

## 1. Preprocessing the textual data

For the purposes of this course work, the amazon\_cells\_labelled data set was selected.

This dataset was created for the paper “From Group to Individual Labels Using Deep Features” by Kotsias et al. KDD 2015. Reviews taken from Amazon.com. These were randomly selected from larger sets of reviews. Reviews have a clearly positive or negative meaning, and there is not a single neutral sentence.

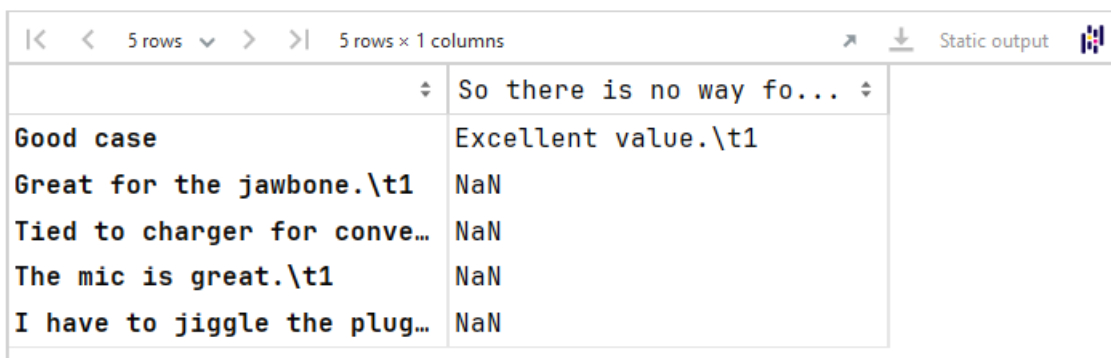
The dataset contains positive and negative sentiment sentences taken from product, movie, and restaurant reviews.

It was not possible to load the data set using the standard method with default settings because the set contained errors within the set. Therefore, the error skipping option was applied:

- `reviews = pd.read_csv ("amazon_cells_labelled.txt", on_bad_lines = ' skip ')`

`on_bad_lines = 'skip'` parameter of the `pd.read_csv` command allowed us to skip lines in the file that could not be read correctly, or lines that contained errors.

Instead of throwing an error and stopping the process, the command skipped the problematic lines and continued reading the rest of the file.



	So there is no way fo...
Good case	Excellent value.\t1
Great for the jawbone.\t1	NaN
Tied to charger for conve...	NaN
The mic is great.\t1	NaN
I have to jiggle the plug...	NaN

After the file was loaded, it was discovered that the data in the DataFrame was not formatted correctly, and the reviews were not divided into different columns. Also, it looks like there are missing values (NaN) and unnecessary text. To pre-process the data and perform text analysis, it was decided that the DataFrame needed to be cleaned and organized.

Assuming the 'Review' and ' Score ' columns are tab delimited, the `read_csv` command call was modified to read as follows:

- `reviews = pd.read_csv ("amazon_cells_labelled.txt", sep='\t', header=None , names=['Review', ' Score '])`

After this, missing lines were removed from the data set values (NaN). This step provided clean kit data for analysis, eliminating any potential problems associated with missing information.

10 rows		10 rows × 2 columns		Static output
	Review		Score	
0	So there is no way for me to plug it in here i...		0	
1	Good case, Excellent value.		1	
2	Great for the jawbone.		1	
3	Tied to charger for conversations lasting more...		0	
4	The mic is great.		1	
5	I have to jiggle the plug to get it to line up...		0	
6	If you have several dozen or several hundred c...		0	
7	If you are Razr owner...you must have this!		1	
8	Needless to say, I wasted my money.		0	
9	What a waste of money and time!.		0	

After these steps DataFrame was right formatted and the opportunity has appeared proceed with further preliminary processing and analysis.

Thus, after formatting the data set, it was found that DataFrame consists of 1000 instances and two features 'Review' and 'Score'. The 'Review' column contains 1000 reviews, consisting out of 500 positive and 500 negative reviews. All reviews labelled 0 and 1, where 0 are negative reviews and 1 are positive reviews.

```
Score
0    500
1    500
Name: count, dtype: int64
```

Type data 'Review' column changed on string. This designed to provide consistency types data and corresponds assumption that \_ reviews present yourself text data.

The following stages of text pre-processing were applied:

1. Remove Punctuation
2. Remove Numbers
3. Convert to Lowercase
4. Remove Stop Words
5. Spell Correction
6. Remove Non-Alphabetic Words
7. Remove Accented Characters
8. Lemmatizing with POS Tags

Since the text is contained in only one column of the data set, all the text was copied into a separate variable.

At the beginning, punctuation marks were removed from the text using the command:

- **review = review.translate ( str.maketrans ("", "", string.punctuation ) )**

Purpose this specific method is cleaning text data by removal signs punctuation. Signs punctuation (for example, commas, periods, interrogatives signs, etc.) often Not have values or can bring in interference at analysis text data. Their deletion will help simplify text by doing his more suitable for tasks processing natural language.

For example, the original text 'Good case, Excellent value.' after removal signs punctuation becomes 'Good case Excellent value'.

#### 1. Remove Punctuation:

```
0 So there is no way for me to plug it in here i...
1 Good case Excellent value
2 Great for the jawbone
3 Tied to charger for conversations lasting more...
4 The mic is great
5 I have to jiggle the plug to get it to line up...
6 If you have several dozen or several hundred c...
7 If you are Razr owneryou must have this
8 Needless to say I wasted my money
9 What a waste of money and time
dtype: object
```

The lowercase conversion method converts to es text to bottom register. This step provides uniformity text and allows to avoid potential problems associated with case sensitivity. For example: 'The mic is great' becomes 'the mic is great'.

#### 3. Convert to Lowercase:

```
0 so there is no way for me to plug it in here i...
1 good case excellent value
2 great for the jawbone
3 tied to charger for conversations lasting more...
4 the mic is great
5 i have to jiggle the plug to get it to line up...
6 if you have several dozen or several hundred c...
7 if you are razr owneryou must have this
8 needless to say i wasted my money
9 what a waste of money and time
dtype: object
```

Main target transformation text to bottom register - provide uniformity and consistency text data. This step serves several goals.

Reformation total text to bottom register helps normalize text data. This guarantees that the same one's same words, whatever from their original spelling with capitals letters used to blow be considered identical. Normalization It has decisive meaning for provision consistency text representations and prevention duplication or various definitions words.

Conversion total text to bottom register simplifies process tokenization, breaking text into individual words or tokens. Without differences registers tokenization becomes simpler that leads to more clean and manageable recruitment words.

Having done register uniform, models can concentrate on internal semantics words and improve generalization.

Without normalization register one and the same word with different capitalization will be considered how separate essence that potentially will lead to an increase size dictionary.

One of the significant text pre-processing processes was Lemmatization.

Lemmatization is a text normalization technique that aims to reduce words to their basic or root form, known as lemmas. Unlike stemming, which removes prefixes or suffixes to obtain the stem of a word, lemmatization takes into account the context of the word and its part of speech. The main purpose of lemmatization is to convert words into their canonical forms, making it easier to analyze and interpret textual data.

#### 8. Lemmatizing with POS Tags:

```
0           way plug u unless go convert
1           good case excellent value
2           great jawbone
3 tie charge conversation last minutesmajor problem
4           mid great
5           jingle plug get line right get decent volume
6 several dozen several hundred contact imagine ...
7           razor owneryou must
8           needle say waste money
9           waste money time
dtype: object
```

The advantages of lemmatization over stemming are that:

- Lemmatization takes into account the grammatical context of words by including part of speech information.
- stores information about parts of speech.
- lemmatized words are often more recognizable and interpretable than their stemmed counterparts.
- Lemmatization, especially when combined with POS tags, provides a more linguistically sound representation of text.

## 2. Classification using the bag-of-words / terms

To classify text data using the bag-of-words / terms representation the following algorithms were selected:

1. *Random Forest Classifier*
2. *Gradient Boosting Classifier*
3. *Logistic Regression*
4. *Decision Tree*

Target is to estimate efficiency these algorithms at classifications text on positive or negative feedback.

Text data were converted to the bag - of-words/terms representation using CountVectorizer. This method transforms text to numbers vectors, fixing frequency words in each document. Received representation BoW then used as input data for algorithms classifications .

### 1. *Random Forest Classifier*

RandomForestClassifier demonstrates good performance with 79% accuracy. Classifier demonstrates balanced precision, recall and f1-scores as for positive and for negative 'Reviews'. Matrix confusion emphasizes ability models right to identify true positive and true negative results as well indicates cases false positive and false negative results.

	precision	recall	f1-score	support
Positive	0.77	0.80	0.79	148
Negative	0.80	0.77	0.79	152
accuracy			0.79	300
macro avg	0.79	0.79	0.79	300
weighted avg	0.79	0.79	0.79	300

roc\_auc\_score for DT: 0.8789784850640114

## 2. Gradient Boosting Classifier

Gradient Boosting classifier demonstrates commendable performance with 78.33% accuracy. It demonstrates balanced precision, recall and f1-scores as for positive and for negative 'Reviews'. Matrix confusion shows ability models right to identify true positive and true negative results as well indicates cases false positive and false negative results.

High ROC-AUC (0.8536) suggests that classifier perfect distinguishes positive and negative cases.

	precision	recall	f1-score	support
Positive	0.73	0.88	0.80	148
Negative	0.85	0.69	0.76	152
accuracy			0.78	300
macro avg	0.79	0.78	0.78	300
weighted avg	0.79	0.78	0.78	300

roc\_auc\_score for Gradient Boosting: 0.8535739687055477

## 3. Logistic Regression

Classifier Logistics Regression demonstrates high performance with 79% accuracy. Indicators precision, recall and f1-scores as for positive and for negative 'Reviews' fine balanced, showing efficiency classifier in the correct identification copies everyone class. Matrix confusion shows significant quantity true positive and true negative values that indicates on ability classifier do accurate forecasts.

In particular, the classifier reaches high ROC-AUC indicator is 0.8763, which emphasizes his reliable ability distinguish positive and negative mood. This grade testifies about efficiency models in discrimination two classes.

Values accuracy and completeness how for positive and for negative 'Reviews' closely coincide that assumes balanced compromise between minimization false positive and false negative results .

	precision	recall	f1-score	support
Positive	0.77	0.82	0.79	148
Negative	0.82	0.76	0.78	152
accuracy			0.79	300
macro avg	0.79	0.79	0.79	300
weighted avg	0.79	0.79	0.79	300

roc\_auc\_score for Logistic Regression: 0.8763113442389758

#### 4. Decision Tree

Decision Tree Classifier demonstrates commendable performance with 79.67% accuracy. Indicators precision, recall and f1-scores as for positive and for negative 'Reviews' fine balanced that reflects ability classifier do accurate forecasts by two classes. Matrix confusion reveals significant quantity true positive and true negative sides, emphasizing skill classifier right to identify cases everyone 'Reviews'.

It is noteworthy that classifier reaches decent ROC-AUC indicator is 0.8135, which indicates on his ability effectively distinguish positive and negative 'Reviews'.

	precision	recall	f1-score	support
Positive	0.76	0.85	0.81	148
Negative	0.84	0.74	0.79	152
accuracy			0.80	300
macro avg	0.80	0.80	0.80	300
weighted avg	0.80	0.80	0.80	300

roc\_auc\_score for DT: 0.8135224039829303

Finally, I can say with confidence that the Decision Tree algorithm is superior other algorithms with the highest accuracy 79.67%. Behind him should Random Forest with an accuracy of 78.67%. Logistics Regression reaches competitive accuracy 79.00%, which demonstrates its efficiency. Gradient Boosting, albeit a little below by accuracy, demonstrates balanced performance by accuracy and completeness for both classes.

### 3. Classification using a BERT-based model with fine-tuning

In that analysis I used model on based on BERT with fine setting classifications text and compared its performance with others algorithms, including Random Forest (RF), Gradient Boosting (GB), Logistic Regression (LR), Decision Tree (DT), and earlier used model on BERT- based without accurate settings .

Dataset was loaded and pre-loaded processed for processing missed values and collateral relevant types data.

Text data were separated on training, validation and testing sets using stratified approach.

The BERT model was finalized on training dialling using architecture binary classifications.

Model trained for 20 epochs and visualized story training.

```
[[94  6]
 [10 90]]
```

	precision	recall	f1-score	support
0	0.90	0.94	0.92	100
1	0.94	0.90	0.92	100
accuracy			0.92	200
macro avg	0.92	0.92	0.92	200
weighted avg	0.92	0.92	0.92	200

Accuracy: 0.92

AUC: 0.9199999999999999

Model demonstrates balanced performance with high accuracy and completeness for both classes that testifies to its processing efficiency how positive and negative cases.

General 92% accuracy deserves praise because reflects skill models right classify copies.

An AUC of 0.92 means reliable discriminatory ability, emphasizing ability models distinguish two class.

Metric		RF	GB	LR	DT	Bert
Accuracy		0.786667	0.783333	0.79	0.796667	0.92
Precision	(Class 0)	0.772727	0.734463	0.767296	0.763636	0.9
Precision	(Class 1)	0.80137	0.853659	0.815603	0.837037	0.94
Recall	(Class 0)	0.804054	0.878378	0.824324	0.851351	0.94
Recall	(Class 1)	0.769737	0.690789	0.756579	0.743421	0.9
F1-Score	(Class 0)	0.788079	0.8	0.794788	0.805112	0.92
F1-Score	(Class 1)	0.785235	0.763636	0.784983	0.787456	0.92
ROC AUC		0.786895	0.784584	0.790452	0.797386	0.92

At comparison indicators productivity various algorithms, including Random Forest (RF), Gradient Boosting (GB), Logistics Regression (LR), Decision Tree (DT) and BERT, I can do some conclusions:

BERT is superior traditional algorithms with 92% accuracy, demonstrating its overall efficiency tasks classifications.

BERT demonstrates high accuracy for both classes (0 and 1) that indicates on his ability minimize false positive and false negative results.



BERT provides balanced call for both classes, guaranteeing effective capture copies everyone class.

BERT reaches ROC AUC of 0.92, demonstrating high discriminatory ability and surpassing traditional algorithms.

Traditional algorithms demonstrate competitive performance, but using BERT contextual information and opportunities deep training gives noticeable advantage.

Traditional algorithms, despite on a reliability, can be difficult fix complex relationships present in the data that emphasizes transformative impact advanced models deep training such like BERT.

#### 4. Topic detection

For promotion quality text data was applied some stages pre-processing. To them relate deletion signs punctuation, numbers and stop words, correction spelling errors and lemmatization. The cleared text is then was converted to words for further analysis.

LDA model was trained on previously processed text data using libraries Gensim. Quantity those was installed equal two, and a model passed 20 passes for clarifications distribution topics. Received topics were researched on item meaningful interpretations.

LDA model revealed in reviews two dominant topics. Here key terms associated with each topic:

```
[ (0,
  '0.047*the" + 0.030*is" + 0.023*and" + 0.020*phone" + 0.017*this" +
  '0.017*it" + 0.017*great" + 0.017*to" + 0.016*very" + 0.014*of"'),
 (1,
  '0.044*the" + 0.038*it" + 0.032*and" + 0.019*this" + 0.018*to" +
  '0.017*not" + 0.016*with" + 0.014*my" + 0.013*in" + 0.012*is"')]
```

First topic, judging by everything, reflects positive moods associated with functions telephone. Such words like 'excellent' and 'very' suggest positive impressions, and the mention of 'phone' indicates on what users express satisfaction product.

Second subject reveals mixture moods and references to the user experience. The term 'no' implies cases when users could face problems or disadvantages Inclusion the words 'my' and 'in' hint on private experience reviewers.

For assessments quality discovered those was used index consistency - an indicator showing how fine vary topics. Received point consistency was 0.41, which testifies about moderate level consistency. Although this score and not is exclusively tall, he indicates on reasonable degree interpretability identified topics.

