Disaster Outreach Classifier

Capstone Project

Contents

- Data Cleaning
- EDA
- Modelling
- Deployment
- Recommendation

Problem Statement

- To classify disaster outreach messages and improve the triage system in Disaster Management Cycle.

Dataset

Dataset is retrieved from Figure 8.

It is multi-labelled with 30,000 messages drawn from earthquakes, floods and super-storms.

Data Cleaning

- Removal of Nulls and Duplicates

```
df = df.dropna()
df = df.drop_duplicates()
```

- Removal of wrong label

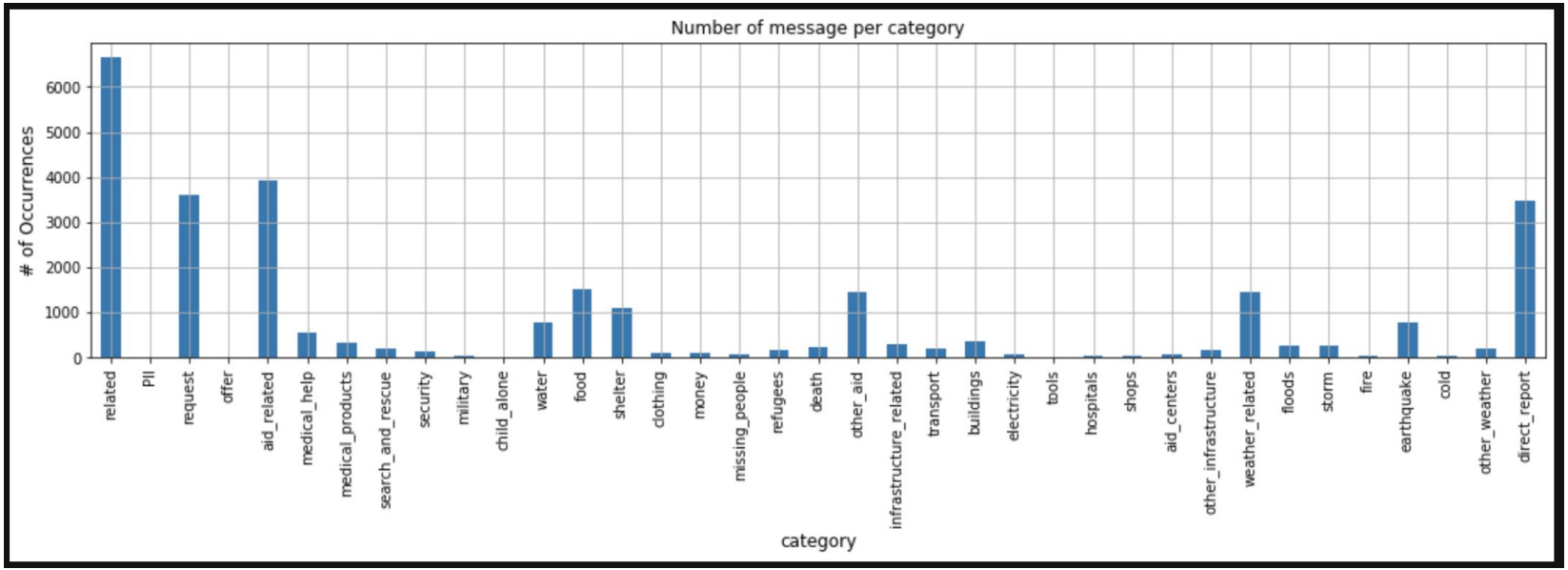
```
#Drop 193 rows of messages that mostly not properly translated
df = df[df.related != 2]
```

EDA

Exploratory Data Analysis

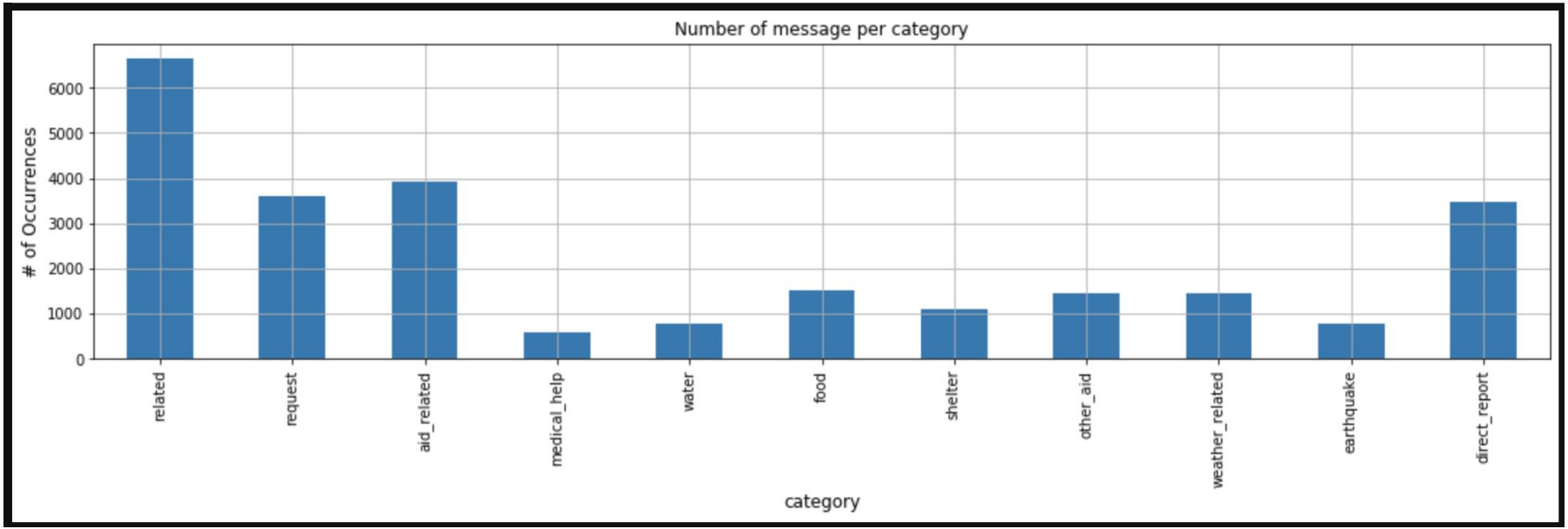
EDA

- Multi-Labels: 37 Labels



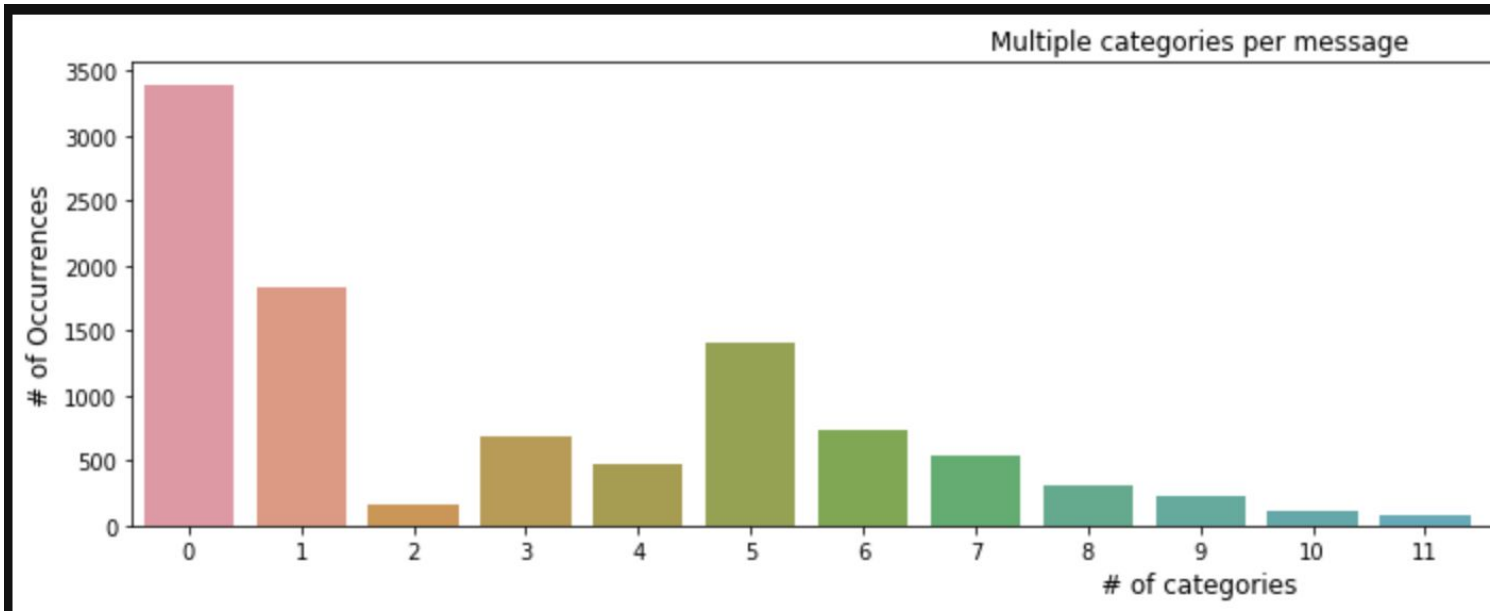
EDA

- Labels with more than 5% occurrence



EDA

- Messages with Multiple - Labels



EDA

- 3 Genres: Direct, News, Social

```
df[df['genre'] == 'direct'].head(5)
```

	id	split	message	original	genre	related
0	9	test	UN reports Leogane 80-90 destroyed. Only Hospital St. Croix functioning. Needs supplies desperately.	UN reports Leogane 80-90 destroyed. Only Hospital St. Croix functioning. Needs supplies desperately.	direct	1
1	39	test	We are at Gressier we needs assistance right away. ASAP, Come help us.	Se gressier nou an difikilte tanpri vin ede nou	direct	1
2	49	test	Delmas 33 in Silo, need water.	Delma 33 silo gen problem dlo	direct	1
3	79	test	SOS SOS, please provide police officers on the streets as they are very insecure	EMGENCY EMGENCY SI POLIS LA TE KA BAY PREZANS LI SOU CHAK PLAS AK LARI LI TAP BON PASKE CHAK SWA YO F ANPIL DEGA MENM DETWI LAVI MOUN ENPOTAN	direct	1
4	99	test	I am a driver, a mechanic .. I want to help	MWEN SE YON NMALIEN, CHOF, MEKANISYEN MWEN BEZWEN BAY SEKOU	direct	1

```
df[df['genre'] == 'news'].head(5)
```

	id	split	message	original	genre	related
1301	15739	test	Local administrative bodies are trained to quickly issue alerts and evacuation orders and distribute food and blankets at shelters.	NaN	news	1
1302	15749	test	As well, security threats from jihads remained.	NaN	news	1
1303	15759	test	- Deployment of mobile emergency health units to ensure families have immediate access to primary care and medical supplies, with the first team of 10 seasoned local emergency medical staffs deploying to the remote Swat Valley, which has been cut off due	NaN	news	1
1304	15769	test	PWJ will target 30 villages in Ingapu Township and plan to distribute equipment/tools for cleaning and repairing damaged houses, repair and improve	NaN	news	1

```
df[df['genre'] == 'social'].head(5)
```

	id	split	message	original	genre	related
981	11189	test	RT selenagomez UNICEF has just announced an emergency alert for the people of Haiti who were hit by a 7.0 earthquake and a tsunami.. ..	NaN	social	1
982	11199	test	RT YURFAVHALFBREED Moment of silence for those devastated by the earthquake in Haiti	NaN	social	1
983	11219	test	Earthquake in #Haiti? Does anyone know anything about this?	NaN	social	1
984	11249	test	RT OrphanProject Most of you have probably heard this but Haiti was hit by a 7.3 Magnitude earthquake at approx. 4 30pm today. .. htt ..	NaN	social	1
985	11259	test	In Punta Cana Dominican Republic for a friend s wedding. Earthquake that rocked Haiti was not felt here. Feel bad for those folks.	NaN	social	1

EDA - Conclusion

- Based on the EDA, 70% of the categories are less than 5% of the total data set.
- Remaining categories does not attribute to the severity of the outreach.
- We will focus on building a robust binary classification model (Disaster/Non-Disaster).

Data Preparation

Data - Preprocessing

- Understanding the dataset balance - *requires sampling*

```
df['related'].value_counts(normalize=True)
```

```
1    0.675014
0    0.324986
Name: related, dtype: float64
```

```
sm = SMOTETomek(random_state=42)
X_res, y_res = sm.fit_sample(X_train, y_train)
```

- Cleaning of text

```
def clean_text(text):
    # remove HTML tags and URLs
    text = re.sub(r'<.*?>', '', text)
    text = re.sub(r'^https?:\/\/.*[\r\n]*','',text)
    # keep only text without punctuation
    text = re.sub(r'[^\w\s]','',text)
    text = re.sub(" \d+", " ", text)
    # convert text to lowercase
    text = text.strip().lower()
    # split text into a list of words
    token_text = re.split('\W+',text) #W+ --> word chars and dashes permitted
    return token_text
```

```
stop_words = stopwords.words('english')
stop_words
```

```
lemmatizer = WordNetLemmatizer()
def lemmatize_words(text):
    lemma_words = [lemmatizer.lemmatize(word) for word in text]
    return lemma_words
```

Modelling

Models and hyper parameter tuning

- Logistic Regression
- NavieBayers
- Random Forest
- Extra Trees
- Decision Trees
- Bagged Decision Trees
- AdaBoost

```
classifier_model_params = {
    'LogisticRegression' : {
        'penalty' : ['l1', 'l2'],
        'C' : np.arange(.05, 1, .05) },
    'KNN' : {
        'n_neighbors' : np.arange(3, 22, 2) },
    'NaiveBayes' : {
        'alpha' : np.arange(.05, 2, .05)},
    'DecisionTree': {
        'max_depth' : [ 6, 10, 14],
        'min_samples_leaf' : [1, 2],
        'min_samples_split': [2, 3] },
    'BaggedDecisionTree' : {
        'n_estimators' : [20, 60, 100] },
    'RandomForest' : {
        'n_estimators' : [20, 60, 100],
        'max_depth' : [ 2, 6, 10],
        'min_samples_split' : [2, 3, 4] },
    'ExtraTrees' : {
        'n_estimators' : [20, 60, 100],
        'max_depth' : [ 6, 10, 14],
        'min_samples_leaf' : [1, 2],
        'min_samples_split' : [2, 3], },
    'AdaBoost' : {
        'n_estimators' : np.arange(100, 151, 25),
        'learning_rate' : np.linspace(0.05, 1, 20) },
    'GradientBoosting' : {
        'n_estimators' : np.arange(5, 150, 10),
        'learning_rate' : np.linspace(0.05, 1, 20),
        'max_depth' : [1, 2, 3] },
    'XGBoost' : {
        'n_estimators'  : np.arange(100, 151, 25),
        'learning_rate' : np.arange(0.1, 1, .3),
        'max_depth' : [3],
        'alpha' : np.arange(0, 1, .3),
        'lambda' : np.arange(0, 1, .3),
        'gamma' : np.arange(0, 1, .3),
        'subsample' : [.5],
```

Models Selection

- Ada Boost
  - Lowest train-test difference : Approx. 3%
  - Accuracy of 81%

	Model Name	Best Params	Best Score	Train Score	Test Score
2	NaiveBayes	{'alpha': 0.05}	0.850435	0.921147	0.783887
6	BaggedDecisionTree	{'n_estimators': 100}	0.844093	0.997349	0.787705
0	LogisticRegression	{'C': 0.9500000000000001, 'penalty': 'l2'}	0.841253	0.891424	0.811378
4	ExtraTrees	{'max_depth': 14, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}	0.815505	0.862268	0.772814
7	AdaBoost	{'learning_rate': 0.7999999999999999, 'n_estimators': 150}	0.813518	0.838413	0.799924
3	RandomForest	{'max_depth': 10, 'min_samples_split': 2, 'n_estimators': 100}	0.807838	0.838697	0.776632
5	DecisionTree	{'max_depth': 14, 'min_samples_leaf': 1, 'min_samples_split': 2}	0.773855	0.805566	0.723559
1	KNN	{'n_neighbors': 3}	0.655718	0.770636	0.565101

Deployment

Flask - Heroku

Twilio - Sms/Whatsapp

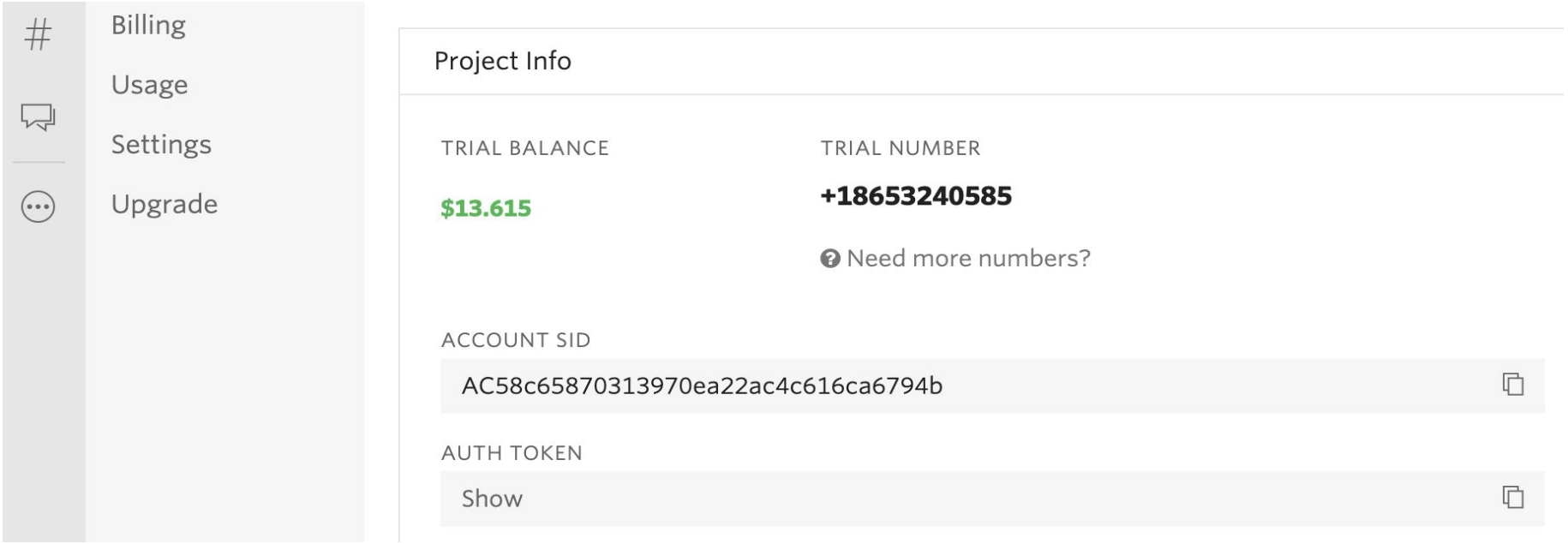
Flask

- Static
- Templates
- Profile
- Requirements
- Pre-train models
- Runtime
- app.py

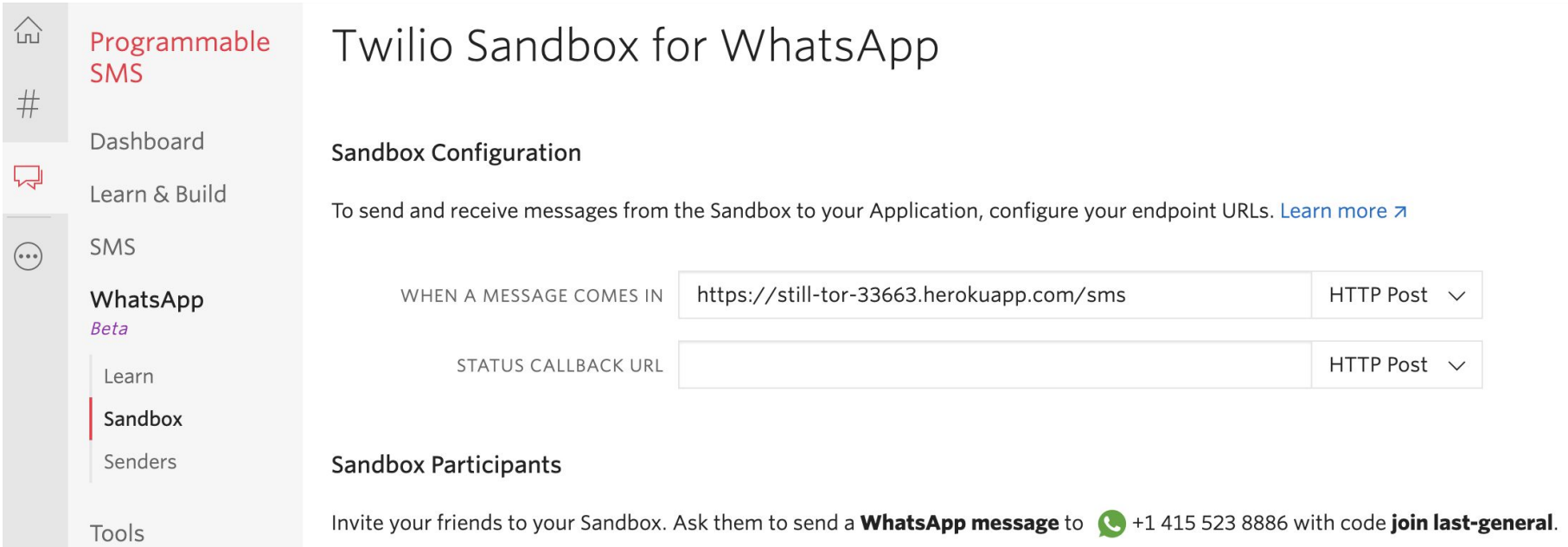
```
app = Flask(__name__, static_url_path='/static')

@app.route('/', methods=['GET', 'POST'])
def home():
    return render_template('index.html')

@app.route('/sms', methods=['POST'])
def sms_reply():
    resp = MessagingResponse()
    #Fetch message
    msg = request.form.get('Body')

    #create reply
    if len(msg) >20:
        #resp.message("Accessing input..")
        #setup of model
        ad = joblib.load('model2.joblib')
        tv = joblib.load('tv.joblib')
        #load text to model
        data = [msg]
        vect = tv.transform(data).toarray()
        my_prediction = ad.predict(vect)
        if my_prediction == 1:
            resp.message('We have received your distress call.\nShare your exact location\nWe will send a pony over')
        else:
            resp.message('Thank you for reaching out.\nPlease do not waste your time and our precious time thank you.')
    else:
        #resp.message("Accessing input..")
        resp.message('Furnish us with following information\n1)Incident nature\n2)Location')

    return str(resp)

@app.route('/predict',methods=['POST'])
def predict():
    ad = joblib.load('model2.joblib')
    tv = joblib.load('tv.joblib')

    if request.method == 'POST':
        message = request.form['contact_message']
        data = [message]
```

Twilio

- Fully programming contact centre platform



Twilio - Sandbox



Heroku

- Try it out
  - https://still-tor-33663.herokuapp.com/
  - WhatsApp message to +1 415 523 8886 with join code:
    - **join last-general**

Recommendation

- Dataset with more relevant labels that attribute to situation of the message.
- Explore Deep Learning methods eg. BERT, ELMo, Glove.
    - It gives semantic understanding to the text
    - Further exploration of Topic modelling with semantic understanding.

Contents

- Data Cleaning
- EDA
- Modelling
- Deployment
- Recommendation

Problem Statement

- To classify disaster outreach messages and improve the triage system in Disaster Management Cycle.

Dataset

Dataset is retrieved from Figure 8.

It is multi-labelled with 30,000 messages drawn from earthquakes, floods and super-storms.

Data Cleaning

- Removal of Nulls and Duplicates

```
df = df.dropna()
df = df.drop_duplicates()
```

- Removal of wrong label

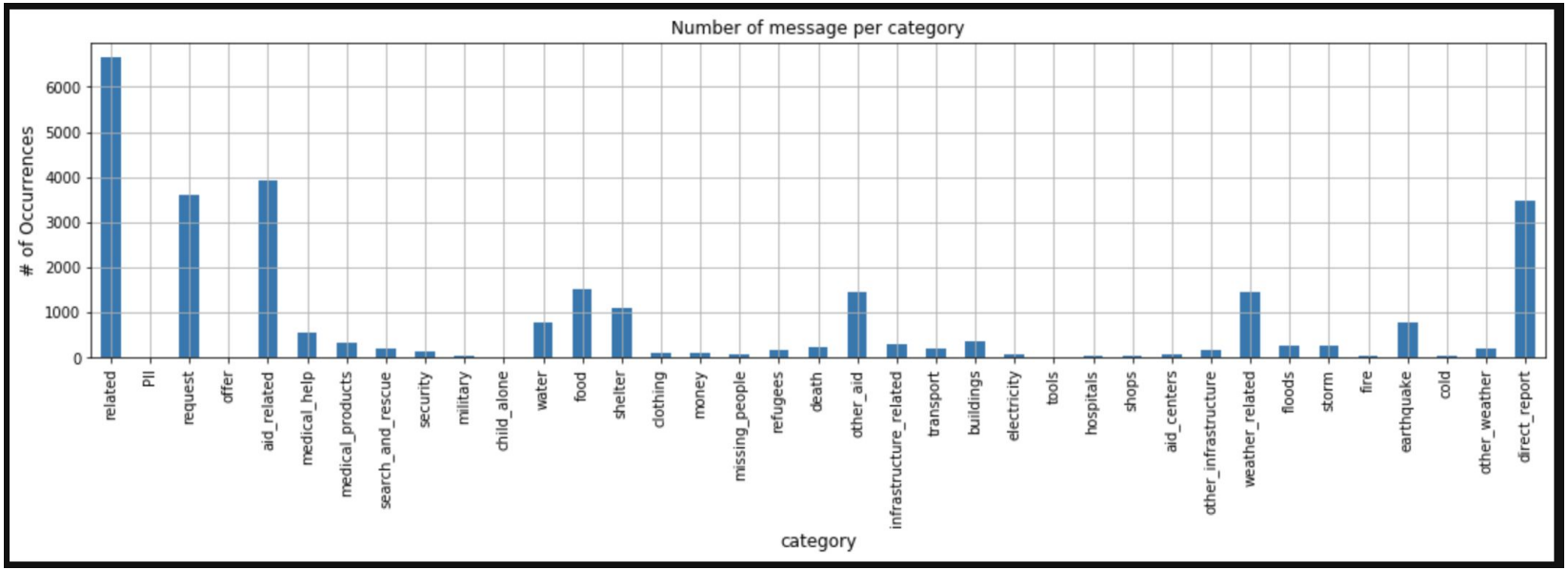
```
#Drop 193 rows of messages that mostly not properly translated
df = df[df.related != 2]
```

EDA

Exploratory Data Analysis

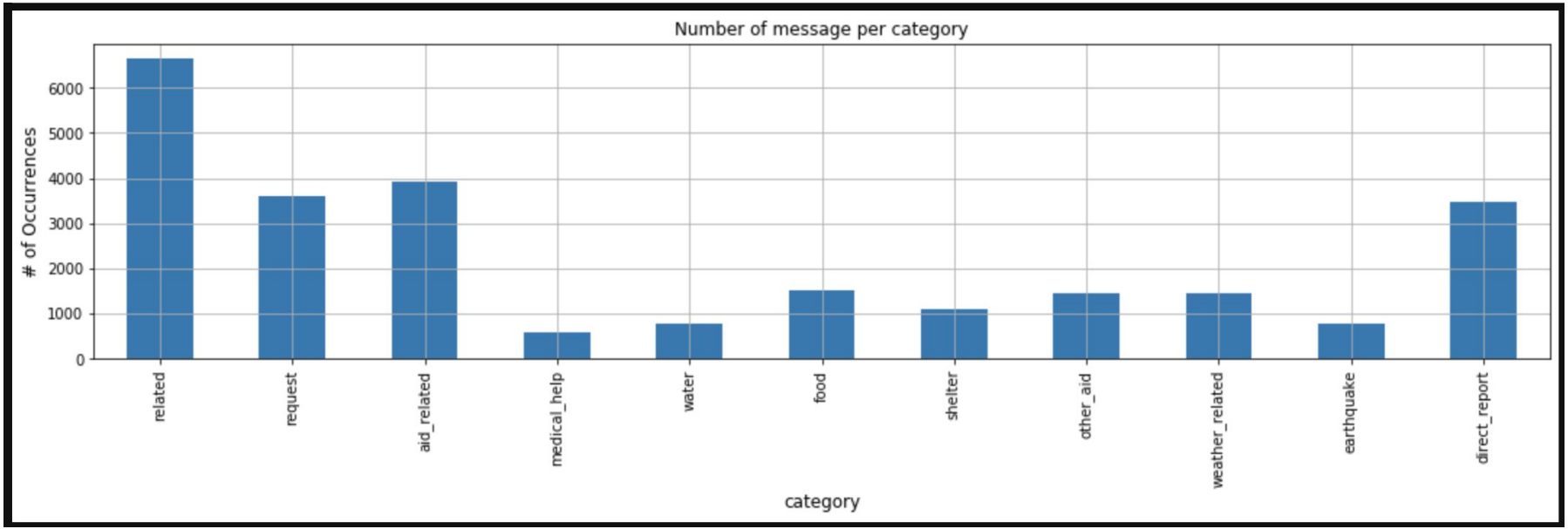
EDA

- Multi-Labels: 37 Labels



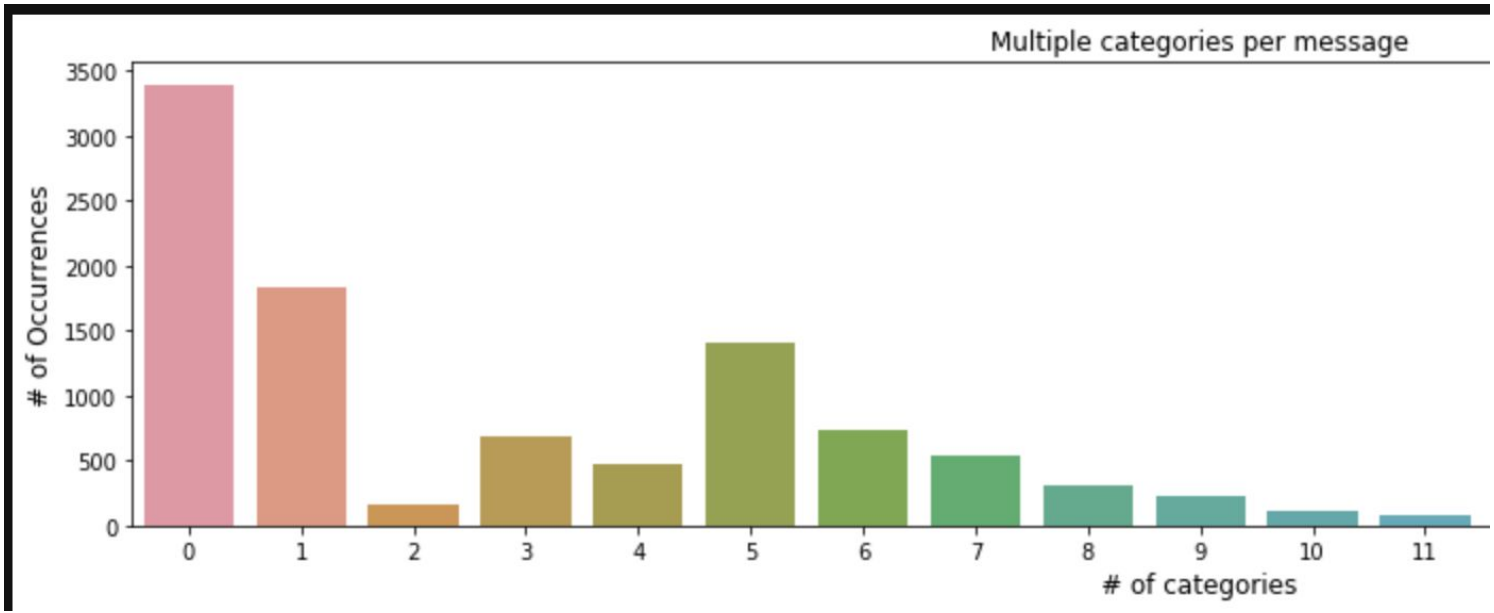
EDA

- Labels with more than 5% occurrence



EDA

- Messages with Multiple - Labels



EDA

- 3 Genres: Direct, News, Social

```
df[df['genre'] == 'direct'].head(5)
```

	id	split	message	original	genre	related
0	9	test	UN reports Leogane 80-90 destroyed. Only Hospital St. Croix functioning. Needs supplies desperately.	UN reports Leogane 80-90 destroyed. Only Hospital St. Croix functioning. Needs supplies desperately.	direct	1
1	39	test	We are at Gressier we needs assistance right away. ASAP, Come help us.	Se gressier nou an difikilte tanpri vin ede nou	direct	1
2	49	test	Delmas 33 in Silo, need water.	Delma 33 silo gen problem dlo	direct	1
3	79	test	SOS SOS, please provide police officers on the streets as they are very insecure	EMGENCY EMGENCY SI POLIS LA TE KA BAY PREZANS LI SOU CHAK PLAS AK LARI LI TAP BON PASKE CHAK SWA YO F ANPIL DEGA MENM DETWI LAVI MOUN ENPOTAN	direct	1
4	99	test	I am a driver, a mechanic .. I want to help	MWEN SE YON NMALIEN, CHOF, MEKANISYEN MWEN BEZWEN BAY SEKOU	direct	1

```
df[df['genre'] == 'news'].head(5)
```

	id	split	message	original	genre	related
1301	15739	test	Local administrative bodies are trained to quickly issue alerts and evacuation orders and distribute food and blankets at shelters.	NaN	news	1
1302	15749	test	As well, security threats from jihads remained.	NaN	news	1
1303	15759	test	- Deployment of mobile emergency health units to ensure families have immediate access to primary care and medical supplies, with the first team of 10 seasoned local emergency medical staffs deploying to the remote Swat Valley, which has been cut off due	NaN	news	1
1304	15769	test	PWJ will target 30 villages in Ingapu Township and plan to distribute equipment/tools for cleaning and repairing damaged houses, repair and improve	NaN	news	1

```
df[df['genre'] == 'social'].head(5)
```

	id	split	message	original	genre	related
981	11189	test	RT selenagomez UNICEF has just announced an emergency alert for the people of Haiti who were hit by a 7.0 earthquake and a tsunami.. ..	NaN	social	1
982	11199	test	RT YURFAVHALFBREED Moment of silence for those devastated by the earthquake in Haiti	NaN	social	1
983	11219	test	Earthquake in #Haiti? Does anyone know anything about this?	NaN	social	1
984	11249	test	RT OrphanProject Most of you have probably heard this but Haiti was hit by a 7.3 Magnitude earthquake at approx. 4 30pm today. .. htt ..	NaN	social	1
985	11259	test	In Punta Cana Dominican Republic for a friend s wedding. Earthquake that rocked Haiti was not felt here. Feel bad for those folks.	NaN	social	1

EDA - Conclusion

- Based on the EDA, 70% of the categories are less than 5% of the total data set.
- Remaining categories does not attribute to the severity of the outreach.
- We will focus on building a robust binary classification model (Disaster/Non-Disaster).

Data Preparation

Data - Preprocessing

- Understanding the dataset balance - *requires sampling*

```
df['related'].value_counts(normalize=True)
```

```
1    0.675014
0    0.324986
Name: related, dtype: float64
```

```
sm = SMOTETomek(random_state=42)
X_res, y_res = sm.fit_sample(X_train, y_train)
```

- Cleaning of text

```
def clean_text(text):
    # remove HTML tags and URLs
    text = re.sub(r'<.*?>', '', text)
    text = re.sub(r'^https?:\/\/.*[\r\n]*','',text)
    # keep only text without punctuation
    text = re.sub(r'[^\w\s]','',text)
    text = re.sub(" \d+", " ", text)
    # convert text to lowercase
    text = text.strip().lower()
    # split text into a list of words
    token_text = re.split('\W+',text) #W+ --> word chars and dashes permitted
    return token_text
```

```
stop_words = stopwords.words('english')
stop_words
```

```
lemmatizer = WordNetLemmatizer()
def lemmatize_words(text):
    lemma_words = [lemmatizer.lemmatize(word) for word in text]
    return lemma_words
```

Modelling

Models and hyper parameter tuning

- Logistic Regression
- NavieBayers
- Random Forest
- Extra Trees
- Decision Trees
- Bagged Decision Trees
- AdaBoost

```
classifier_model_params = {
    'LogisticRegression' : {
        'penalty' : ['l1', 'l2'],
        'C' : np.arange(.05, 1, .05) },
    'KNN' : {
        'n_neighbors' : np.arange(3, 22, 2) },
    'NaiveBayes' : {
        'alpha' : np.arange(.05, 2, .05)},
    'DecisionTree': {
        'max_depth' : [ 6, 10, 14],
        'min_samples_leaf' : [1, 2],
        'min_samples_split': [2, 3] },
    'BaggedDecisionTree' : {
        'n_estimators' : [20, 60, 100] },
    'RandomForest' : {
        'n_estimators' : [20, 60, 100],
        'max_depth' : [ 2, 6, 10],
        'min_samples_split' : [2, 3, 4] },
    'ExtraTrees' : {
        'n_estimators' : [20, 60, 100],
        'max_depth' : [ 6, 10, 14],
        'min_samples_leaf' : [1, 2],
        'min_samples_split' : [2, 3], },
    'AdaBoost' : {
        'n_estimators' : np.arange(100, 151, 25),
        'learning_rate' : np.linspace(0.05, 1, 20) },
    'GradientBoosting' : {
        'n_estimators' : np.arange(5, 150, 10),
        'learning_rate' : np.linspace(0.05, 1, 20),
        'max_depth' : [1, 2, 3] },
    'XGBoost' : {
        'n_estimators'  : np.arange(100, 151, 25),
        'learning_rate' : np.arange(0.1, 1, .3),
        'max_depth' : [3],
        'alpha' : np.arange(0, 1, .3),
        'lambda' : np.arange(0, 1, .3),
        'gamma' : np.arange(0, 1, .3),
        'subsample' : [.5],
```

Models Selection

- Ada Boost
  - Lowest train-test difference : Approx. 3%
  - Accuracy of 81%

	Model Name	Best Params	Best Score	Train Score	Test Score
2	NaiveBayes	{'alpha': 0.05}	0.850435	0.921147	0.783887
6	BaggedDecisionTree	{'n_estimators': 100}	0.844093	0.997349	0.787705
0	LogisticRegression	{'C': 0.9500000000000001, 'penalty': 'l2'}	0.841253	0.891424	0.811378
4	ExtraTrees	{'max_depth': 14, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}	0.815505	0.862268	0.772814
7	AdaBoost	{'learning_rate': 0.7999999999999999, 'n_estimators': 150}	0.813518	0.838413	0.799924
3	RandomForest	{'max_depth': 10, 'min_samples_split': 2, 'n_estimators': 100}	0.807838	0.838697	0.776632
5	DecisionTree	{'max_depth': 14, 'min_samples_leaf': 1, 'min_samples_split': 2}	0.773855	0.805566	0.723559
1	KNN	{'n_neighbors': 3}	0.655718	0.770636	0.565101

Deployment

Flask - Heroku

Twilio - Sms/Whatsapp

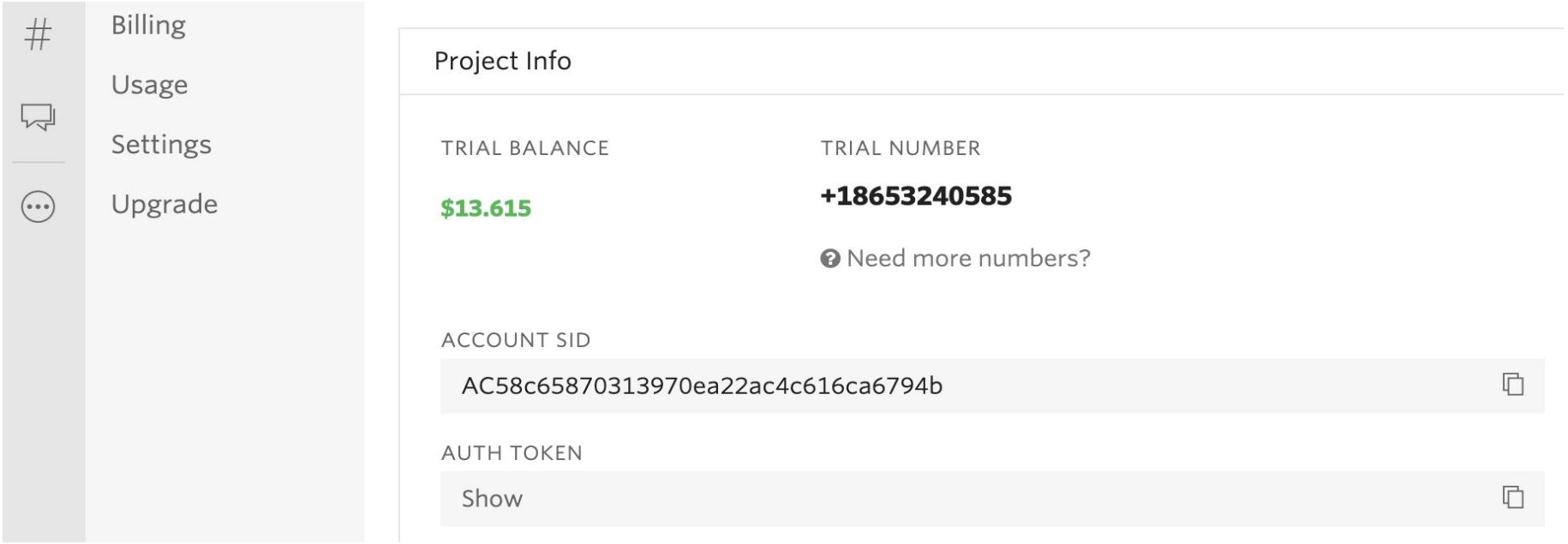
Flask

- Static
- Templates
- Profile
- Requirements
- Pre-train models
- Runtime
- app.py

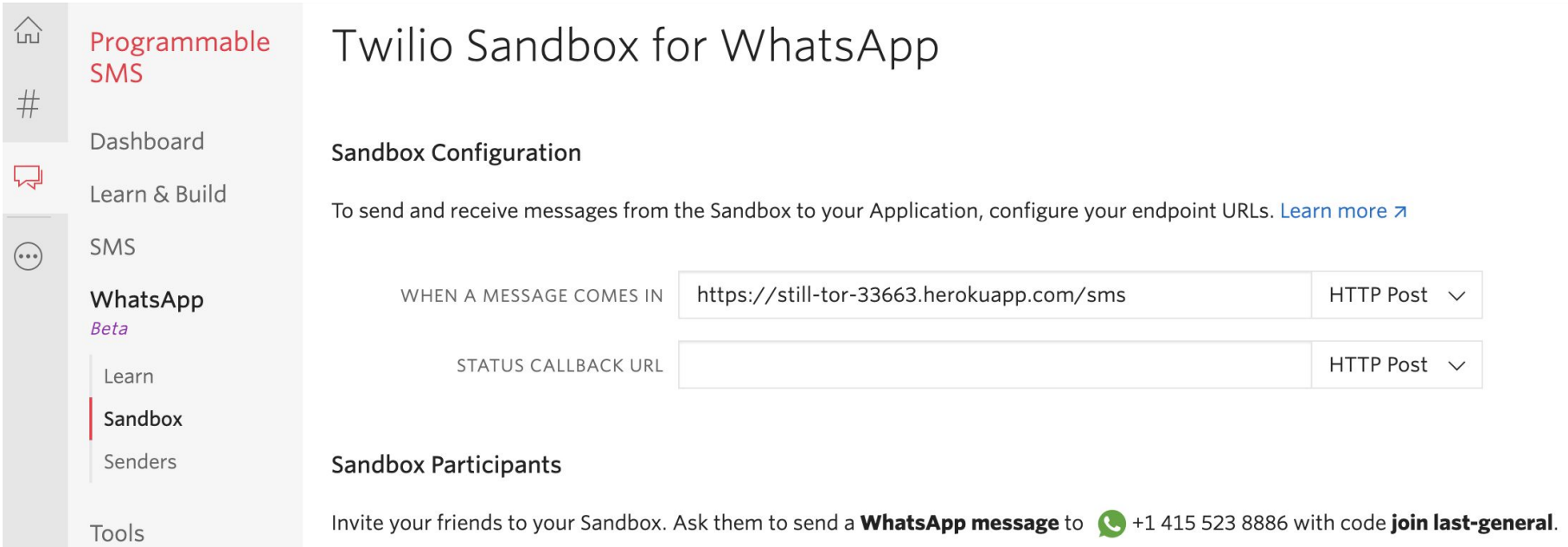
```
app = Flask(__name__, static_url_path='/static')

@app.route('/', methods=['GET', 'POST'])
def home():
    return render_template('index.html')

@app.route('/sms', methods=['POST'])
def sms_reply():
    resp = MessagingResponse()
    #Fetch message
    msg = request.form.get('Body')

    #create reply
    if len(msg) >20:
        #resp.message("Accessing input..")
        #setup of model
        ad = joblib.load('model2.joblib')
        tv = joblib.load('tv.joblib')
        #load text to model
        data = [msg]
        vect = tv.transform(data).toarray()
        my_prediction = ad.predict(vect)
        if my_prediction == 1:
            resp.message('We have received your distress call.\nShare your exact location\nWe will send a pony over')
        else:
            resp.message('Thank you for reaching out.\nPlease do not waste your time and our precious time thank you.')
    else:
        #resp.message("Accessing input..")
        resp.message('Furnish us with following information\n1)Incident nature\n2)Location')

    return str(resp)

@app.route('/predict',methods=['POST'])
def predict():
    ad = joblib.load('model2.joblib')
    tv = joblib.load('tv.joblib')

    if request.method == 'POST':
        message = request.form['contact_message']
        data = [message]
```

Twilio

- Fully programming contact centre platform



Twilio - Sandbox



Heroku

- Try it out
  - https://still-tor-33663.herokuapp.com/
  - WhatsApp message to +1 415 523 8886 with join code:
    - **join last-general**

Recommendation

- Dataset with more relevant labels that attribute to situation of the message.
- Explore Deep Learning methods eg. BERT, ELMo, Glove.
    - It gives semantic understanding to the text
    - Further exploration of Topic modelling with semantic understanding.