Machine Learning Course

Predict users' product reviews sentiment

JOHN KOUMENTIS, MTN2012

Scope

Recognize sentiment based on product reviews written in Greek

Why:

- Track customer sentiment on products and
- Check how it changes over time
- Identify product's pros and cons
- Initiate marketing campaigns

How:

- Evaluate the performance of different Machine Learning Classification algorithms over predicting the correct sentiment (Positive or Negative) of the given review
- Choose the best model and tune it
- Inspect the results on online product reviews

Get the Data

Publicly available data:

- 'Amazon Cell Phones Reviews' dataset found in Kaggle (https://www.kaggle.com/grikomsn/amazon-cell-phones-reviews)
- 67986 cell phone ratings and reviews extracted from Amazon

	asin	rating	date	verified	title	body	helpfulVotes
0	B0000SX2UC	3	October 11, 2005	False	Def not best, but not worst	I had the Samsung A600 for awhile which is abs...	1.0
1	B0000SX2UC	1	January 7, 2004	False	Text Messaging Doesn't Work	Due to a software issue between Nokia and Spri...	17.0
2	B0000SX2UC	5	December 30, 2003	False	Love This Phone	This is a great, reliable phone. I also purcha...	5.0
3	B0000SX2UC	3	March 18, 2004	False	Love the Phone, BUT...!	I love the phone and all, because I really did...	1.0
4	B0000SX2UC	4	August 28, 2005	False	Great phone service and options, lousy case!	The phone has been great for every purpose it ...	1.0
5	B0000SX2UC	4	September 25, 2005	False	Worked great for me	Hello, I have this phone and used it until I d...	NaN
6	B0000SX2UC	5	April 16, 2004	False	Wanna cool Nokia? You have it here!	Cool. Cheap. Color: 3 words that describe the ...	2.0
7	B0000SX2UC	4	April 3, 2004	False	Problem with 3588i universal headset	The 3599i is overall a nice phone, except that...	2.0
8	B0000SX2UC	5	November 24, 2003	False	cool phone!!!!!!!!	I've never owned a Nokia phone before, so this...	7.0
9	B0000SX2UC	3	February 2, 2004	False	Pissed off-a little bit	ok well im in school and i need the text messa...	3.0

Explore Data 1/5

Remove rows with empty reviews

Isolate reviews and ratings (‘body’, ‘rating’ columns respectively) in a new dataset

Load reviews’ column in XLSX format at Google Translate and retrieve their Greek language equivalent.

	reviews	greek	ratings
0	I had the Samsung A600 for awhile which is abs...	Είχα το Samsung A600 για λίγο που είναι απόλυτ...	3
1	Due to a software issue between Nokia and Spri...	Λόγω ενός προβλήματος λογισμικού μεταξύ της No...	1
2	This is a great, reliable phone. I also purcha...	Αυτό είναι ένα υπέροχο, αξιόπιστο τηλέφωνο. Αγ...	5
3	I love the phone and all, because I really did...	Λατρεύω το τηλέφωνο και όλα, γιατί πραγματικά ...	3
4	The phone has been great for every purpose it ...	Το τηλέφωνο ήταν τέλειο για κάθε σκοπό που προ...	4
5	Hello, I have this phone and used it until I d...	Γεια σας, έχω αυτό το τηλέφωνο και το χρησιμοπ...	4
6	Cool. Cheap. Color: 3 words that describe the ...	Δροσερός. Φτηνός. Χρώμα: 3 λέξεις που περιγράφ...	5
7	The 3599i is overall a nice phone, except that...	Το 3599i είναι γενικά ένα ωραίο τηλέφωνο, εκτό...	4
8	I've never owned a Nokia phone before, so this...	Δεν είχα ποτέ προηγουμένως τηλέφωνο Nokia, γι ...	5
9	ok well im in school and i need the text messa...	εντάξει, είμαι στο σχολείο και χρειάζομαι τα γ...	3

Explore Data 2/5

Clean reviews text

greek	ratings	cleaned_text
Είχα το Samsung A600 για λίγο που είναι απόλυτ...	3	έχω λίγο είναι απόλυτος μπορείτε διαβάσω κριτι...
Λόγω ενός προβλήματος λογισμικού μεταξύ της Νο...	1	λόγω ενός προβλήματος λογισμικός μεταξύ της τη...
Αυτό είναι ένα υπέροχο, αξιόπιστο τηλέφωνο. Αγ...	5	αυτό είναι ένα υπέροχος αξιόπιστος τηλέφωνο αγ...
Λατρεύω το τηλέφωνο και όλα, γιατί πραγματικά ...	3	λατρεύω τηλέφωνο όλος γιατί πραγματικά χρειαζά...
Το τηλέφωνο ήταν τέλειο για κάθε σκοπό που προ...	4	τηλέφωνο ήταν τέλειος κάθε σκοπό προσφέρω εκτό...

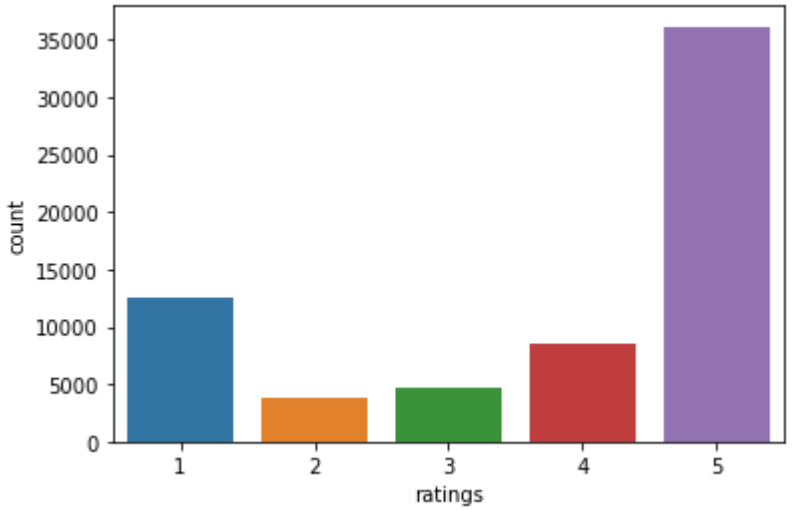
- Cleaning function
  - Convert to lowercase
  - Apply Regular Expressions to remove special characters and all but Greek characters (translation was not perfect)
  - Removed punctuation
  - Removed Greek stop words (existing in NLTK Greek stop words)
  - Apply lemmatization using a Spacy package for Greek language (el_core_news_sm)
  - Trim resulting text

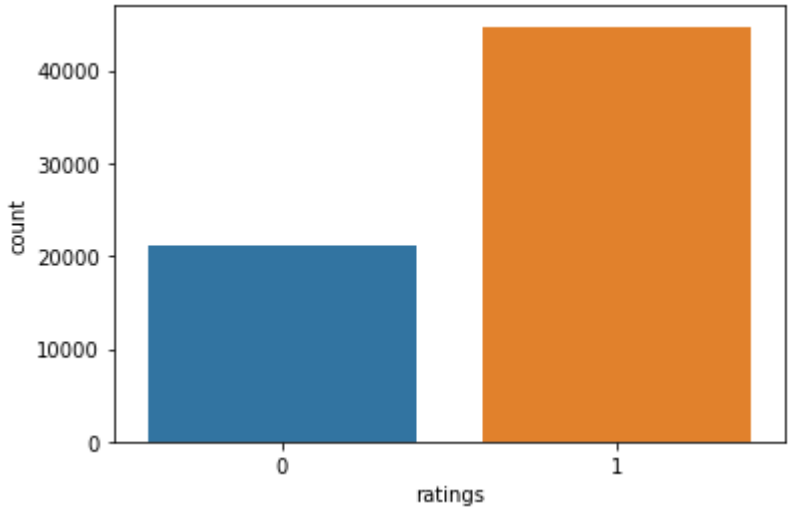
Explore Data 3/5

Ratings are integers from 1 to 5

Represent them in positive/negative sentiment

Map 1,2,3 ratings as negative (class 0) 4,5 as positive (class 1)





Explore Data 4/5

Count words in reviews

- Calculate logarithmic ratios of positive and negative uses of the most common words (used at least 100 times)
  - Pos-to-neg ratio for 'λόγω' = 0.2786746254947829
  - Pos-to-neg ratio for 'υπέροχο' = 2.2204814395004866
  - Pos-to-neg ratio for 'απογοητευμένος' = -1.526315069184163

Balance positive and negative reviews by randomly keeping only as many positive reviews as the negative ones

Explore Data 5/5

Transform the words existing in the reviews to vectors, taking into account their attributes and features.

The tf-idf vectorizer (scikit-learn package) was used to transform the words in vectors according to their frequency (TF) and importance (IDF).

{'έχω': 5, 'λίγο': 81, 'είναι': 48, 'απόλυτος': 21, 'μπορείτε': 95, 'διαβάσω': 41, 'κριτική': 78, 'μου': 93, 'αυτό': 27, 'εντοπίσω': 51, 'οργός': 100, 'ηλίθιος': 57, 'πράγμα': 113, 'τελικά': 133, 'πέθανε': 105, 'μένα': 85, 'γι': 32, 'χρησιμοποίησα': 156, 'τηλέφωνο': 136, 'αγόρασα': 7, 'γκαράζ': 34, 'αναρωτιέμαι': 13, 'ότι': 161, 'πούλησε': 112, 'τόσο': 142, 'φτηνό': 150, 'κακό': 66, 'μισώ': 91, 'μενού': 89, 'χρειάζομαι': 155, 'πάντα': 103, 'φτάσω': 149, 'θέλω': 58, 'επειδή': 55, 'πρέπει': 114, 'κάνω': 64, 'κύλιση': 79, 'ατελείωτο': 25, 'συνήθω': 128, 'τηλέφωνα': 135, 'αριθμημένες': 22, 'κατηγορία': 74, 'έτσι': 4, 'απλά': 17, 'πατήσω': 109, 'εκεί': 49, 'πάτε': 104, 'πόνος': 118, 'βάζει': 28, 'αθόρυβη': 8, 'δόνηση': 45, 'εάν': 46, 'είμαι': 47, 'τάξη': 132, 'χτυπάω': 158, 'απενεργοποιήσω': 16, 'αμέσως': 12, 'υπάρχω': 143, 'γρήγορος': 36, 'τρόπος': 140, 'σιγήσω': 123, 'καταραμένος': 70, 'θυμάστε': 61, 'βάζω': 29, 'σιωπηλό': 124, 'έμαθο': 2, 'σκληρός': 125, 'αλήθειος': 10, 'υπόθεση': 145, 'αποστολή': 19, 'κατεβείτε': 73, 'σπάσω': 127, 'νύχι': 98, 'σας': 122, 'διαδικασία': 42, 'επίσης': 54, 'καταστρέφω': 71, 'θήκη': 59, 'κάθε': 63, 'φορά': 146, 'δοκιμάζω': 43, 'κάποιο': 65, 'λόγο': 84, 'άρχομαι': 1, 'δίνω': 38, 'προβλήμαα': 116, 'όταν': 160, 'κατάφερα': 69, 'ανοίξω': 14, 'κουμπί': 77, 'μπορώ': 96, 'μεγαλύτερος': 88, 'ισχυρότερη': 62, 'καλό': 68, 'λήψη': 80, 'πολύ': 110, 'άθλιας': 0, 'χρησιμοποιούσο': 157, 'ασανσέρ': 24, 'ένα': 3, 'αξιοθαύμαστος': 15, 'κατόρθωμα': 75, 'δεδομένου': 39, 'παλιό': 107, 'χάσω': 152, 'εξυπηρέτηση': 52, 'απλώς': 18, 'τσέπη': 141, 'σύγκριση': 131, 'παλί': 106, 'λειτουργώ': 82, 'αρκετά': 23, 'καλά': 67, 'ήχος': 6, 'κλήση': 76, 'δυνατός': 44, 'ακούσω': 9, 'φορτίζω': 147, 'πραγματικά': 115, 'γρήγορο': 35, 'μεγάλη': 86, 'διάρκειο': 40, 'ζωή': 56, 'της': 138, 'μπαταρίας': 94, 'θερμαίνομαι': 60, 'σαν': 121, 'μια': 90, 'πατάτα': 108, 'φούρνο': 148, 'ούτε': 102, 'τη': 134, 'μεγάλος': 87, 'τηλεφωνικός': 137, 'συνεδριάσεο': 129, 'ωραίος': 159, 'φωτεινός': 151, 'οθόνη': 99, 'χαριτωμένος': 153, 'τρόποι': 139, 'προσαρμογή': 117, 'γραμμή': 37, 'οριστώ': 101, 'μοβ': 92, 'ροζ': 119, 'πορτοκαλί': 111, 'λπ': 83, 'συνολικά': 130, 'εντάξω': 50, 'εξυπηρετώ': 53, 'σκοπό': 126, 'αλλά': 11, 'σίγουρα': 120, 'χλωμό': 154, 'αυτά': 26, 'νέος': 97, 'βγαίνω': 30, 'από': 20, 'γιατί': 33, 'καταφέρω': 72, 'γίνω': 31, 'υπέροχος': 144}

Core ML tasks 1/4

Training pipeline

- 10-fold cross validation split
- Record average training/testing time, train/test set accuracy, test set F1 score

Used classifiers:

- Multinomial Naive Bayes
- Linear Support Vector Classifier
- Logistic Regression Classifier
- Decision Tree Classifier
- AdaBoost Classifier

Core ML tasks 2/4

Train models in both datasets and collect performance metrics

```
########## Using MultinomialNB() classifier ##########

Average training time:  0.0171875
Average testing time:  0.0015625

Average training accuracy: 0.8655390121205983, σ: 0.00047733704721759554
Average testing accuracy: 0.8473564266180492, σ: 0.003319616219528673
Average F1-Score: 0.8963954671494354


--------------------------------------------------------------------------------
########## Using LinearSVC() classifier ##########

Average training time:  0.5671875
Average testing time:  0.0

Average training accuracy: 0.9413383301259326, σ: 0.0003908976120878462
Average testing accuracy: 0.88529322394409, σ: 0.0029057883123209454
Average F1-Score: 0.916572589175376


--------------------------------------------------------------------------------
########## Using LogisticRegression(max_iter=1000, solver='saga') classifier ##########

Average training time:  0.9359375
Average testing time:  0.003125

Average training accuracy: 0.9108578952699281, σ: 0.00038109334616701627
Average testing accuracy: 0.8894257064721968, σ: 0.0038193276527013907
Average F1-Score: 0.9201833394906861


--------------------------------------------------------------------------------
########## Using DecisionTreeClassifier(max_depth=15) classifier ##########

Average training time:  8.5890625
Average testing time:  0.00625

Average training accuracy: 0.7990090820081704, σ: 0.0021801026131743695
Average testing accuracy: 0.7579914919477363, σ: 0.0046800453014597555
Average F1-Score: 0.8371567902426809


--------------------------------------------------------------------------------
########## Using AdaBoostClassifier(base_estimator=MultinomialNB(), learning_rate=0.01,
                   n_estimators=200) classifier ##########

Average training time:  8.3859375
Average testing time:  0.4984374999999999

Average training accuracy: 0.6797326040717108, σ: 0.0005965489115379864
Average testing accuracy: 0.6797326040717108, σ: 0.005368940203841951
Average F1-Score: 0.8093221273500902
```

Whole dataset

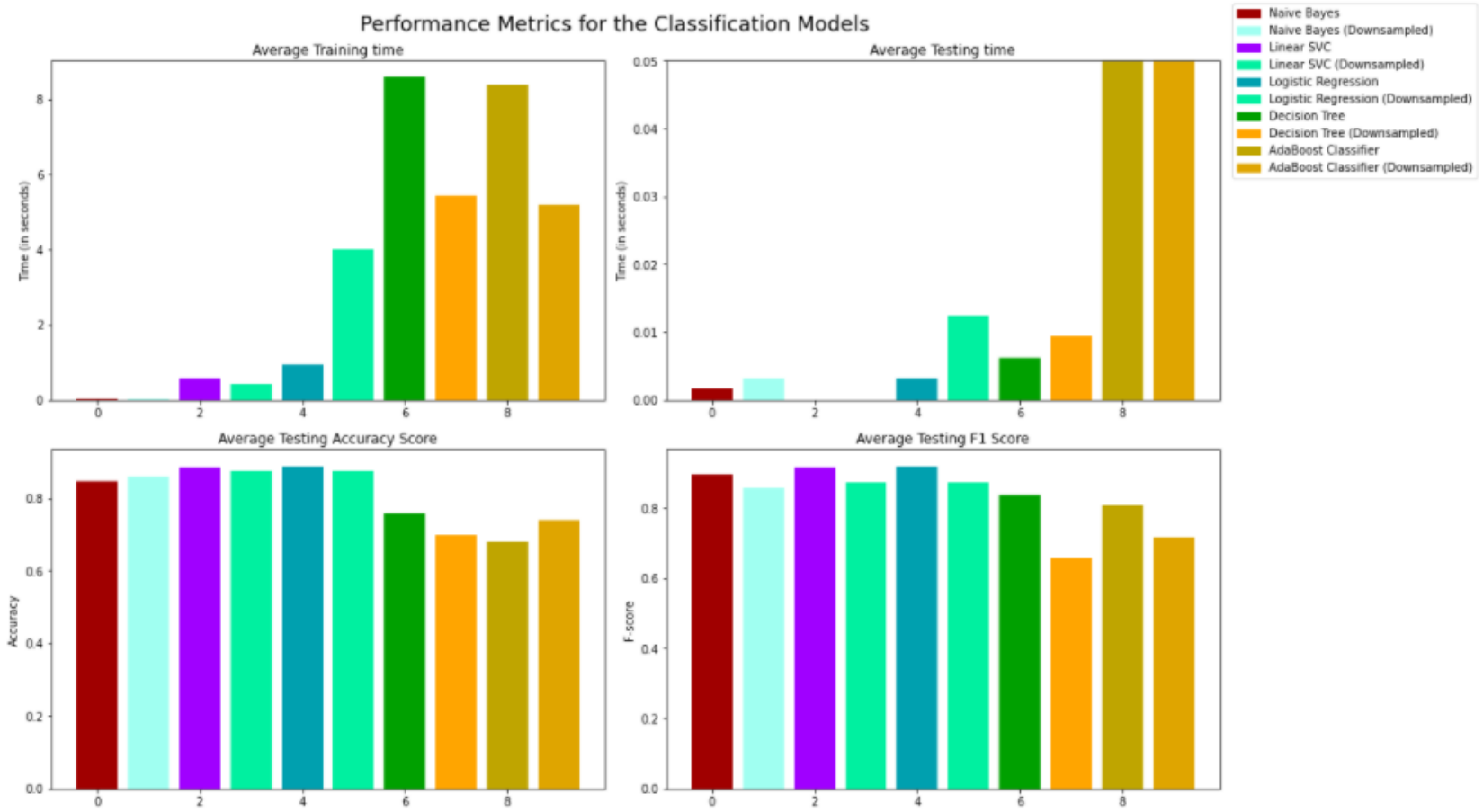
```
########## Using MultinomialNB() classifier ##########

Average training time:  0.015624999999999998
Average testing time:  0.003125

Average training accuracy: 0.8873023402909551, σ: 0.0005431611440830446
Average testing accuracy: 0.8592030360531308, σ: 0.004932445861466122
Average F1-Score: 0.8581022462882931


--------------------------------------------------------------------------------
########## Using LinearSVC() classifier ##########

Average training time:  0.4140625
Average testing time:  0.0

Average training accuracy: 0.9419592030360532, σ: 0.0003253595958239828
Average testing accuracy: 0.8739800759013283, σ: 0.005760437428830688
Average F1-Score: 0.8733375148158116


--------------------------------------------------------------------------------
########## Using LogisticRegression(max_iter=1000) classifier ##########

Average training time:  4.01875
Average testing time:  0.0125

Average training accuracy: 0.9035183428209994, σ: 0.0005094629739905493
Average testing accuracy: 0.8749762808349146, σ: 0.0038031371724502857
Average F1-Score: 0.8732919470657559


--------------------------------------------------------------------------------
########## Using DecisionTreeClassifier(max_depth=15) classifier ##########

Average training time:  5.4468749999999995
Average testing time:  0.009375

Average training accuracy: 0.7522954880877083, σ: 0.004779096944783228
Average testing accuracy: 0.6975094876660342, σ: 0.009734716389391058
Average F1-Score: 0.6563481319650479


--------------------------------------------------------------------------------
########## Using AdaBoostClassifier(base_estimator=MultinomialNB(), learning_rate=0.01,
                   n_estimators=200) classifier ##########

Average training time:  5.1781250000000005
Average testing time:  0.32031250000000006

Average training accuracy: 0.7473408180476492, σ: 0.03769528957891911
Average testing accuracy: 0.7397058823529412, σ: 0.038378635263510985
Average F1-Score: 0.7159551704601889
```

Balanced dataset

Core ML tasks 3/4

Best overall performance Linear SVC

- Lower training and testing times
- High average accuracy
- High average F1 score



Core ML tasks 4/4

Hyperparameter tuning function

- Stratified split (train_test_split), train data 2/3, test data 1/3
- Used Grid Search
- Tuned based on F1 Score metric

Whole dataset

```
Unoptimized Model
Accuracy: 0.8807605543022881
F1 Score: 0.9131455399061033
Hyperparameters: C: 1.0, Loss: squared_hinge, Dual: True, Fit-Intercept: True Class Weigth: None

Best Model
Accuracy: 0.8866995073891626
F1 Score: 0.9175792893265012
Hyperparameters: C: 1.0, Loss: hinge, Dual: True, Fit_Intercept: True Class Weigth: None

--------------------------------------------------------------------------------
```

Balanced dataset

```
Unoptimized Model
Accuracy: 0.871055847049522
F1 Score: 0.8692610406646262
Hyperparameters: C: 1.0, Loss: squared_hinge, Dual: True, Fit-Intercept: True Class Weigth: None

Best Model
Accuracy: 0.8735714799108747
F1 Score: 0.8723419696639815
Hyperparameters: C: 1.0, Loss: hinge, Dual: True, Fit_Intercept: False Class Weigth: None

--------------------------------------------------------------------------------
```

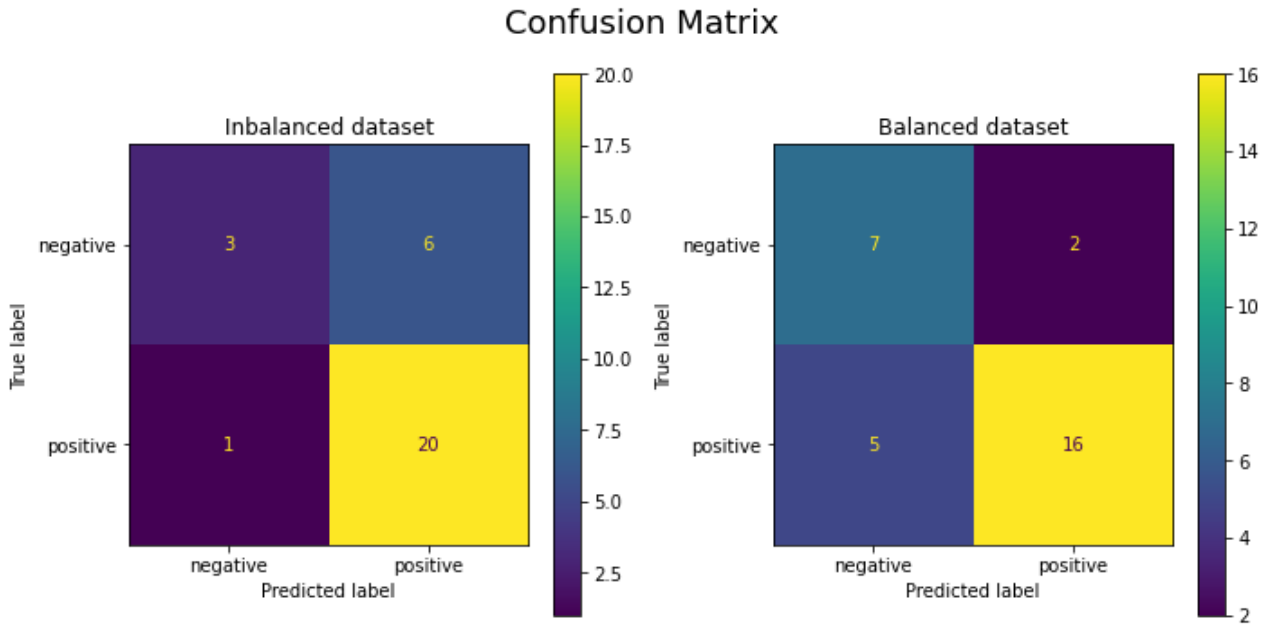
Communicate 1/2

Evaluate models on Greek language product reviews from Skroutz

3 cell phones, 1 GPU, 1 pair of sneakers

Model trained on full data performs better in correctly classifying positive reviews (higher precision on the positive label)

Model trained on balanced dataset has higher recall on the negative label in some experiments, but generally performs much worse



https://www.skroutz.gr/s/23272388/Xiaomi-Redmi-Note-9-Pro-128GB-Tropical-Green.html#reviews

Communicate 2/2

- Sentiment classification task using artificially translated text is a feasible, well defined procedure (accuracy > 0.88).
- Model trained on full data performs better in classifying the majority class
- Model performs rather well on product reviews different than cell phones.
- Could be used in identifying customer preferences from product reviews in various web store platforms towards:
  - Analyzing product popularity
  - Define its pros and cons
  - Target certain customer groups

➢ Could introduce another class to express neutral sentiment

➢ Sample additional reviews to balance the data

➢ Take Greek syntax into account to improve classification results

Code available in https://github.com/JoKoum/Machine-Learning-Project

- Main dependencies: pandas, nltk, spacy, sklearn