PROJECT REPORT

MIGROS GROUP

Lam Bryan, Manzocchi Nicola, Raghupathy Abirami

INTRODUCTION

Exploratory Data Analysis

01

ACCURACY

Quick presentation of the accuracy of our submissions

02

DEVELOPMENT

The issues we encountered, how we solved them
An overview of our notebook

03

FUTURE DEVELOPMENT

Extended idea we could work on to improve our model accuracy

04

01

INTRODUCTION

GitHub Setup, EDA, First submission

Real or Not?

NLP with Disaster Tweets: a bit of data exploration

Train Set

5 Rows

X

6471 Columns

Keywords

222 different keywords

But 55 null values

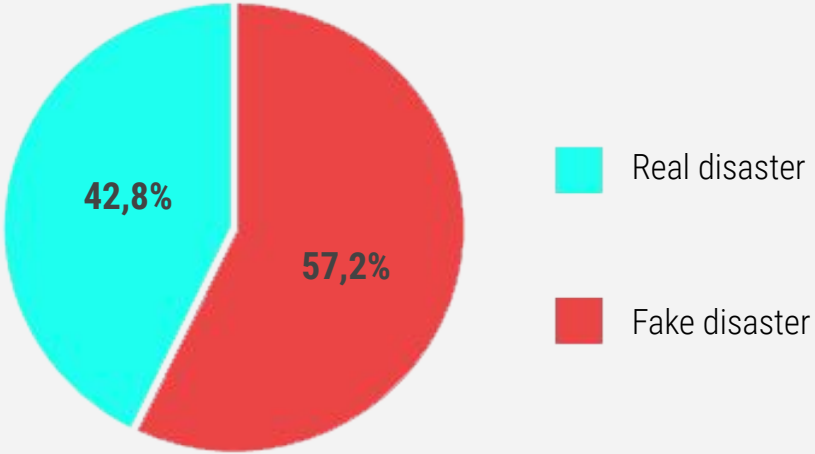
Location

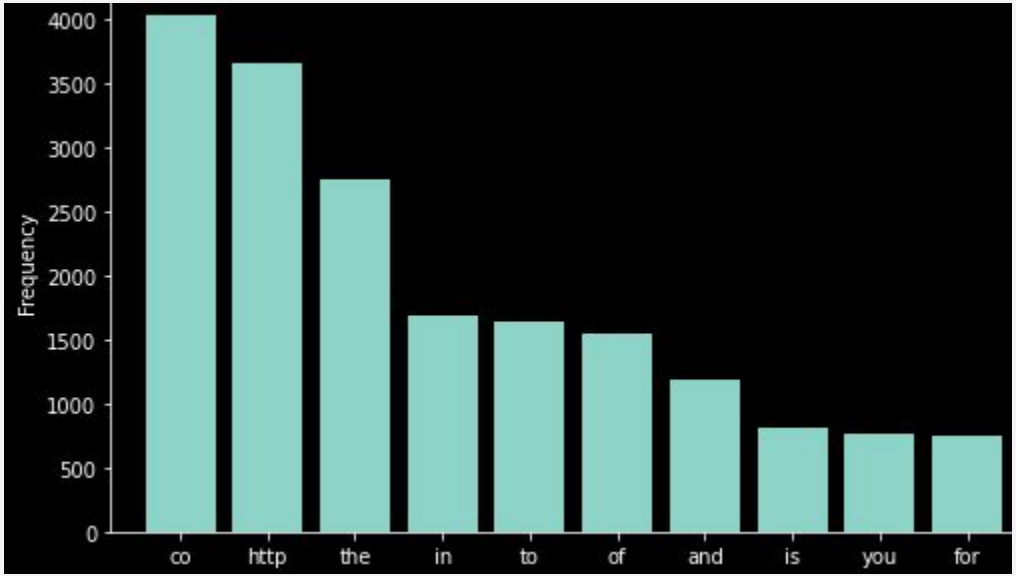
2922 different locations

But 2141 null values

Base rate

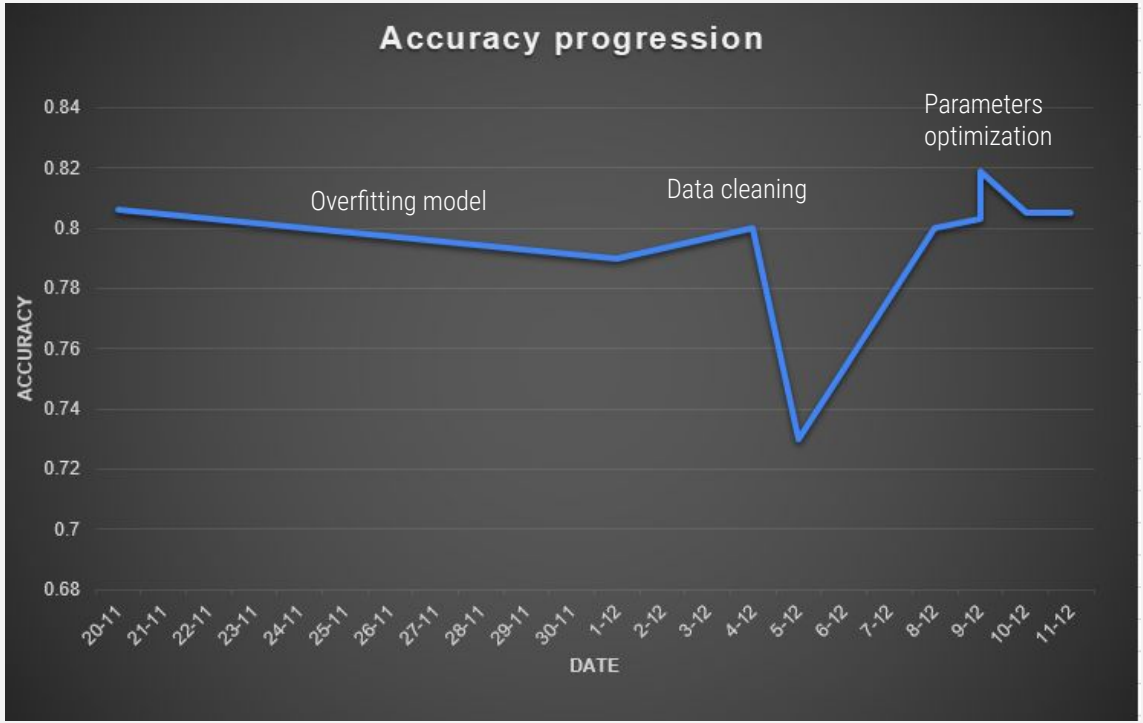
14,4% more fake tweets than real reports





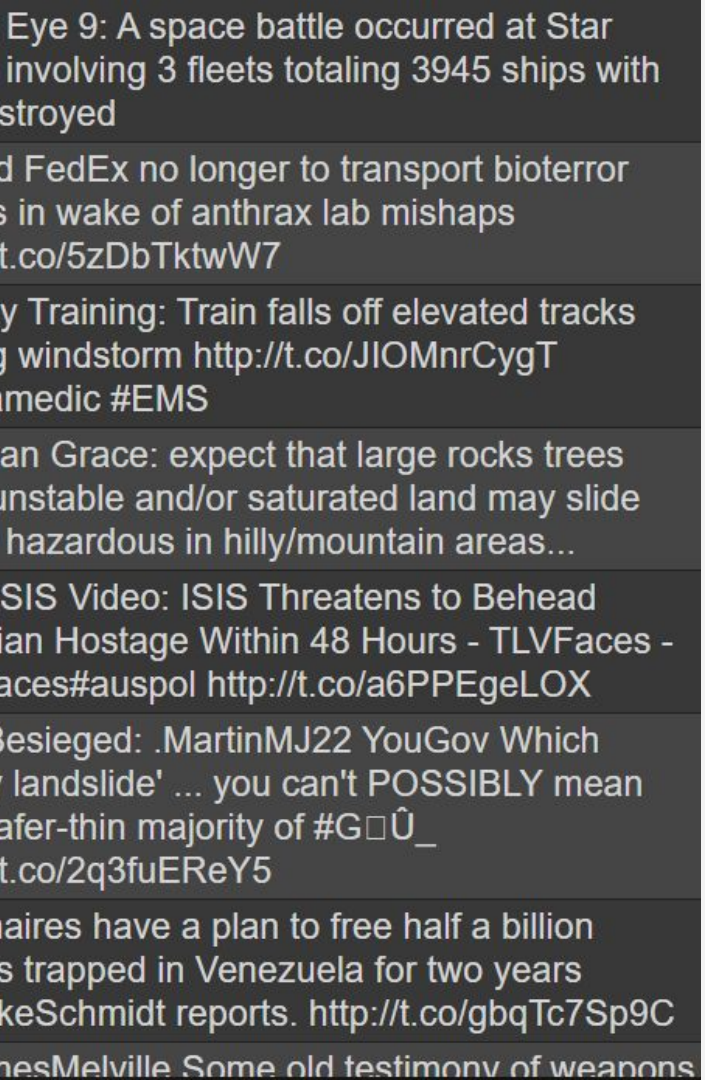
Most common words

Lots of noise! Stopwords and URL makes the data inaccurate



02 ACCURACY

The problem of overfitting



03

DEVELOPMENT

Mainly an issue of data cleaning and parameters optimization

Simple logistic regression

Without data cleaning and only text column

Extensive Text cleaning

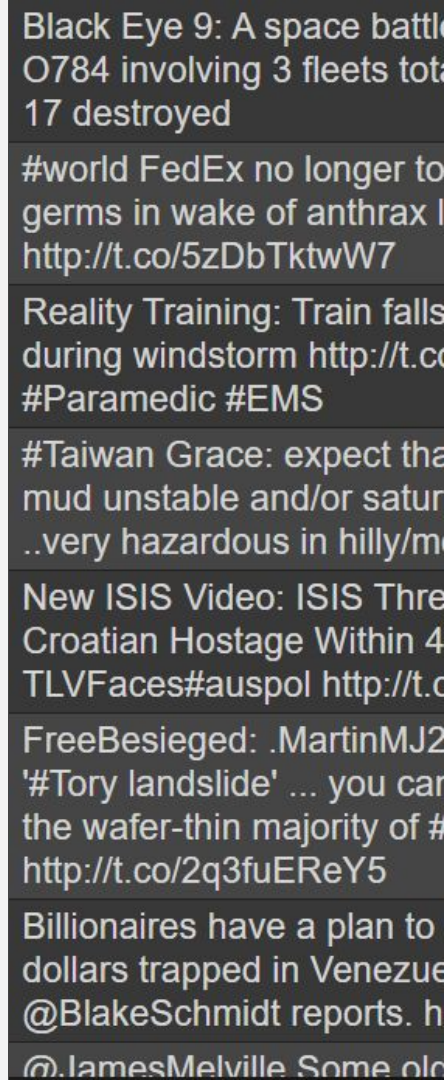
Different variation of removing noise

Different Classifier

Logistic Regression, Random Forest, kkn model after data cleaning

Optimization of parameters

Optimization of LR Hyperparameters with GridsearchCV



Simple logistic regression

Without data cleaning and using only text column

```
TRAIN ACCURACY SCORE:
0.8937
CONFUSION MATRIX:
[[2882   79]
 [ 471 1744]]
```

```
TEST ACCURACY SCORE:
0.7884
CONFUSION MATRIX:
[[656  84]
 [190 365]]
```

Different Classifier

With basic data cleaning and using only text column

Random Forest Classifier

- Train accuracy score:
  0.8374
- Test accuracy score:
  0.7409

Doc2Vec

- Train accuracy:
  0.5939
- Test accuracy score:
  0.5876

Extensive Text Cleaning

Accuracy improved

- Cleaning duplicates
- Replacing %20
- Removing Twitter tags
- Removing hashtags
- Removing “amp”
- Removing URL
- Removing noise (RT, co, û_, □©, ûªs)

Accuracy decreased

- Removing digits
- Removing punctuations
- Removing specific noise (rt, û)

Optimal TF-IDF vectorizer

```
[ ]  # Improved TF-IDF tokenizer
     tfidf_vector = TfidfVectorizer(tokenizer=spacy_tokenizer, ngram_range=(3, 4), min_df=1, max_df=1.0, analyzer='char')
```

Comparing different classifier after text cleaning using cross-validation

Logistic Regression

- Mean accuracy score on test set: **0.8008**
- Standard deviation: **0.0164**

Knn model

- Mean accuracy score on test set: **0.7828**
- Standard deviation: **0.0260**

Random Forest Classifier

- Mean accuracy score on test set: **0.7806**
- Standard deviation: **0.0198**

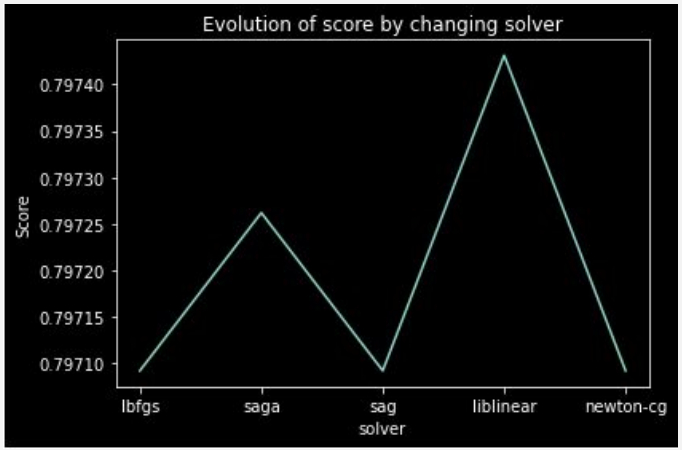
Optimization of Logistic regression Hyperparameters using GridsearchCV

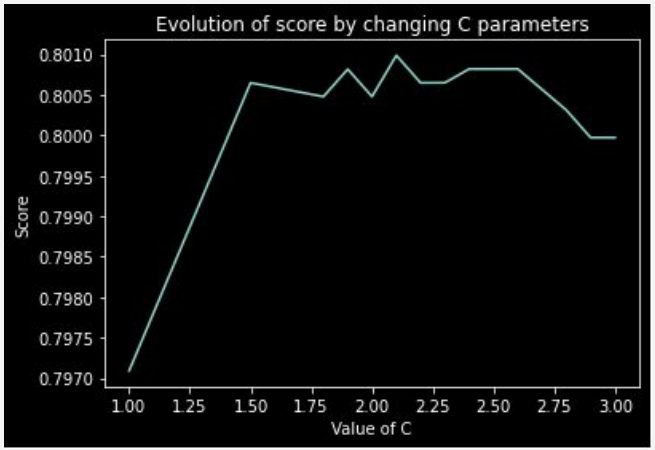
```
# Gridsearch Logistic regression
parameters = dict({'solver': ['lbfgs', 'saga', 'liblinear'], 'C':[1, 10]})
LR =linear_model.LogisticRegression(max_iter=1000, random_state=72)

clf = GridSearchCV(
    LR, parameters, scoring='accuracy'
    )
best_model = clf.fit(X_train_vec, y_train)
```

Train accuracy: **0.8892**

Test Accuracy : **0.8122**

How the optimization looks like in plots





Solution

Effective cleaning

Optimized parameter for TF-IDF

Logistic Regression with GridSearchCV

Result

Accuracy: 0.819

AICrowd best Rank: 6

Future Development

04

1. We actually used the column **Keywords** in our model but only after the Challenge closed
2. It would be interesting to use the **Location** of the tweets as one of the features
3. Further optimize some **hyperparameters**
4. Expand the train dataframe so we can use **other classifiers** such as Random Forest

THANKS!

For your attention and your time

MIGROS GROUP

Lam Bryan, Manzocchi Nicola, Raghupathy Abirami

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**