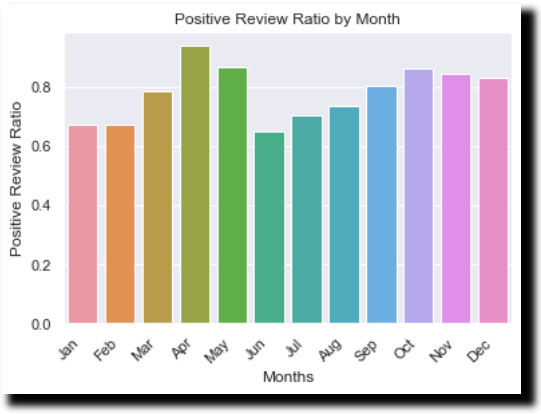
3.1. 'rating' Feature



- There is an imbalance between rating classes, most of the ratings are 5.0
- There are not many ratings in the middle range, i.e. in range 2.0-4.0.

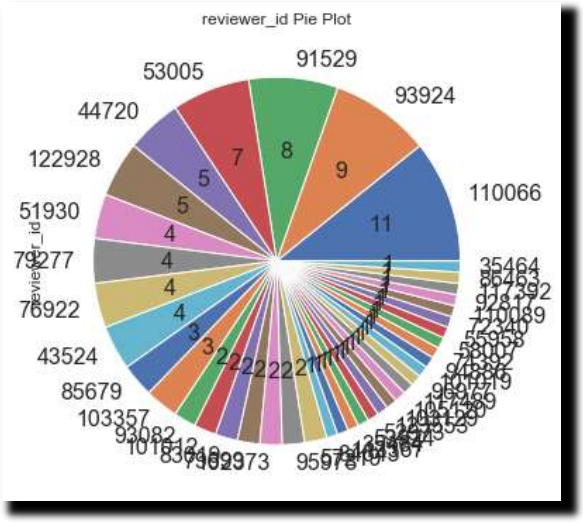


- I categorized 'rating' values > 3 as 1 and 'rating' values <= 3 as 0 to do a binary classification.
- Around 78% of reviews are positive.



- Looking at the monthly ratio of positive reviews, there is an increase around April.
- There is a steady increase in number of positive reviews towards winter.

3.2. ‘reviewer_id’ Feature

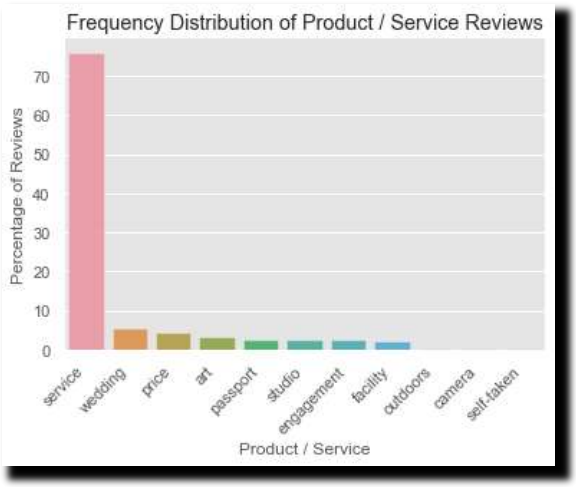


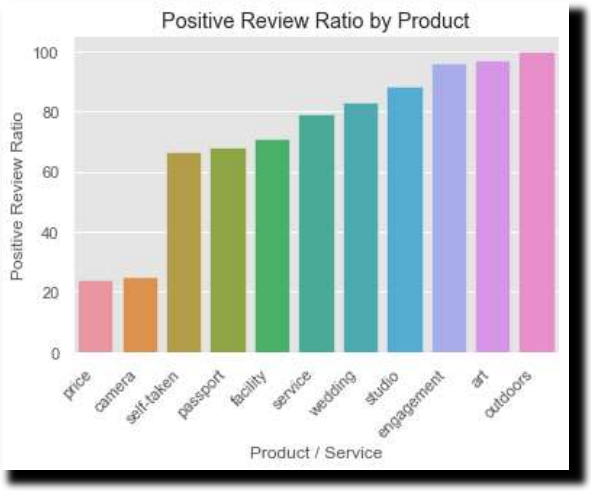
- Some reviewers write more reviews.

	sex	rating	product	review	review_date	reviewer_id
0	f	1	service	İstanbul'da en sevdiğim mekan, analog dostlu lü...	2017-03-08	44720
118	f	1	service	Kendini kandlamış bir Fotoğraf Stüdyosu Herke...	2017-05-07	44720
180	f	1	service	İşini düzgün yapan bir işletme...	2017-08-06	44720
1146	f	1	service	Filmlerinizi yıkatmak için bir yer arıyorsanız...	2017-12-07	44720
1182	f	1	price	Film yıkatmak için öğrenci dostu mekan. Renkli...	2017-01-09	44720

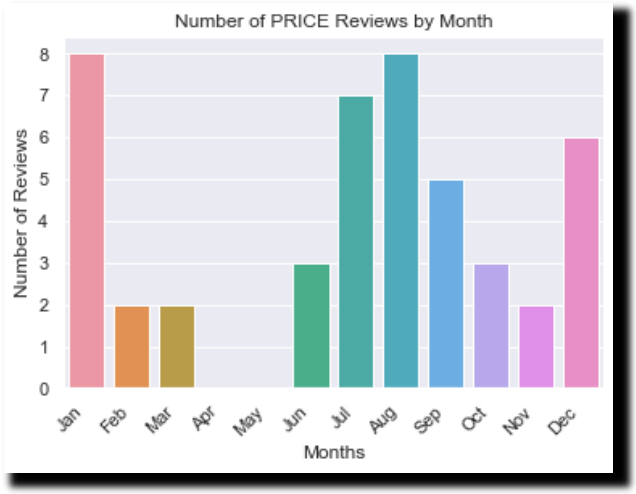
- After verifying validity,
- It would be beneficial for the company to keep track of these customers.

3.3. ‘product’ Feature:



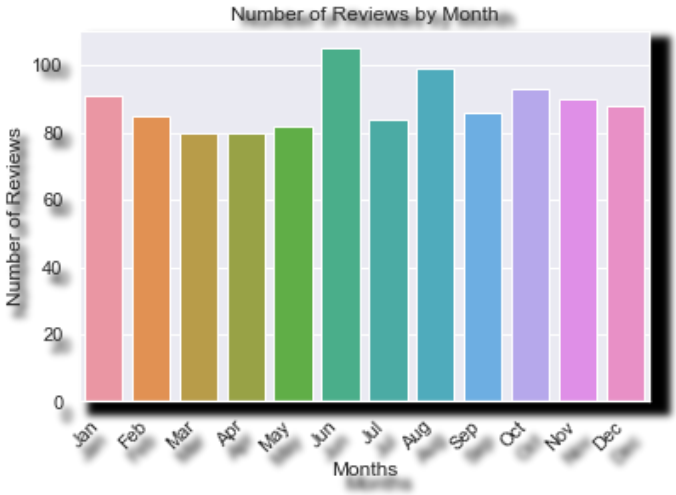


- Around 76% of the reviews are on the service provided, i.e. customer satisfaction where percentage of the positive reviews are 79%.
- Not that much, but wedding photos and the prices are also two important categories that are reviewed relatively often.
- Prices got the lowest rating, but they are reviewed only by 1% of the reviewers.
- Similarly, outdoors photo shoots got the highest rating and they are reviewed by only .5% of the reviewers.



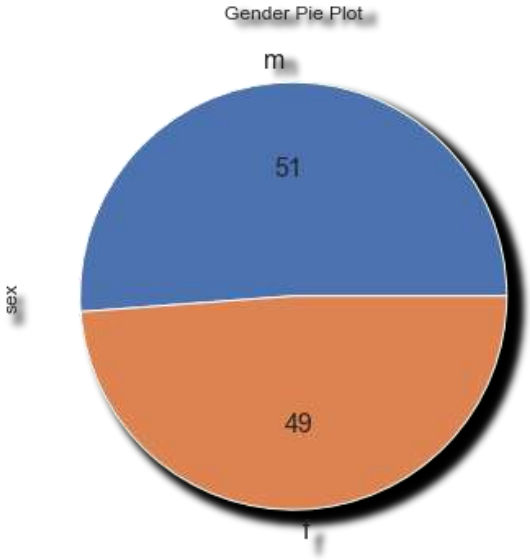
- Review highs on the prices might indicate price changes in July and December.

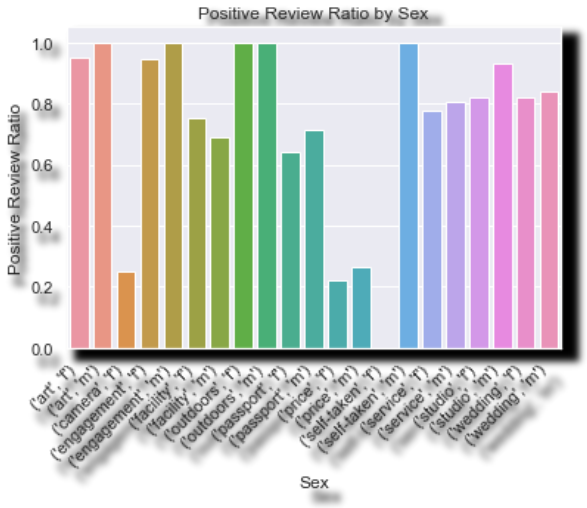
3.4. 'review_date' Feature:



- The data covers the period between January 1st, 2017 and December 31st, 2017.
- The number of reviews show slight decrease towards spring and there is an increase in summer months.

3.5. 'sex Feature:





product	sex	Number_Total_Reviews	Number_Positive_Reviews	Ratio_Positive_Reviews
art	f	20	19	0.950000
	m	15	15	1.000000
camera	f	4	1	0.250000
engagement	f	19	18	0.947368
	m	6	6	1.000000
facility	f	8	6	0.750000
	m	16	11	0.687500
outdoors	f	3	3	1.000000
	m	2	2	1.000000
passport	f	14	9	0.642857
	m	14	10	0.714286
price	f	27	6	0.222222
	m	19	5	0.263158
self-taken	f	1	0	0.000000
	m	2	2	1.000000
service	f	384	299	0.778646
	m	424	341	0.804245
studio	f	11	9	0.818182
	m	15	14	0.933333
wedding	f	28	23	0.821429
	m	31	26	0.838710

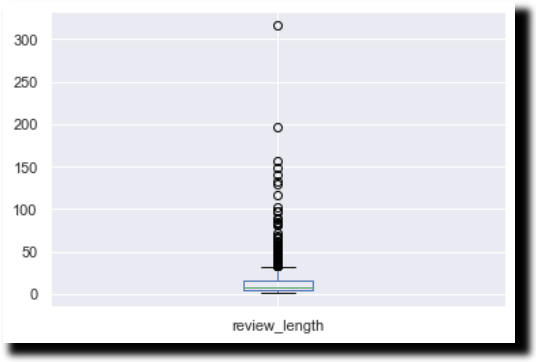
- Around 51% of the reviewers are male and 49% are female.
- All the reviews on camera sales are made by female reviewers.
- Engagement photos are more reviewed by women.

- Men made more reviews on facility and service provided.
- The only significant difference between men and women reviews seems to be in the category 'self-taken'. There are only 3 reviews here. 2 of them are positive and made by males, 1 of them is negative and made by a female. However, we don't have enough data to conclude that female reviews on this category are significantly different than male reviews.

3.6. 'review' feature

All reviews

```
count    1063.000000
mean       14.034807
std        19.958000
min         1.000000
25%         5.000000
50%         8.000000
75%        16.000000
max       317.000000
Name: review_length, dtype: float64
```



- Minimum review length is 1 word, and the maximum is 317 words.
- On average, reviews are around 14 words long.

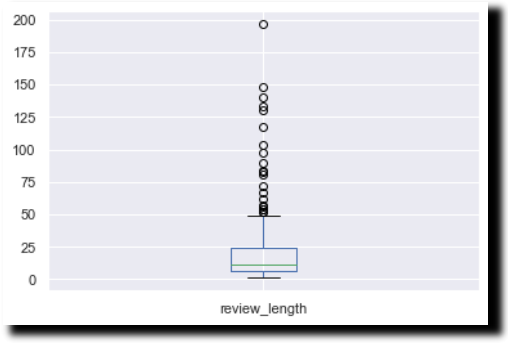
Positive reviews

```
count    825.000000
mean      12.156364
std       16.759102
min        1.000000
25%        4.000000
50%        8.000000
75%       14.000000
max      317.000000
Name: review_length, dtype: float64
```

- Minimum positive review length is also 1 word, and the maximum is also 317 words.
- On average, positive reviews are around 12 words long.

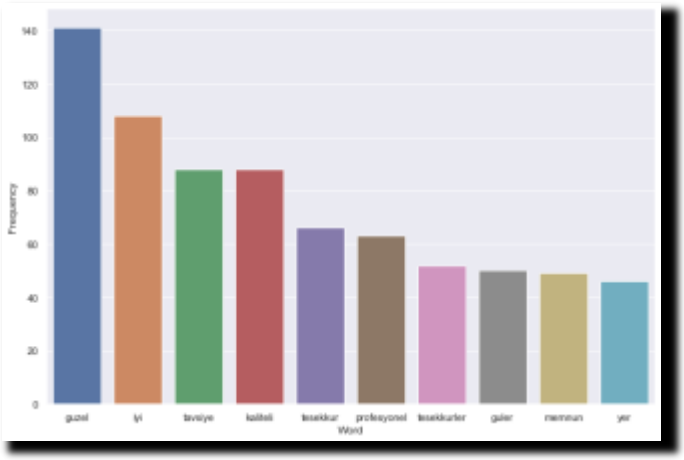
Negative reviews

```
count    238.000000
mean      20.546218
std       27.450069
min        1.000000
25%        6.000000
50%       11.000000
75%       23.750000
max      197.000000
Name: review_length, dtype: float64
```



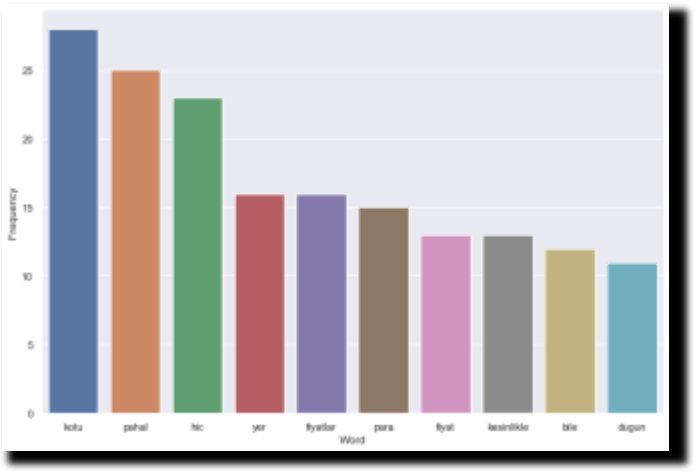
- Minimum negative review length is 1 word, and the maximum is 197 words.

- On average, negative reviews are around 21 words long.
- Negative reviews tend to be a little longer, as expected.



- I used the Word Cloud above to restrict some frequent words that do not contribute to the analysis.
- As a result, the words:
  - beautiful, good, quality,
  - recommend, thanks, professional,
  - smiling, glad and fast

  emerged as the most used in positive reviews (translated to English).



- Similarly, after restricting frequent words that do not contribute to the analysis; the words:
  - bad, expensive, never,
  - prices, place, absolutely,

- money, even and wedding

emerged as the most used in negative reviews (translated to English).

3.7. EDA Summary

- 'review_date' spans Jan 1st - Dec 31st 2017,
- around 78% of the reviews are positive → imbalanced data,
- 'rating' column categorized as 0 and 1 to help reduce imbalance,
- around 51% of the reviewers are male,
- ratio of positive reviews does not significantly differ by gender,
- there are reviewers who provide valuable feedback by writing more reviews,
- it would be beneficial for the company to keep track of these customers,
- around 76% of the reviews are about the service provided, i.e. customer satisfaction (around 79% positive),
- wedding photos (around 83% positive) and prices (around 24% positive) are also two important categories reviewed relatively often (around 6% and 4% respectively),
- prices got the lowest rating, but they are reviewed only by 4% of the reviewers,
- similarly, outdoors photo shoots got the highest rating (100%) and they are reviewed by only 0.5% of the reviewers,
- review highs on the prices might indicate price changes in July and December,
- minimum review length is 1 word, and the maximum is 317 words. On average, reviews are around 14 words long,
- on average, positive reviews are around 12 words long.
- on average, negative reviews are around 21 words long.
- negative reviews tend to be longer, as expected,
- Word Clouds show most frequent words are shared between positive and negative reviews,
- after eliminating these and other non-essential words,
  - the words beautiful, good, quality, recommend, thanks, professional, smiling, glad and fast emerged as the most used in positive reviews (translated to English),
  - the words bad, expensive, never, prices, place, absolutely, money, even and wedding emerged as the most used in negative reviews (translated to English).

-

- **Recap crucial points:**
  - Among rating classes,
    - Data set is imbalanced,
    - There is a great number of matching words.

- **Conclusion:**
  - reduce the imbalance with reducing the number of rating classes.

4. FEATURE ENGINEERING AND MODELING

In accordance with EDA findings, the number classes (ratings) has been reduced. Five classes have been split into two groups as “negative” (1, 2, 3) and “positive” (4, 5). So, the analysis became a supervised binary-classification problem. I am trying to predict the ratings based on the reviews left by customers who bought products / services from the photo company. I employed 12 machine learning algorithms to predict sentiments of customers, namely,

- Logistic Regression,
- K-Nearest Neighbors,
- Linear Support Vector Machines (SVM),
- Kernel SVM,
- Naive Bayes,
- Decision Trees,
- Random Forest,
- AdaBoost,
- Gradient Boosting,
- Stochastic Gradient Boosting,
- Extreme Gradient Boosting (XGBoost) and
- CatBoost.

In regards of feature engineering, review data has been vectorized with 7 different methods. These bag of words methods are

- Count Vectorizer,
- TfIdf Vectorizer,
- Hashing Vectorizer,
- SMOTE,
- PCA with SMOTE,
- Truncated SVD with SMOTE,
- Word2Vec.

Finally, I implemented deep learning models with

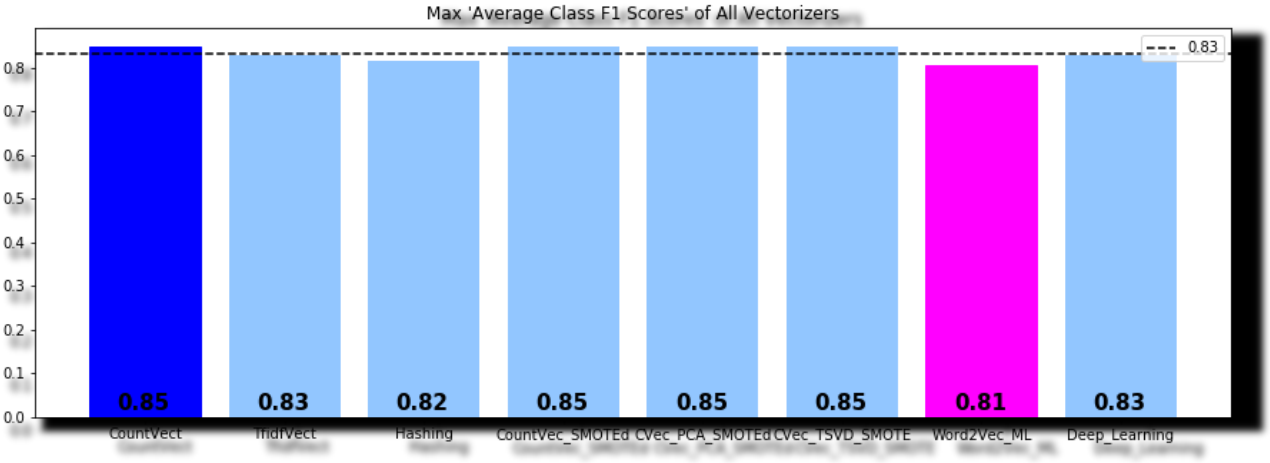
- Word2Vec,
- GloVe,
- FastText,
- Convolutional Neural Networks (CNN),
- Recurrent Neural Networks (RNN) Long Short Term Memory (LSTM).

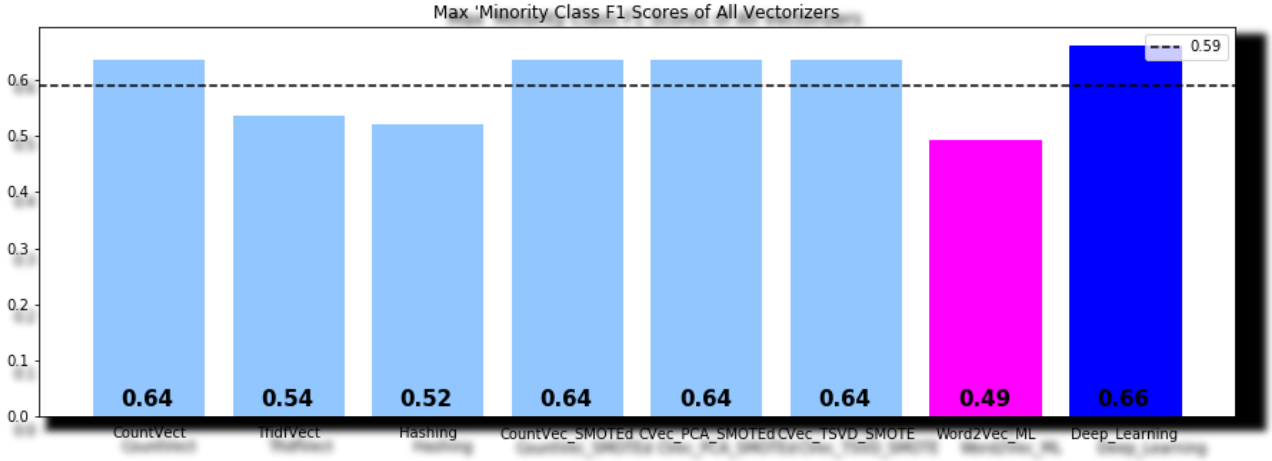
Best average F1 score of 85% was achieved by Count Vectorizer with Naïve Bayes algorithm.
The model results are included at the end of the report as an attachment.

5. CONCLUSION

In this project, I tried to capture the most used words by the customers in positive and negative reviews and predict the sentiments of customers based on their reviews. Here are the results of modeling:

- Overall, the best average F1 score of 0.85 was made by Count Vectorizer using Naive Bayes machine learning algorithm.
- Count Vectorizer + SMOTE and Count Vectorizer + Truncated Support Vector Machines (TSVD) + SMOTE using Naive Bayes also share the same best average F1 score of 0.85.
- The best minority class F1 score of 0.66 was made by Long Short Term Memory (LSTM) Recurrent Neural Network (RNN).
- Word2Vec with machine learning has the lowest average score of 0.81 and minority score of 0.49.





I have seen the effect of imbalanced data throughout the project. However, a more important issue was the huge amount of matching words among the classes. The oversampling solution for imbalanced data didn't work well because of the high rate of matching words. Deleting most common words didn't solve the problem either.

Two possible areas for further improvement could be the need for more data to train neural networks and implementation of Dask library for parallel processing to decrease run time.

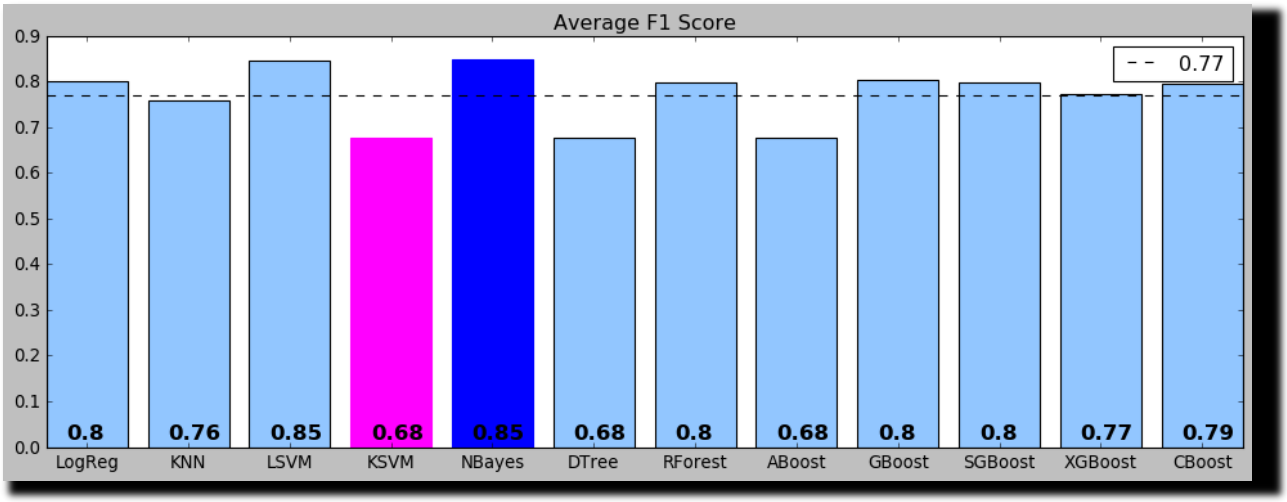
ATTACHMENT: MODEL RESULTS

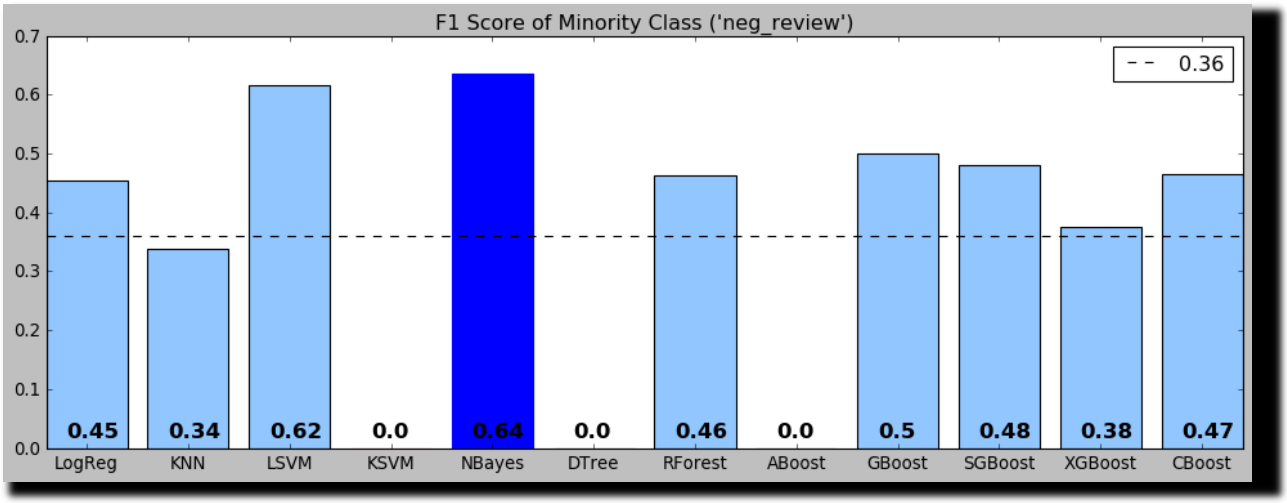
A.1. Modeling with Count-Vectorizing

12 different machine learning algorithms implemented with Count-Vectorizing method. Uni-gram has been used as the best parameter for ngram_range. Accuracy scores and classification report results have been gathered as a comparison table. Best average f-1 scores and minor class f-1 scores of each model have been plotted.

model	accuracy	class	precision	recall	f1-score	support
LogReg	0.830986	neg_review	0.833333	0.3125	0.454545	48
LogReg	0.830986	pos_review	0.830769	0.981818	0.9	165
LogReg	0.830986	average	0.831347	0.830986	0.799616	213
KNN	0.798122	neg_review	0.647059	0.229167	0.338462	48
KNN	0.798122	pos_review	0.811224	0.963636	0.880886	165
KNN	0.798122	average	0.774229	0.798122	0.75865	213
LSVM	0.859155	neg_review	0.8	0.5	0.615385	48
LSVM	0.859155	pos_review	0.868852	0.963636	0.913793	165
LSVM	0.859155	average	0.853336	0.859155	0.846546	213
KSVM	0.774648	neg_review	0	0	0	48
KSVM	0.774648	pos_review	0.774648	1	0.873016	165
KSVM	0.774648	average	0.600079	0.774648	0.67628	213
NBayes	0.85446	neg_review	0.72973	0.5625	0.635294	48
NBayes	0.85446	pos_review	0.880682	0.939394	0.909091	165
NBayes	0.85446	average	0.846664	0.85446	0.84739	213
DTree	0.774648	neg_review	0	0	0	48
DTree	0.774648	pos_review	0.774648	1	0.873016	165
DTree	0.774648	average	0.600079	0.774648	0.67628	213
RForest	0.826291	neg_review	0.761905	0.333333	0.463768	48
RForest	0.826291	pos_review	0.833333	0.969697	0.896359	165
RForest	0.826291	average	0.817237	0.826291	0.798873	213
ABoost	0.774648	neg_review	0	0	0	48
ABoost	0.774648	pos_review	0.774648	1	0.873016	165
ABoost	0.774648	average	0.600079	0.774648	0.67628	213
GBoost	0.821596	neg_review	0.678571	0.395833	0.5	48
GBoost	0.821596	pos_review	0.843243	0.945455	0.891429	165
GBoost	0.821596	average	0.806134	0.821596	0.803219	213
SGBoost	0.816901	neg_review	0.666667	0.375	0.48	48
SGBoost	0.816901	pos_review	0.83871	0.945455	0.888889	165
SGBoost	0.816901	average	0.799939	0.816901	0.796745	213

XGBoost	0.812207	neg_review	0.75	0.25	0.375	48
XGBoost	0.812207	pos_review	0.817259	0.975758	0.889503	165
XGBoost	0.812207	average	0.802102	0.812207	0.773558	213
CBoost	0.816901	neg_review	0.68	0.354167	0.465753	48
CBoost	0.816901	pos_review	0.835106	0.951515	0.889518	165
CBoost	0.816901	average	0.800153	0.816901	0.794022	213





- The best scores with Count Vectorizer, both in average and minority class F1 score was made by Naive Bayes: 0.85 and 0.64 respectively.
- LVSM also has a high average score of 0.85, but the minority score is lower (0.62).

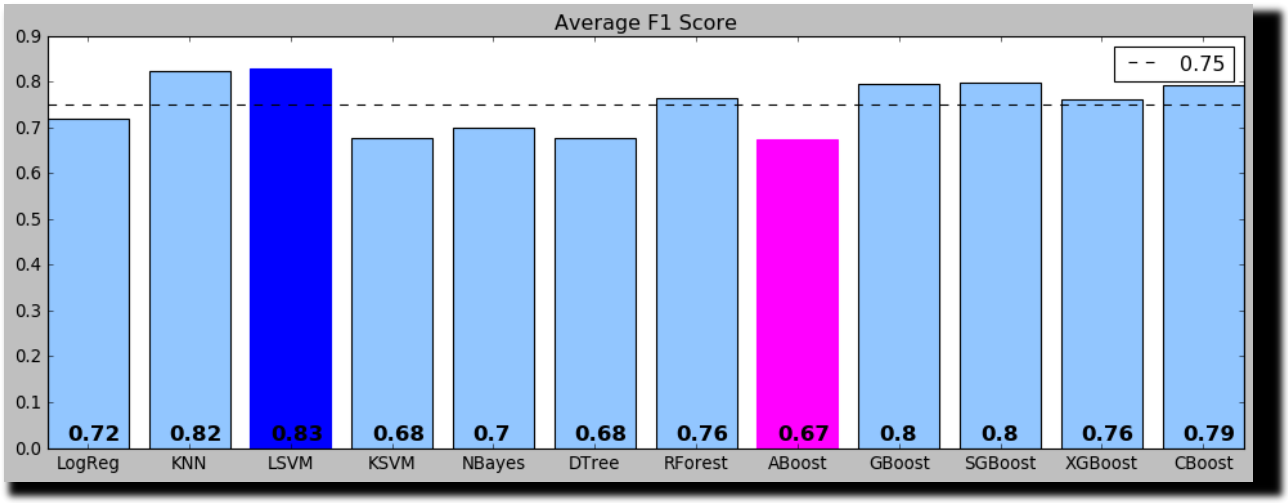
- KSVM, Decision Tree and AdaBoost are the weakest algorithms for Count Vectorizing.

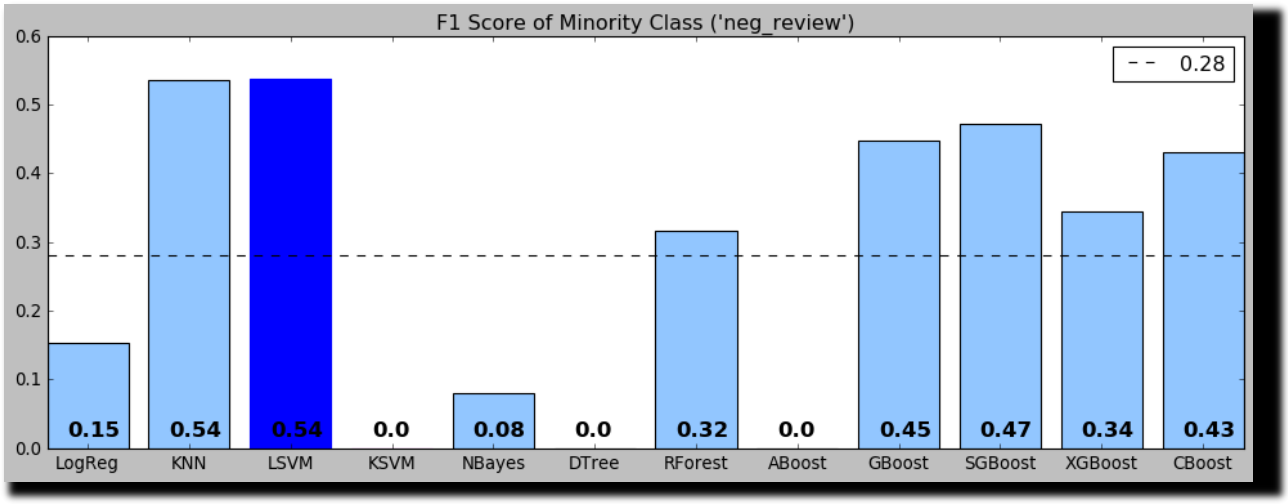
A.2. Modeling with Tfidf – Vectorizing

12 different machine learning algorithms implemented with Tfidf-Vectorizing method. Uni-gram has been used as the best parameter for ngram_range. Accuracy scores and classification report results have been gathered as a comparison table. Best average f-1 scores and minor class f-1 scores of each model have been plotted.

model	accuracy	class	precision	recall	f1-score	support
LogReg	0.793427	neg_review	1	0.083333	0.153846	48
LogReg	0.793427	pos_review	0.789474	1	0.882353	165
LogReg	0.793427	average	0.836916	0.793427	0.718182	213
KNN	0.84507	neg_review	0.826087	0.395833	0.535211	48
KNN	0.84507	pos_review	0.847368	0.975758	0.907042	165
KNN	0.84507	average	0.842573	0.84507	0.823249	213
LSVM	0.85446	neg_review	0.947368	0.375	0.537313	48
LSVM	0.85446	pos_review	0.845361	0.993939	0.913649	165
LSVM	0.85446	average	0.868348	0.85446	0.828841	213
KSVM	0.774648	neg_review	0	0	0	48
KSVM	0.774648	pos_review	0.774648	1	0.873016	165
KSVM	0.774648	average	0.600079	0.774648	0.67628	213
NBayes	0.784038	neg_review	1	0.041667	0.08	48
NBayes	0.784038	pos_review	0.781991	1	0.87766	165
NBayes	0.784038	average	0.831119	0.784038	0.697905	213
DTree	0.774648	neg_review	0	0	0	48
DTree	0.774648	pos_review	0.774648	1	0.873016	165
DTree	0.774648	average	0.600079	0.774648	0.67628	213
RForest	0.816901	neg_review	1	0.1875	0.315789	48
RForest	0.816901	pos_review	0.808824	1	0.894309	165
RForest	0.816901	average	0.851906	0.816901	0.763938	213
ABoost	0.769953	neg_review	0	0	0	48
ABoost	0.769953	pos_review	0.773585	0.993939	0.870027	165
ABoost	0.769953	average	0.599256	0.769953	0.673964	213
GBoost	0.826291	neg_review	0.789474	0.3125	0.447761	48
GBoost	0.826291	pos_review	0.829897	0.975758	0.896936	165
GBoost	0.826291	average	0.820787	0.826291	0.795713	213
SGBoost	0.821596	neg_review	0.708333	0.354167	0.472222	48
SGBoost	0.821596	pos_review	0.835979	0.957576	0.892655	165

SGBoost	0.821596	average	0.807214	0.821596	0.79791	213
XGBoost	0.802817	neg_review	0.6875	0.229167	0.34375	48
XGBoost	0.802817	pos_review	0.812183	0.969697	0.883978	165
XGBoost	0.802817	average	0.784085	0.802817	0.762236	213
CBoost	0.826291	neg_review	0.823529	0.291667	0.430769	48
CBoost	0.826291	pos_review	0.826531	0.981818	0.897507	165
CBoost	0.826291	average	0.825854	0.826291	0.792327	213





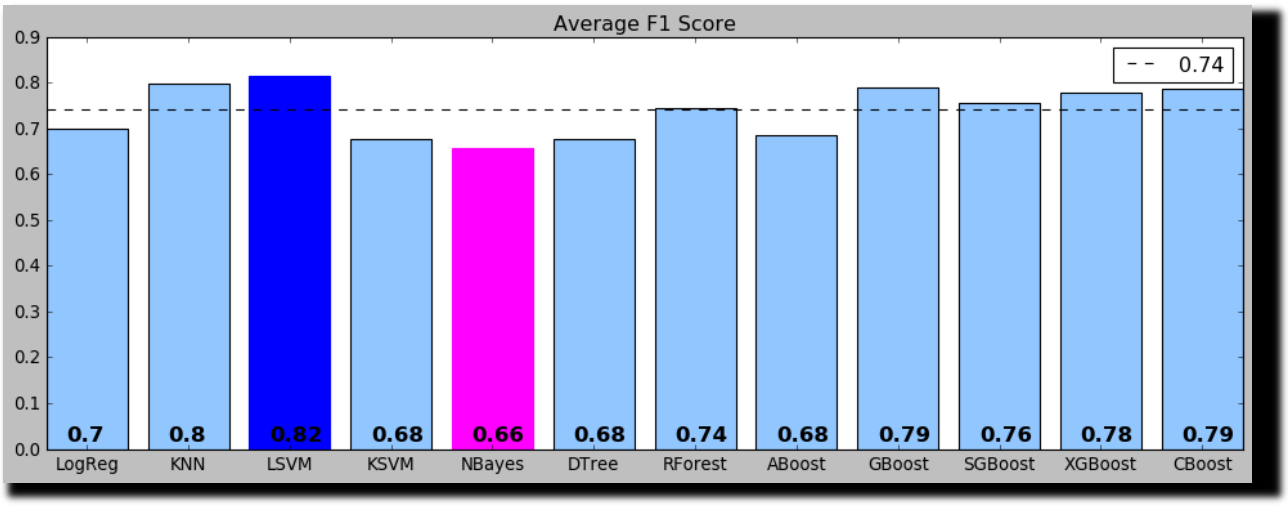
- The best scores with Tfidf Vectorizer, both in average and minority class F1 score was made by LSVM: 0.83 and 0.54 respectively.
- AdaBoost is the weakest algorithms with Tfidf-vectorizing.
- KSVM, Decision Tree and AdaBoost share the lowest minority score of 0.

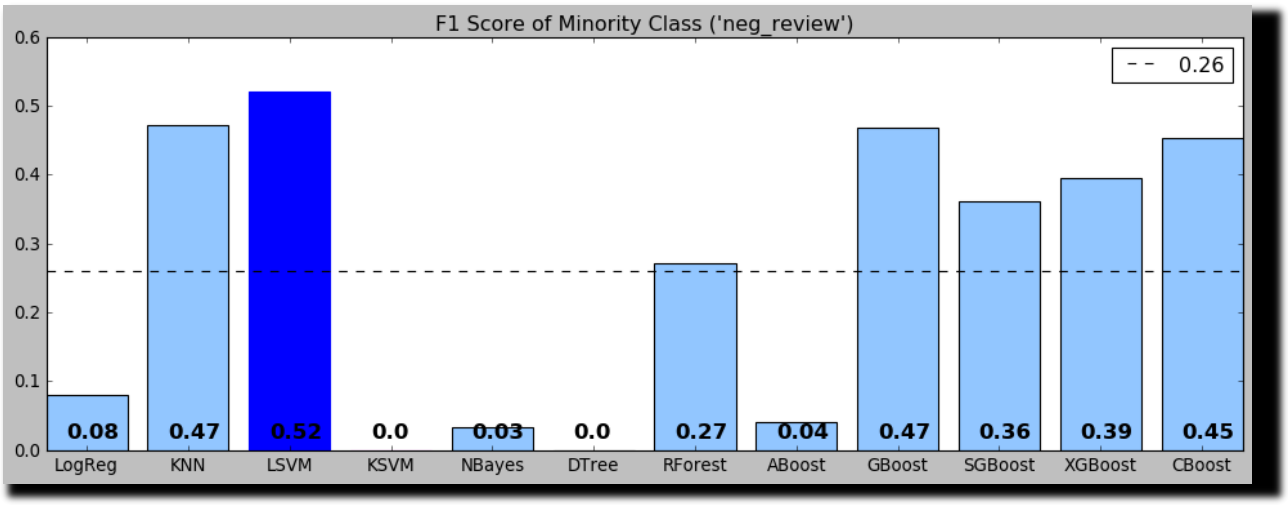
A.3. Modeling with Hashing-Vectorizing

12 different machine learning algorithms implemented with Hashing-Vectorizing method. Unigram has been used as the best parameter for ngram_range. Accuracy scores and classification report results have been gathered as a comparison table. Best average f-1 scores and minor class f-1 scores of each model have been plotted.

model	accuracy	class	precision	recall	f1-score	support
LogReg	0.784038	neg_review	1	0.041667	0.08	48
LogReg	0.784038	pos_review	0.781991	1	0.87766	165
LogReg	0.784038	average	0.831119	0.784038	0.697905	213
KNN	0.821596	neg_review	0.708333	0.354167	0.472222	48
KNN	0.821596	pos_review	0.835979	0.957576	0.892655	165
KNN	0.821596	average	0.807214	0.821596	0.79791	213
LSVM	0.835681	neg_review	0.76	0.395833	0.520548	48
LSVM	0.835681	pos_review	0.845745	0.963636	0.90085	165
LSVM	0.835681	average	0.826422	0.835681	0.815148	213
KSVM	0.774648	neg_review	0	0	0	48
KSVM	0.774648	pos_review	0.774648	1	0.873016	165
KSVM	0.774648	average	0.600079	0.774648	0.67628	213
NBayes	0.723005	neg_review	0.076923	0.020833	0.032787	48
NBayes	0.723005	pos_review	0.765	0.927273	0.838356	165
NBayes	0.723005	average	0.60994	0.723005	0.656819	213
DTree	0.774648	neg_review	0	0	0	48
DTree	0.774648	pos_review	0.774648	1	0.873016	165
DTree	0.774648	average	0.600079	0.774648	0.67628	213
RForest	0.798122	neg_review	0.727273	0.166667	0.271186	48
RForest	0.798122	pos_review	0.80198	0.981818	0.882834	165
RForest	0.798122	average	0.785145	0.798122	0.744998	213
ABoost	0.774648	neg_review	0.5	0.020833	0.04	48
ABoost	0.774648	pos_review	0.777251	0.993939	0.87234	165
ABoost	0.774648	average	0.714772	0.774648	0.684771	213
GBoost	0.807512	neg_review	0.62069	0.375	0.467532	48
GBoost	0.807512	pos_review	0.836957	0.933333	0.882521	165
GBoost	0.807512	average	0.78822	0.807512	0.789003	213
SGBoost	0.784038	neg_review	0.541667	0.270833	0.361111	48
SGBoost	0.784038	pos_review	0.814815	0.933333	0.870056	165
SGBoost	0.784038	average	0.75326	0.784038	0.755365	213
XGBoost	0.812207	neg_review	0.722222	0.270833	0.393939	48

XGBoost	0.812207	pos_review	0.820513	0.969697	0.888889	165
XGBoost	0.812207	average	0.798363	0.812207	0.777351	213
CBoost	0.807512	neg_review	0.62963	0.354167	0.453333	48
CBoost	0.807512	pos_review	0.833333	0.939394	0.883191	165
CBoost	0.807512	average	0.787428	0.807512	0.786322	213





- The best scores with Hashing Vectorizer, both in average and minority class F1 score was made by LSVM: 0.82 and 0.52 respectively.
- Naïve Bayes is the weakest algorithm with hashing-vectorizing.
- Among the Bag of Words vectorizers, best scores were made with Count Vectorizer by Naïve Bayes.
- Tfidf and Hashing Vectorizers made very close scores via the LSVM algorithm.

- Count Vectorizer minority class F1 score with Naive Bayes was relatively high (0.64) by comparison to Tfidf (0.54) and Hashing (0.52) Vectorizer scores both with the LSVM machine learning algorithm.

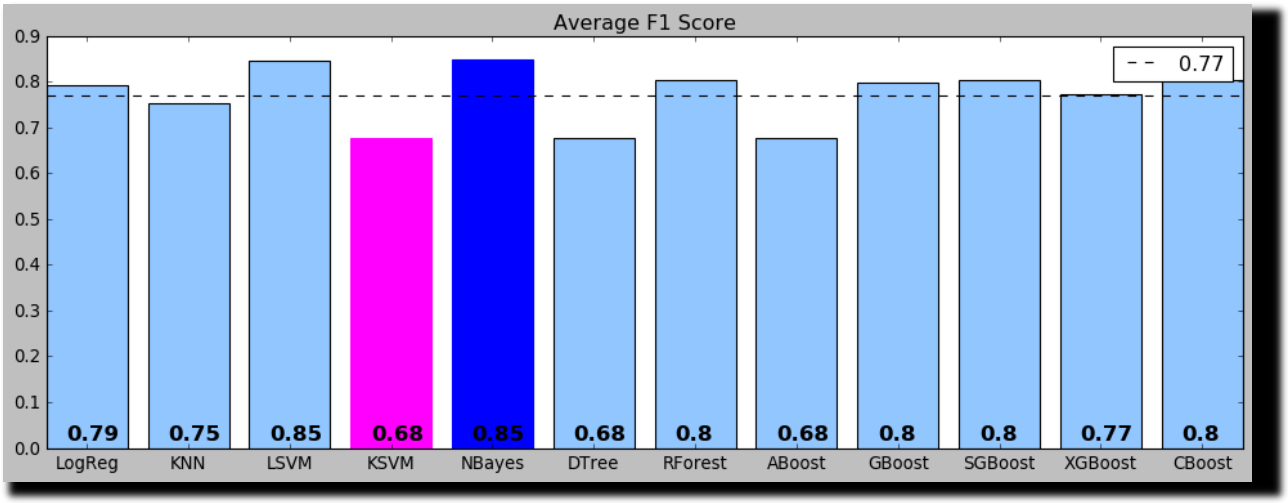
Based on the scores, I will use Count Vectorizer as the Word Vectorizer for further analysis.

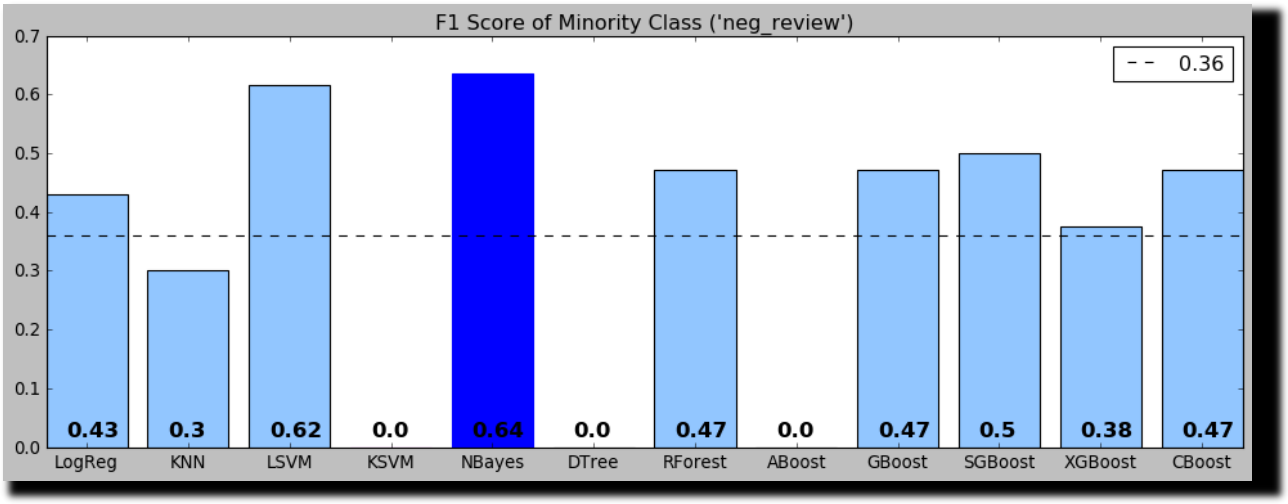
A.4. Modeling with SMOTE

12 different machine learning algorithms implemented with SMOTE method. Since I got the best results with, Count-vectorizing based features were used.. Accuracy scores and classification report results have been gathered as a comparison table. Best average f-1 scores and minor class f-1 scores of each model have been plotted.

model	accuracy	class	precision	recall	f1-score	support
LogReg	0.826291	neg_review	0.823529	0.291667	0.430769	48
LogReg	0.826291	pos_review	0.826531	0.981818	0.897507	165
LogReg	0.826291	average	0.825854	0.826291	0.792327	213
KNN	0.802817	neg_review	0.75	0.1875	0.3	48
KNN	0.802817	pos_review	0.80597	0.981818	0.885246	165
KNN	0.802817	average	0.793357	0.802817	0.75336	213
LSVM	0.859155	neg_review	0.8	0.5	0.615385	48
LSVM	0.859155	pos_review	0.868852	0.963636	0.913793	165
LSVM	0.859155	average	0.853336	0.859155	0.846546	213
KSVM	0.774648	neg_review	0	0	0	48
KSVM	0.774648	pos_review	0.774648	1	0.873016	165
KSVM	0.774648	average	0.600079	0.774648	0.67628	213
NBayes	0.85446	neg_review	0.72973	0.5625	0.635294	48
NBayes	0.85446	pos_review	0.880682	0.939394	0.909091	165
NBayes	0.85446	average	0.846664	0.85446	0.84739	213
DTree	0.774648	neg_review	0	0	0	48
DTree	0.774648	pos_review	0.774648	1	0.873016	165
DTree	0.774648	average	0.600079	0.774648	0.67628	213
RForest	0.830986	neg_review	0.8	0.333333	0.470588	48
RForest	0.830986	pos_review	0.834197	0.975758	0.899441	165
RForest	0.830986	average	0.826491	0.830986	0.802798	213
ABoost	0.774648	neg_review	0	0	0	48
ABoost	0.774648	pos_review	0.774648	1	0.873016	165
ABoost	0.774648	average	0.600079	0.774648	0.67628	213
GBoost	0.821596	neg_review	0.708333	0.354167	0.472222	48
GBoost	0.821596	pos_review	0.835979	0.957576	0.892655	165
GBoost	0.821596	average	0.807214	0.821596	0.79791	213
SGBoost	0.821596	neg_review	0.678571	0.395833	0.5	48

SGBoost	0.821596	pos_review	0.843243	0.945455	0.891429	165
SGBoost	0.821596	average	0.806134	0.821596	0.803219	213
XGBoost	0.812207	neg_review	0.75	0.25	0.375	48
XGBoost	0.812207	pos_review	0.817259	0.975758	0.889503	165
XGBoost	0.812207	average	0.802102	0.812207	0.773558	213
CBoost	0.830986	neg_review	0.8	0.333333	0.470588	48
CBoost	0.830986	pos_review	0.834197	0.975758	0.899441	165
CBoost	0.830986	average	0.826491	0.830986	0.802798	213





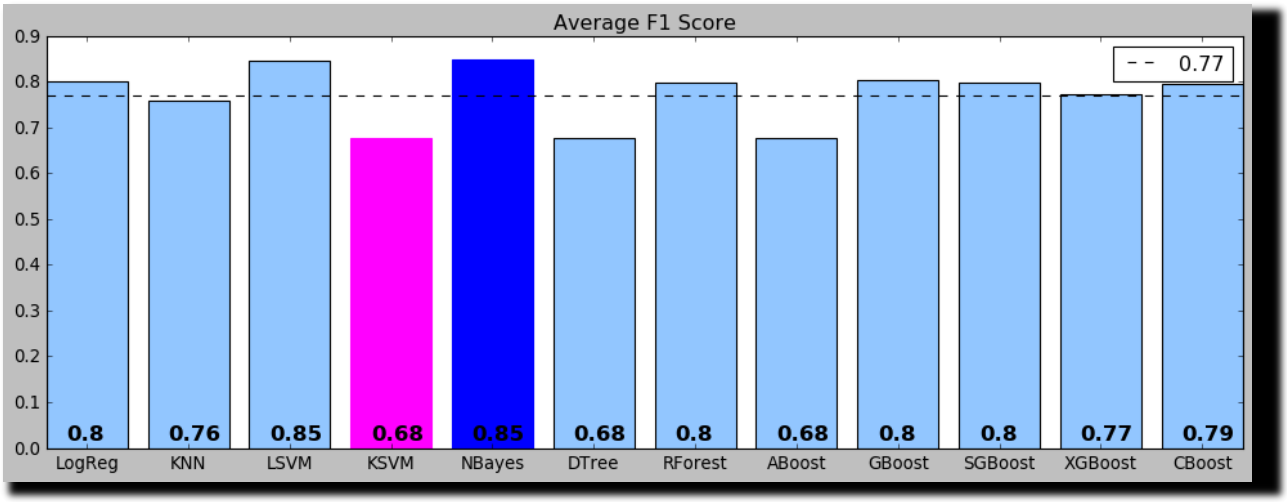
- The best scores using SMOTE with Count Vectorizer, both in average and minority class F1 score was made by Naive Bayes: 0.85 and 0.64 respectively.
- LVSM also has a high average score of 0.85, but the minority score is lower (0.62).
- KSVM, Decision Tree and AdaBoost share the lowest average score of 0.68 and minority score of 0.

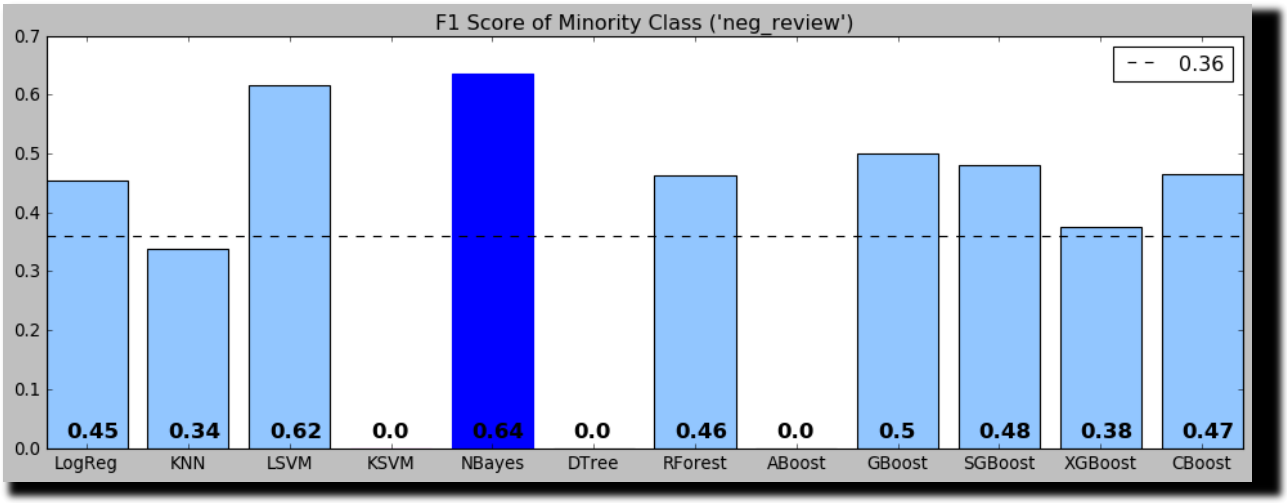
A.5. Modeling with PCA-SMOTE Combination

12 different machine learning algorithms implemented with PCA-SMOTE combination method. Since we got the best results with, Count-vectorizing based features were used for this combination. Accuracy scores and classification report results have been gathered as a comparison table. Best average f-1 scores and minor class f-1 scores of each model have been plotted.

model	accuracy	class	precision	recall	f1-score	support
LogReg	0.830986	neg_review	0.833333	0.3125	0.454545	48
LogReg	0.830986	pos_review	0.830769	0.981818	0.9	165
LogReg	0.830986	average	0.831347	0.830986	0.799616	213
KNN	0.798122	neg_review	0.647059	0.229167	0.338462	48
KNN	0.798122	pos_review	0.811224	0.963636	0.880886	165
KNN	0.798122	average	0.774229	0.798122	0.75865	213
LSVM	0.859155	neg_review	0.8	0.5	0.615385	48
LSVM	0.859155	pos_review	0.868852	0.963636	0.913793	165
LSVM	0.859155	average	0.853336	0.859155	0.846546	213
KSVM	0.774648	neg_review	0	0	0	48
KSVM	0.774648	pos_review	0.774648	1	0.873016	165
KSVM	0.774648	average	0.600079	0.774648	0.67628	213
NBayes	0.85446	neg_review	0.72973	0.5625	0.635294	48
NBayes	0.85446	pos_review	0.880682	0.939394	0.909091	165
NBayes	0.85446	average	0.846664	0.85446	0.84739	213
DTree	0.774648	neg_review	0	0	0	48
DTree	0.774648	pos_review	0.774648	1	0.873016	165
DTree	0.774648	average	0.600079	0.774648	0.67628	213
RForest	0.826291	neg_review	0.761905	0.333333	0.463768	48
RForest	0.826291	pos_review	0.833333	0.969697	0.896359	165
RForest	0.826291	average	0.817237	0.826291	0.798873	213
ABoost	0.774648	neg_review	0	0	0	48
ABoost	0.774648	pos_review	0.774648	1	0.873016	165
ABoost	0.774648	average	0.600079	0.774648	0.67628	213
GBoost	0.821596	neg_review	0.678571	0.395833	0.5	48
GBoost	0.821596	pos_review	0.843243	0.945455	0.891429	165
GBoost	0.821596	average	0.806134	0.821596	0.803219	213
SGBoost	0.816901	neg_review	0.666667	0.375	0.48	48
SGBoost	0.816901	pos_review	0.83871	0.945455	0.888889	165
SGBoost	0.816901	average	0.799939	0.816901	0.796745	213

XGBoost	0.812207	neg_review	0.75	0.25	0.375	48
XGBoost	0.812207	pos_review	0.817259	0.975758	0.889503	165
XGBoost	0.812207	average	0.802102	0.812207	0.773558	213
CBoost	0.816901	neg_review	0.68	0.354167	0.465753	48
CBoost	0.816901	pos_review	0.835106	0.951515	0.889518	165
CBoost	0.816901	average	0.800153	0.816901	0.794022	213





- The best scores using PCA+SMOTE with Count Vectorizer, both in average and minority class F1 score was made by Naive Bayes: 0.85 and 0.64 respectively.
- KSVM, Decision Tree and AdaBoost share the lowest average score of 0.68 and minority score of 0.

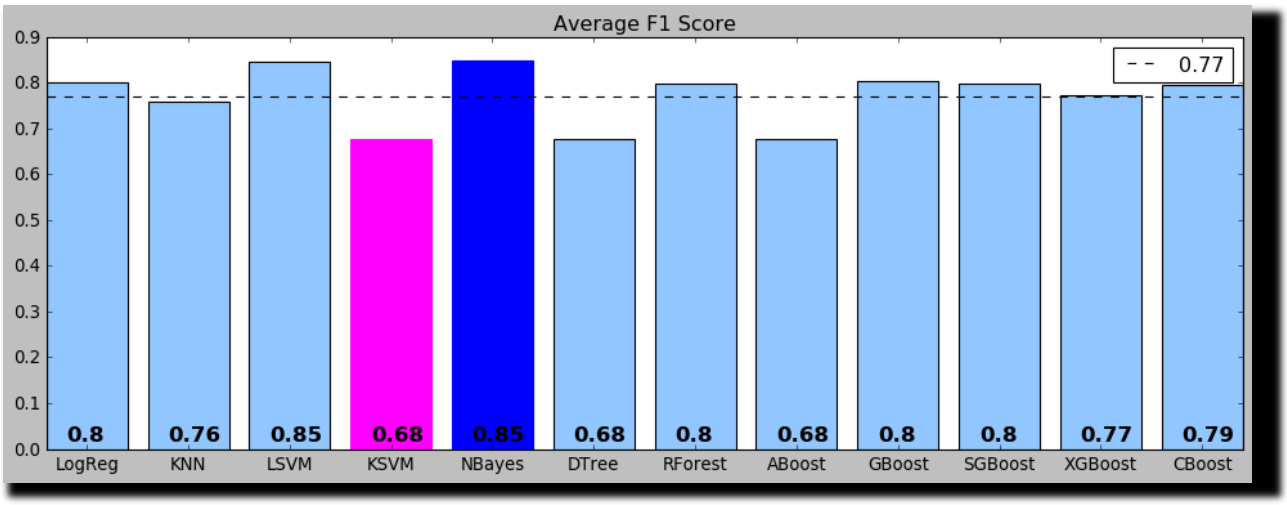
A.6. Modeling with Truncated SVD – SMOTE Combination

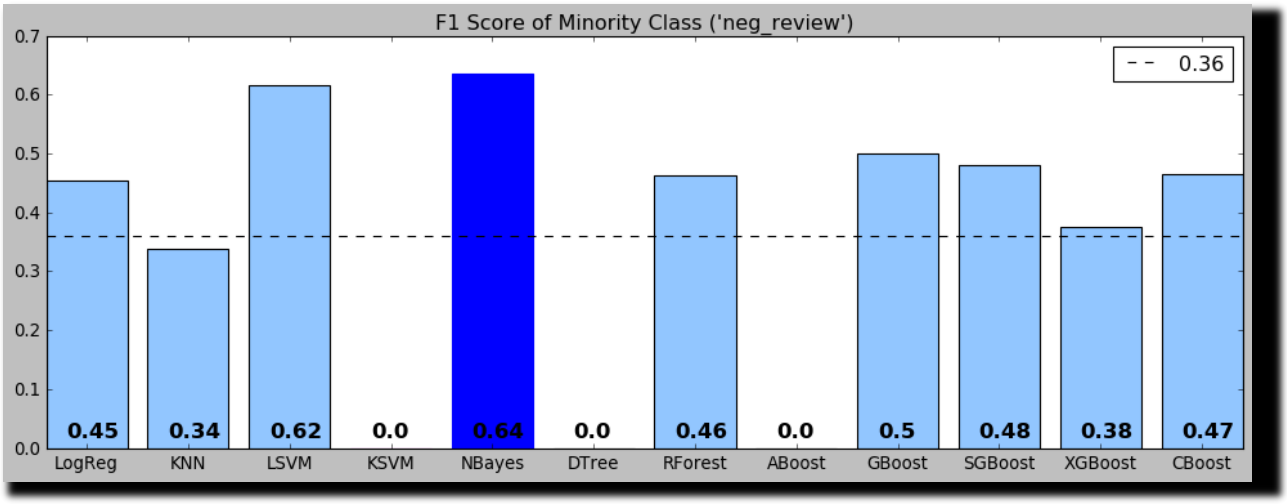
12 different machine learning algorithms implemented with Truncated SVD - SMOTE combination method. Since we got the best results with, Count-vectorizing based features were used for this combination. Accuracy scores and classification report results have been gathered as a comparison table. Best average f-1 scores and minor class f-1 scores of each model have been plotted.

model	accuracy	class	precision	recall	f1-score	support
LogReg	0.830986	neg_review	0.833333	0.3125	0.454545	48
LogReg	0.830986	pos_review	0.830769	0.981818	0.9	165
LogReg	0.830986	average	0.831347	0.830986	0.799616	213
KNN	0.798122	neg_review	0.647059	0.229167	0.338462	48
KNN	0.798122	pos_review	0.811224	0.963636	0.880886	165
KNN	0.798122	average	0.774229	0.798122	0.75865	213
LSVM	0.859155	neg_review	0.8	0.5	0.615385	48
*LSVM	0.859155	pos_review	0.868852	0.963636	0.913793	165
LSVM	0.859155	average	0.853336	0.859155	0.846546	213
KSVM	0.774648	neg_review	0	0	0	48
KSVM	0.774648	pos_review	0.774648	1	0.873016	165
KSVM	0.774648	average	0.600079	0.774648	0.67628	213
NBayes	0.85446	neg_review	0.72973	0.5625	0.635294	48
NBayes	0.85446	pos_review	0.880682	0.939394	0.909091	165
NBayes	0.85446	average	0.846664	0.85446	0.84739	213
DTree	0.774648	neg_review	0	0	0	48
DTree	0.774648	pos_review	0.774648	1	0.873016	165
DTree	0.774648	average	0.600079	0.774648	0.67628	213
RForest	0.826291	neg_review	0.761905	0.333333	0.463768	48
RForest	0.826291	pos_review	0.833333	0.969697	0.896359	165
RForest	0.826291	average	0.817237	0.826291	0.798873	213
ABoost	0.774648	neg_review	0	0	0	48
ABoost	0.774648	pos_review	0.774648	1	0.873016	165
ABoost	0.774648	average	0.600079	0.774648	0.67628	213
GBoost	0.821596	neg_review	0.678571	0.395833	0.5	48
GBoost	0.821596	pos_review	0.843243	0.945455	0.891429	165
GBoost	0.821596	average	0.806134	0.821596	0.803219	213
SGBoost	0.816901	neg_review	0.666667	0.375	0.48	48
SGBoost	0.816901	pos_review	0.83871	0.945455	0.888889	165
SGBoost	0.816901	average	0.799939	0.816901	0.796745	213
XGBoost	0.812207	neg_review	0.75	0.25	0.375	48

XGBoost	0.812207	pos_review	0.817259	0.975758	0.889503	165
- XGBoost	0.812207	average	0.802102	0.812207	0.773558	213
CBoost	0.816901	neg_review	0.68	0.354167	0.465753	48
CBoost	0.816901	pos_review	0.835106	0.951515	0.889518	165
CBoost	0.816901	average	0.800153	0.816901	0.794022	213

-





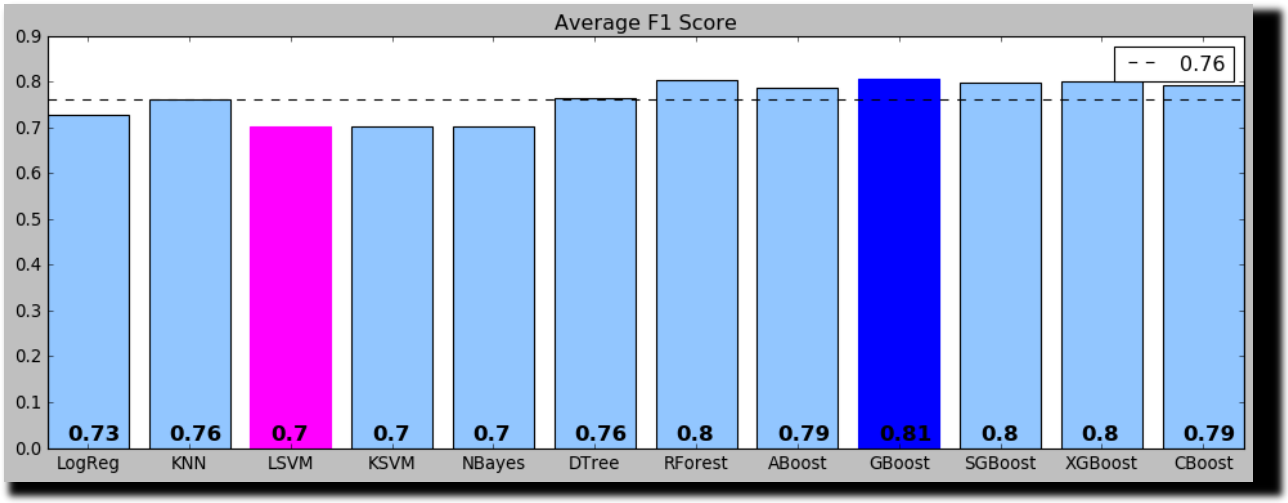
- The best scores using Truncated SVD and SMOTE with Count Vectorizer, both in average and minority class F1 score was made by Naive Bayes: 0.85 and 0.64 respectively.
- LSVM also has a high average score of 0.85, but the minority score is lower (0.62).
- KSVM, Decision Tree and AdaBoost share the lowest average score of 0.68 and minority score of 0.

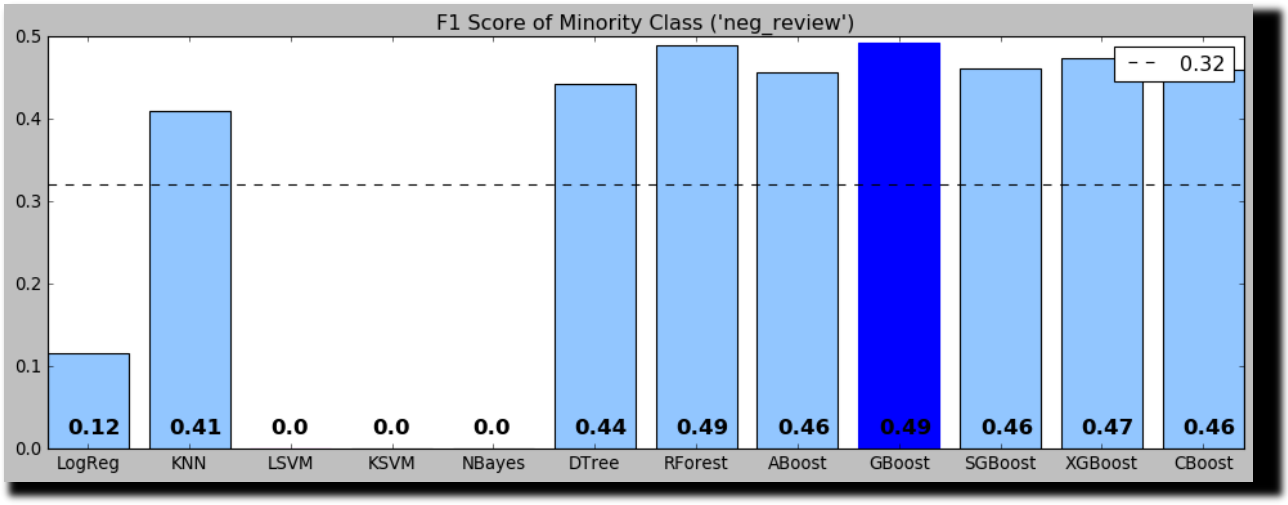
A.7. Modeling with Word2Vec

12 different machine learning algorithms implemented with Word2Vec method. Accuracy scores and classification report results have been gathered as a comparison table. Best average f-1 scores and minor class f-1 scores of each model have been plotted.

model	accuracy	class	precision	recall	f1-score	support
LogReg	0.79854	neg_review	0.6	0.06383	0.115385	141
LogReg	0.79854	pos_review	0.802985	0.988971	0.886326	544
LogReg	0.79854	average	0.761203	0.79854	0.727636	685
KNN	0.763504	neg_review	0.421053	0.397163	0.408759	141
KNN	0.763504	pos_review	0.846014	0.858456	0.85219	544
KNN	0.763504	average	0.758541	0.763504	0.760914	685
LSVM	0.794161	neg_review	0	0	0	141
LSVM	0.794161	pos_review	0.794161	1	0.885273	544
LSVM	0.794161	average	0.630691	0.794161	0.703049	685
KSVM	0.794161	neg_review	0	0	0	141
KSVM	0.794161	pos_review	0.794161	1	0.885273	544
KSVM	0.794161	average	0.630691	0.794161	0.703049	685
NBayes	0.794161	neg_review	0	0	0	141
NBayes	0.794161	pos_review	0.794161	1	0.885273	544
NBayes	0.794161	average	0.630691	0.794161	0.703049	685
DTree	0.760584	neg_review	0.424837	0.460993	0.442177	141
DTree	0.760584	pos_review	0.857143	0.838235	0.847584	544
DTree	0.760584	average	0.768157	0.760584	0.764135	685
RForest	0.810219	neg_review	0.548673	0.439716	0.488189	141
RForest	0.810219	pos_review	0.861888	0.90625	0.883513	544
RForest	0.810219	average	0.797416	0.810219	0.802139	685
ABoost	0.794161	neg_review	0.5	0.41844	0.455598	141
ABoost	0.794161	pos_review	0.855379	0.891544	0.873087	544
ABoost	0.794161	average	0.782228	0.794161	0.787152	685
GBoost	0.816058	neg_review	0.570093	0.432624	0.491935	141
GBoost	0.816058	pos_review	0.861592	0.915441	0.887701	544
GBoost	0.816058	average	0.80159	0.816058	0.806236	685
SGBoost	0.808759	neg_review	0.54902	0.397163	0.460905	141
SGBoost	0.808759	pos_review	0.854202	0.915441	0.883762	544
SGBoost	0.808759	average	0.791384	0.808759	0.796722	685
XGBoost	0.811679	neg_review	0.557692	0.411348	0.473469	141
XGBoost	0.811679	pos_review	0.857143	0.915441	0.885333	544

XGBoost	0.811679	average	0.795504	0.811679	0.800555	685
CBoost	0.8	neg_review	0.517857	0.411348	0.458498	141
CBoost	0.8	pos_review	0.855148	0.900735	0.87735	544
CBoost	0.8	average	0.785721	0.8	0.791134	685





- The best scores using Word2Vec with machine learning models, both in average and minority class F1 score was made by Gradient Boosting: 0.81 and 0.49 respectively.
- Random Forest also has the minority score of 0.49, but the average score is lower (0.8).
- LSVM, KSVM and Naive Bayes share the lowest average score of 0.7 and minority score of 0.

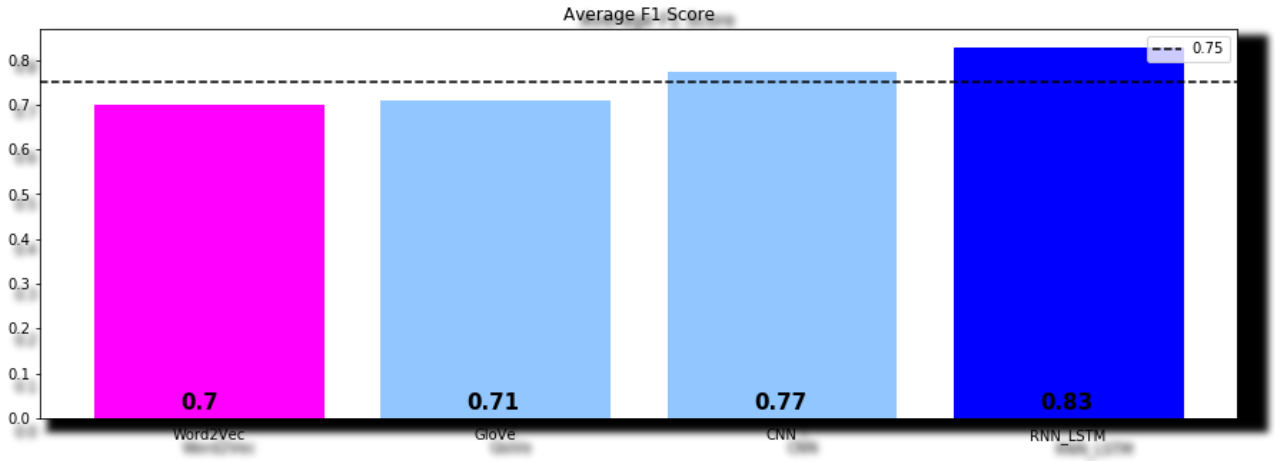
A.9. Deep Learning Models with Word2Vec, GloVe, CNN, RNN LSTM

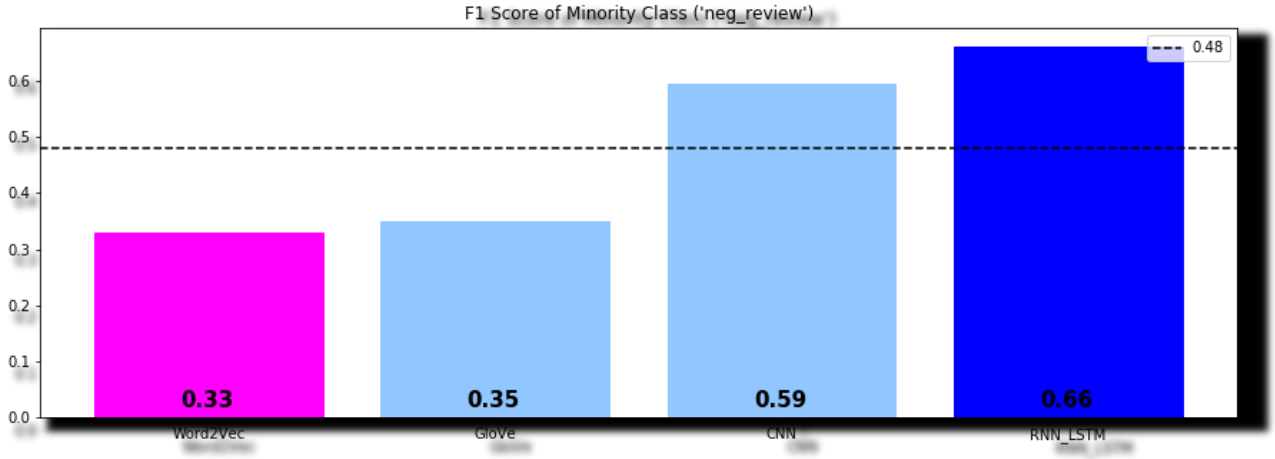
For Word2Vec and GloVe, I used a fully-connected four layer deep neural network with three hidden layers of 1000 neurons or units and one output layer with two units that was used to either predict a positive (good) or negative (bad) sentiment based on the input layer features.

As a Convolutional Neural Network (CNN), I utilized 1D convolutions to scan through the sentences. The model first transformed each word into lower dimensional embedding/vector space followed by 1d convolutions and then passing the data through a dense layer with 1000 neurons before the final layer for classification.

As a Recurrent Neural Network (RNN), I used an embedding layer with an output dimension of 1280 to generate dense word embeddings which were then passed to the LSTM layer having 640 units.

model	accuracy	class	precision	recall	f1-score	support
Word2Vec	0.737643	neg_review	0.53125	0.239437	0.330097	71
Word2Vec	0.737643	pos_review	0.766234	0.921875	0.836879	192
Word2Vec	0.737643	average	0.702797	0.737643	0.700067	263
GloVe	0.745247	neg_review	0.5625	0.253521	0.349515	71
GloVe	0.745247	pos_review	0.770563	0.927083	0.841608	192
GloVe	0.745247	average	0.714394	0.745247	0.708761	263
CNN	0.771863	neg_review	0.571429	0.619718	0.594595	71
CNN	0.771863	pos_review	0.854839	0.828125	0.84127	192
CNN	0.771863	average	0.778329	0.771863	0.774677	263
RNN_LSTM	0.8327	neg_review	0.728814	0.605634	0.661538	71
RNN_LSTM	0.8327	pos_review	0.862745	0.916667	0.888889	192
RNN_LSTM	0.8327	average	0.826589	0.8327	0.827513	263





- The best scores using deep learning models with Keras, both in average and minority class F1 score was made by RNN_LSTM: 0.83 and 0.66 respectively.

- Word2Vec has the lowest average score of 0.7 and minority score of 0.33.