Fake News Detector

Agenda

1

Introduction & Business Case

Background

Business Model

2

Solution Design & Implementation

Modeling Framework

Testing Results

3

Future goals

Introduction & Business Case

- ➔ Background
- ➔ Business Model

Solution Design & Implementation

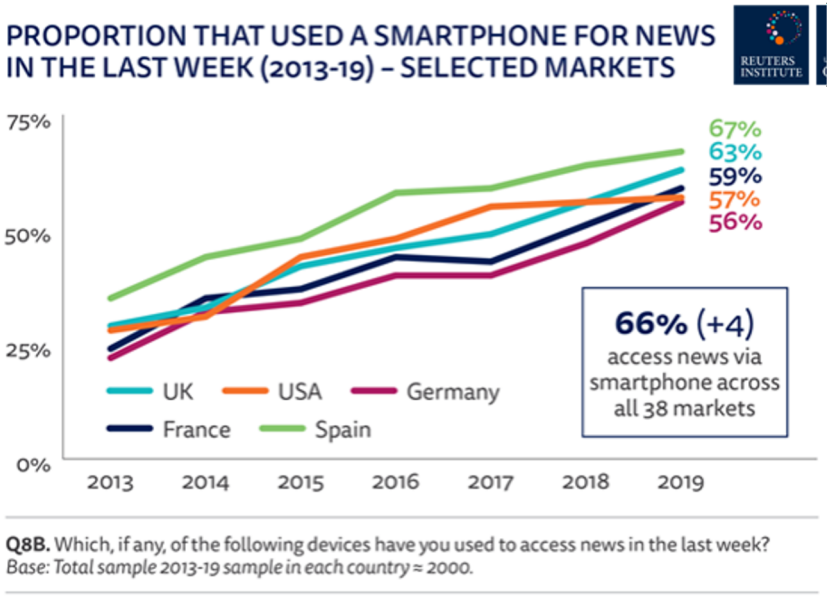
- ➔ Modeling Framework
- ➔ Testing Results

Why the need for fake detector?

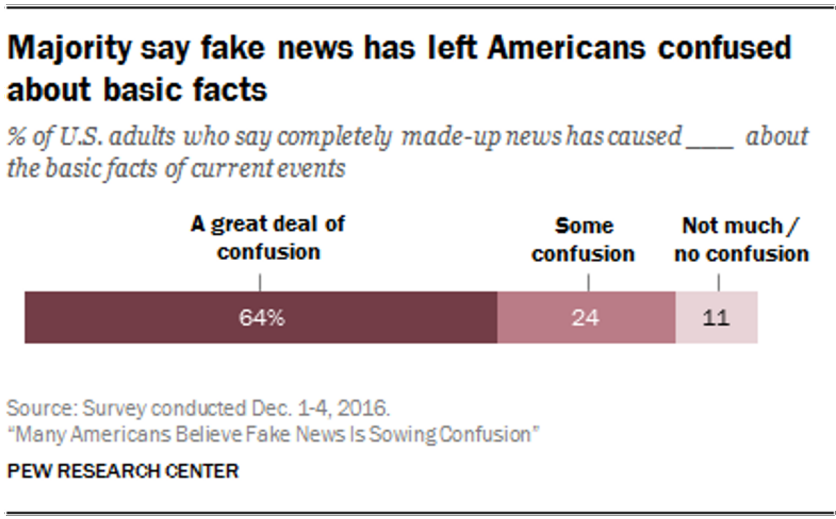
Fake news, defined by the New York Times as “a made-up story with an intention to deceive”

THREE KEY TRENDS are driving a real need for a tool to detect and alert fake news...

1. Increasing Consumption of Digital Mass Media & Social Media



2. Increasing Instances of Fake News creating a “Great Deal of Confusion”



3. Increased Uncertainty in the World

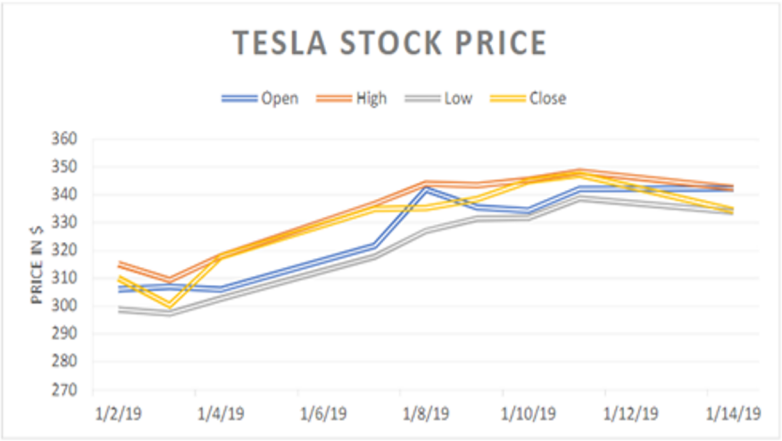
- **2020 US Election Campaign**
- **COVID-19 pandemic**
- **Economic Recession**

Real need for a tool to detect and alert Fake News

1. Source: Reuters study on Digital news consumption
2. Source: Pew Research Center Study

Fake news brings serious damage: Tesla's Case

- On January 7th, the day before the CES 2019 convention described as a "Global Stage for Innovation", a video went viral on twitter.
- The video alleged to show a Tesla autonomous driving car crashing into a robot prototype at the CES convention.
- This video is proved to be completely fake.



HUGE LOSS:

Tesla's stock price dropped due to the fake video!!!!

Another Cases

SCENARIO 1: India's fake news problem on WhatsAPP

BBC News analysis:

31

India reported at least murders in 2017 and 2018, triggered by DISINFORMATION on SOCIAL MEDIA platforms.

SCENARIO 2: Coronavirus

Coronavirus: Fake news is spreading fast

Rory Cellan-Jones
Technology correspondent
@BBCRoryCJ

Misinformation came with the COVID on social media.

- **Rumor: drinking alcohol provided a level of protection against infection.**
- **Result: It was even reported that deaths were caused by people consuming industrial alcohol to try and disinfect themselves.**

How fake news detector works ?

Fake News Detector: What Is It?

MAIN IDEA: built a NLP & machine learning based model that will determine that an article is fake, using it to classify a news article into one of the four categories: agree, disagree, discusses and unrelated

Digital News Articles

Blogs /Forums

Social Media

Journals

Feed

NLP & Machine Learning Engine

classify

Agree

Disagree

Discusses

Unrelated

Text data trained to the NLP ML model

Detect fake news only when they are under-written, for instance when the content is totally unrelated to the headline (so-called “clickbait”)

Introduction & Business Case

- ➔ Background
- ➔ Business Model

Solution Design & Implementation

- ➔ Modeling Framework
- ➔ Testing Results

Dataset

train_bodies.csv - shape (1683, 2): contains the article body column ('Body ID') with corresponding body text of article ('article Body') column.

train_stances.csv - shape (49952, 3): contains article headline ('Headline') for pairs of article body ('Body ID'), and labeled stance ('Stance') columns.

train_stances

Headline	Body ID	Stance
Police find mass graves with at least '15 bodies' near Mexico town where 43 students disappeared after police clash	712	unrelated
Hundreds of Palestinians flee floods in Gaza as Israel opens dams	158	agree
Christian Bale passes on role of Steve Jobs, actor reportedly felt he wasn't right for part	137	unrelated
HBO and Apple in Talks for $15/Month Apple TV Streaming Service Launching in April	1034	unrelated
Spider burrowed through tourist's stomach and up into his chest	1923	disagree
'Nasa Confirms Earth Will Experience 6 Days of Total Darkness in December' Fake News Story Goes Viral	154	agree
Accused Boston Marathon Bomber Severely Injured In Prison, May Never Walk Or Talk Again	962	unrelated

train_bodies

Body ID	articleBody
0	A small meteorite crashed into a wooded area in Nicaragua's capital of Managua overnight, the government said Sunday. Residents reported hearing a mysterious boom that left a 16-foot deep crater near the city' Government spokeswoman Rosario Murillo said a committee formed by the government to study the event determined it was a "relatively small" meteorite that "appears to have come off an asteroid that was pass Murillo said Nicaragua will ask international experts to help local scientists in understanding what happened. The crater left by the meteorite had a radius of 39 feet and a depth of 16 feet, said Humberto Saballos, a volcanologist with the Nicaraguan Institute of Territorial Studies who was on the committee. He said it is stil Humberto Garcia, of the Astronomy Center at the National Autonomous University of Nicaragua, said the meteorite could be related to an asteroid that was forecast to pass by the planet Saturday night. "We have to study it more because it could be ice or rock," he said. Wilfried Strauch, an adviser to the Institute of Territorial Studies, said it was "very strange that no one reported a streak of light. We have to ask if anyone has a photo or something." Local residents reported hearing a loud boom Saturday night, but said they didn't see anything strange in the sky. "I was sitting on my porch and I saw nothing, then all of a sudden I heard a large blast. We thought it was a bomb because we felt an expansive wave," Jorge Santamaria told The Associated Press. The site of the crater is near Managua's international airport and an air force base. Only journalists from state media were allowed to visit it.

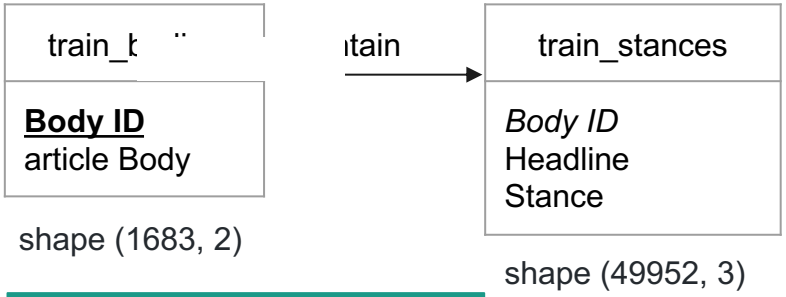
Data source

http://www.fakenewschallenge.org/

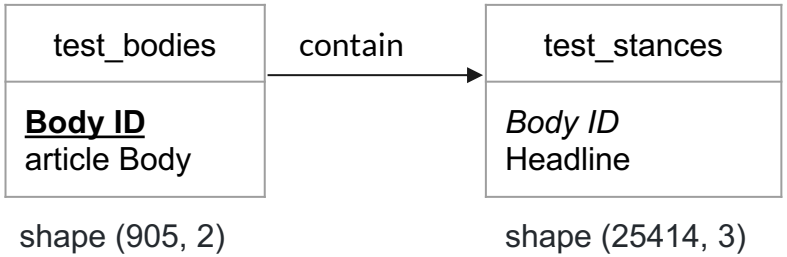
Source: The data is created by accredited journalists (derived from the Emergent Dataset created by Craig Silverman), making it both high quality and credible. It is also in the public domain.

Dataset - summary

Training Sets



Testing Sets



Headline:

- Short
- describes the main topic of the article

Body Text ("article Body"):

- elaborates the details
- highlights and shapes the perspectives

The distribution of output labels (agree, disagree, discuss, unrelated) may not be the same in the test set as the training set, since real world does not follow independent and identically distributed assumptions.

Data source: the Emergent Dataset created by Craig Silverman

Formal definition

- Input: A headline and a body text
- Output : classify into 4 categories: agree, disagree, related, unrelated

Agrees	The body text agrees with the headline
Disagrees	The body text disagrees with the headline
Discusses	The body text discusses the same topic as the headline, but does not take a position
Unrelated	The body text discusses a different topic than the headline

What do agree, disagree, discuss, and unrelated mean?

WITH THE SAME TOPIC OF "the famous band Led Zeppelin's Robert Plant turning down a huge sum of money for the band's reunion", here the different classifications are:

"... Led Zeppelin's Robert Plant turned down £500 MILLION to reform supergroup. ..."

CORRECT CLASSIFICATION: AGREE

"... No, Robert Plant did not rip up an $800 million deal to get Led Zeppelin back together. ..."

CORRECT CLASSIFICATION: DISAGREE

"... Robert Plant reportedly tore up an $800 million Led Zeppelin reunion deal. ..."

CORRECT CLASSIFICATION: DISCUSSES

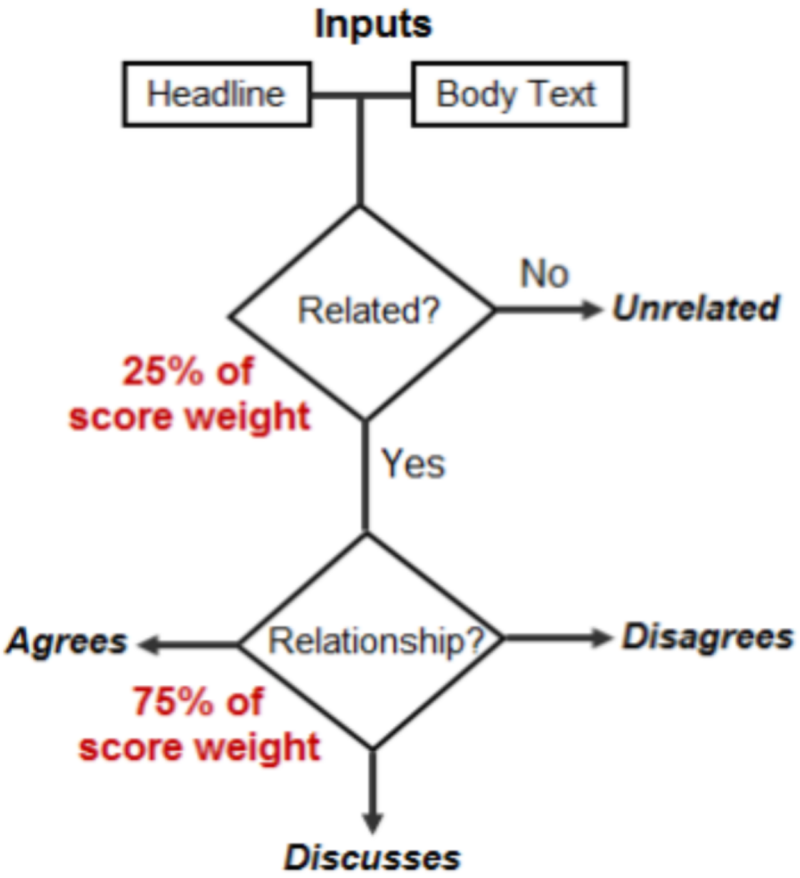
"... Richard Branson's Virgin Galactic is set to launch SpaceShipTwo today. ..."

CORRECT CLASSIFICATION: UNRELATED

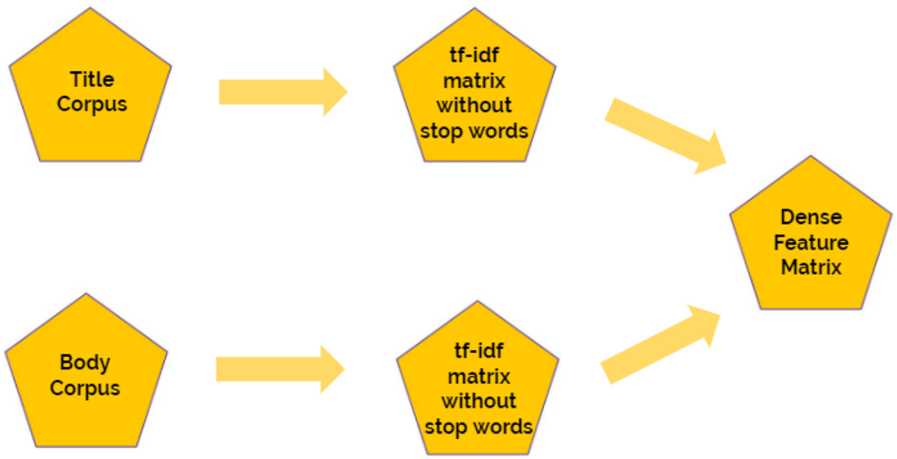
Evaluation metrics

A weighted, two-level scoring system

Level 1	**Classify headline and body text as related or unrelated: 25% score weighting**
Level 2	**Classify related pairs as agrees, disagrees, or discusses: 75% score weighting**



Feature Engineering



Work overlap features

Number of words appearing in the headline as well as body

Refuting features

Calculate if that particular refuting word appears in the headline (15 refuting features for 15 refuting words).

Polarity features

Calculate the total numbers of refuting words in headline and body, then divided by 2

Hand features

Counts how many times a “token”/n-gram of the title appears in the news body, ignoring stop words

Feature Engineering

Work overlap features

Number of words appearing in the headline as well as body

```
def word_overlap_features(headlines, bodies):
    X = []
    for i, (headline, body) in tqdm(enumerate(zip(headlines, bodies))):
        clean_headline = clean(headline)
        clean_body = clean(body)
        clean_headline = get_tokenized_lemmas(clean_headline)
        clean_body = get_tokenized_lemmas(clean_body)
        features = [
            len(set(clean_headline).intersection(clean_body)) / float(len(set(clean_headline).union(clean_body)))]
        X.append(features)
    return X
```

Multi-class Perceptron Model

Perceptron (used in supervised learning) is a single layer neural network (a multi-layer perceptron is called Neural Networks)

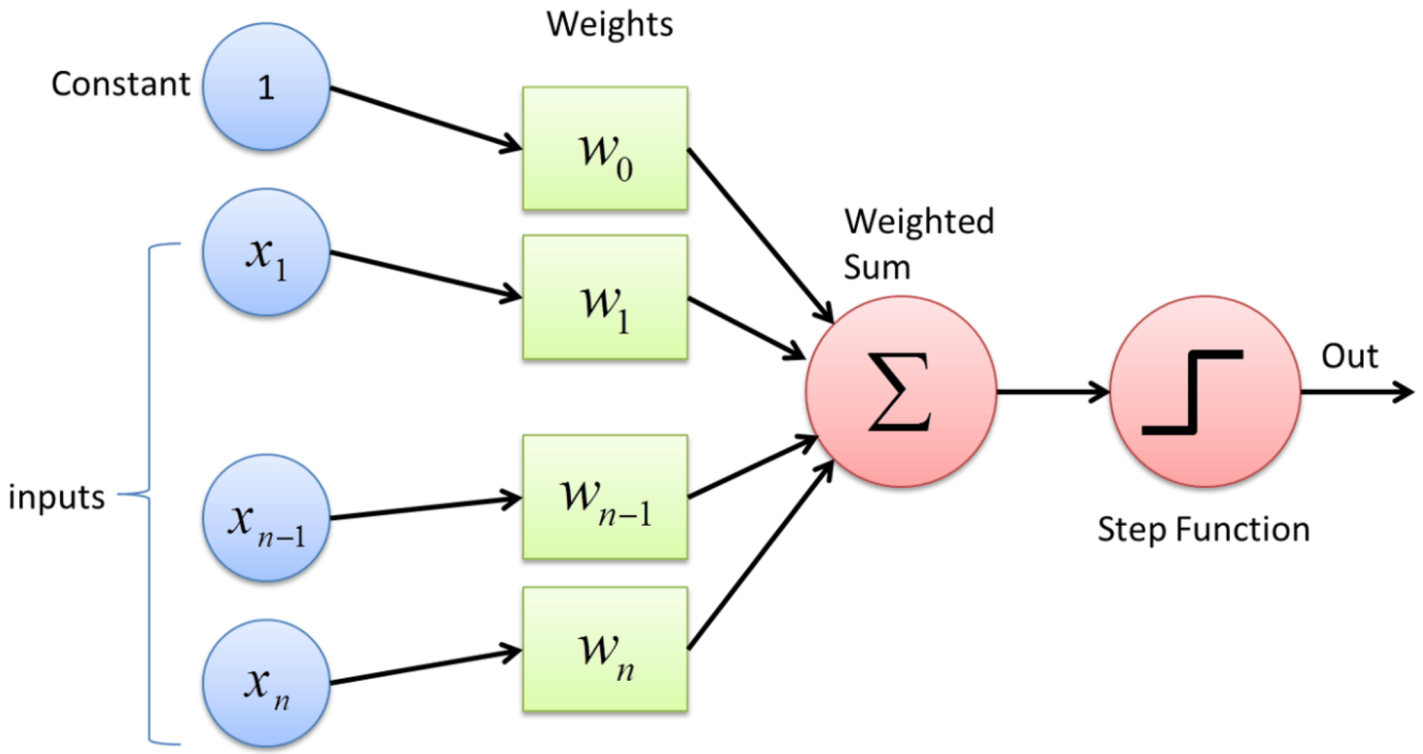


Fig : Perceptron

Multi-class Perceptron Model

A feature representation function f(x,y) maps each possible input/output pair (input x, output y) to a finite-dimensional real-valued feature vector.

The feature vector is multiplied by a weight vector w, but the resulting score is used to choose among many possible outputs:

$$\hat{y} = \operatorname{argmax}_y f(x, y) \cdot w.$$

Learning again iterates over the examples, predicting an output for each, leaving the weights unchanged when the predicted output matches the target, and changing them when it does not. The update becomes:

$$w_{t+1} = w_t + f(x, y) - f(x, \hat{y}).$$

Multi-class Perceptron Model

```
def predict(base_features,weights,labels):
    '''
    prediction function
    :param base_features: a dictionary of base features and counts
    :param weights: a defaultdict of features and weights. features are tuples (label,base_feature).
    :param labels: a list of candidate labels
    :returns: top scoring label, scores of all labels
    :rtype: string, dict
    '''
    scores = {}
    for label in labels:
        fv = make_feature_vector(base_features,label)
        scores[label] = 0
        if len(fv) > len(weights):
            for (x, y) in weights:
                if x == label:
                    scores[x] += weights[x, y] * fv.get((x, y), 0)
        else:
            for (x, y) in fv:
                if x == label:
                    scores[x] += fv[x, y] * weights.get((x, y), 0)

    return argmax(scores),scores
```

$$\hat{y} = \text{argmax}_y \, f(x, y) \cdot w.$$

Multi-class Perceptron Model

```
def perceptron_update(x,y,weights,labels):
    '''
    compute the perceptron update for a single instance
    :param x: instance, a counter of base features and weights
    :param y: label, a string
    :param weights: a weight vector, represented as a dict
    :param labels: set of possible labels
    :returns: updates to weights, which should be added to weights
    :rtype: defaultdict
    '''
    y_hat,_ = predict(x,weights,labels)
    update = defaultdict(float)
    if y != y_hat:
        f_xy = make_feature_vector(x,y)
        f_xy_hat = make_feature_vector(x,y_hat)

        for key in f_xy:
            update[key] = f_xy[key] - f_xy_hat.get(key, 0)
        for key in f_xy_hat:
            update[key] = -(f_xy_hat[key] - f_xy.get(key, 0))
    return update
```

$$w_{t+1} = w_t + f(x, y) - f(x, \hat{y}).$$

Multi-class Perceptron Model

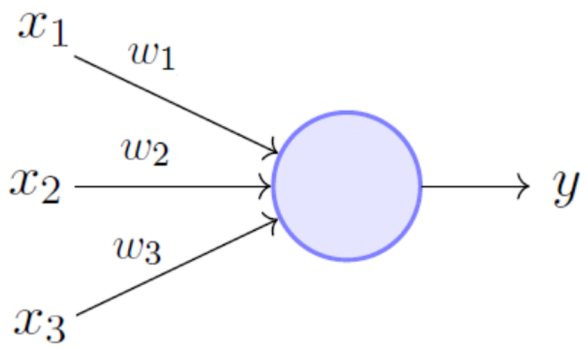
```
def estimate_perceptron(x,y,N_its):
    '''
    estimate perceptron weights for N_its iterations over the dataset (x,y)
    :param x: instance, a counter of base features and weights
    :param y: label, a string
    :param N_its: number of iterations over the entire dataset
    :returns: weight dictionary
    :returns: list of weights dictionaries at each iteration
    :rtype: defaultdict, list
    '''

    labels = set(y)
    weights = defaultdict(float)
    weight_history = []
    for it in range(N_its):
        for x_i,y_i in zip(x, y):
            update = perceptron_update(x_i,y_i,weights,labels)
            for key in update:
                weights[key] += update[key] #+ weights.get(key, 0)
                if update[key] == 0:
                    del weights[key]
        weight_history.append(weights.copy())

    return weights, weight_history
```

Multi-class Perceptron Model

In our model, we have performed 10-fold cross validation on the input to the perceptron and recorded its accuracy after each fold



Perceptron Model (Minsky-Papert in 1969)

```
Reading dataset
Total stances: 49972
Total bodies: 1683
Reading dataset
Total stances: 25413
Total bodies: 904
Evaluating ...
Score for fold 6 was - 0.6416259695489802
Evaluating ...
Score for fold 0 was - 0.7233390536896298
Evaluating ...
Score for fold 7 was - 0.7591571206335738
Evaluating ...
Score for fold 5 was - 0.7098674521354934
Evaluating ...
Score for fold 2 was - 0.4501308519918581
Evaluating ...
Score for fold 8 was - 0.7386666666666667
Evaluating ...
Score for fold 9 was - 0.6128942227183924
Evaluating ...
```

Multi-class Perceptron Model

Validation Set Confusion Matrix

Scores on the dev set

	agree	disagree	discuss	unrelated
agree	0	3	704	55
disagree	0	0	158	4
discuss	0	13	1674	113
unrelated	0	2	1241	5655

Score: 3307.25 out of 4448.5 (74.34528492750366%)

Test Set Confusion Matrix

Scores on the test set

	agree	disagree	discuss	unrelated
agree	0	10	1726	167
disagree	0	10	551	136
discuss	1	25	4014	424
unrelated	0	26	2885	15438

Score: 8461.75 out of 11651.25 (72.62525480098702%)
Out[32]: 72.62525480098702

Feedforward Neural network

Additionally,

Validation Set Confusion Matrix

Scores on the dev set

	agree	disagree	discuss	unrelated
agree	445	10	259	48
disagree	28	78	49	7
discuss	93	30	1565	112
unrelated	13	1	76	6808

Score: 3907.25 out of 4448.5 (87.8329774081151%)

Test Set Confusion Matrix

Scores on the test set

	agree	disagree	discuss	unrelated
agree	599	3	987	314
disagree	127	9	302	259
discuss	320	4	3476	664
unrelated	59	1	431	17858

Score: 8984.25 out of 11651.25 (77.10975217251368%)