Text Analytics

**Corpus Text Preprocessing**

Für das Text Preprocessing haben wir eine Funktion geschrieben, wo wir dann festlegen konnten welche Cleaning Schritte ausgeführt werden sollten.

So konnten wir aus folgenden Cleaning Schritte auswählen.

* Lowercase
* Stopwords
* Zahlen entfernen
* Punktuation
* Special chars
* 1 char
* Stemming
* Lemmatisierung

So konnten wir verschiedene Sachen ausprobieren.

**Hyperparameter Tuning**

Nachdem Text Preprocessing haben wir dann für das Hyperparameter Tuning und evalierung der Performance der Modelle die gridsearchCV() Funktion von Sklearn verwendet.

Hierfür haben wir für jeden Algorithmus ein Parameter Grid definiert um die festgelegten Parameter zu tunen und mit Cross-Validation mit 5-Fold haben wir dann die Güte des trainierten Models bestimmt.

Für das Trainieren haben wir 80% der Daten verwendet und für das anschliessende Testen 20% der Daten.

Die Resultate haben wir alle automatisch abgespeichert.

**Modell Vorhersagen**

**Folie 1**

Für die Vorhersage auf den Testdaten haben wir verschiedene Modelle getestet. Als Baseline haben wir verschiedene Sklearn Algorithmen getestet mit Bag of Words und TF-IDF.

Verglichen haben wir es dann mit Word Embeddings und Fasttext Modellen.

**Folie 2**

Die Resultate für die Sklearn Algorithmen. Wir haben RandomForest, MultinomialNB, Linear Support Vector machine und LogisticRegression ausprobiert.

Auf der Linken Seite seht ihr die Accuracy ohne Text preprocessing also auf den Rohdaten und ohne Hyperparameter Tuning.

Auf der Rechten Seite sind die Resultate mit Text Preprocessing und hyperparameter tuning.

Die Grünen Balken zeigt die Accuracy für Bag of Words und Blau für TF-IDF

Wir ihr sehen könnt, sind die Resultate ohne Text Preprocessing auch sehr hoch und lustiger weise haben wir die höchste accuracy mit LinearSVC ohne text preprocessing erreicht. 97.5%.

Es scheint so als würde Linear SVC die Grenzen besser mit den Rohdaten ziehen zu können als mit Text preprocessing.

Ansonsten für die andren Modellen haben wir ungefähr 1% höhere accuracy mit text preprocessing und hyperparameter tuning.

Hier ist die LogisticRegression mit 97.1% am besten.

Hier muss auch noch angemerkt werden, dass die Berechnungszeit bedeutend kleiner war.

**Folie 3**

Anschliessend haben wir versucht die Baseline Modellen mit den Neuronale netzten mit Embedding layer und Fasttext zu vergleichen.

Für die Embedding Layer haben wir die vortrainierten word embeddings aus den Übungen verwendet und auch versucht die word embeddings selber zu trainieren.

Die beste Accuracy mit 97.8% haben wir mit dem Neuronalem Netz mit den vortraineirten word embeddings welche wir noch weiter trainiert haben, erzielt.

Das ist ein höherer Wert als bei den basline Modellen.

Aber auch dir word embeddings von scratch zu lernen hat eine hohe accuracy erreicht. Fasttext hat leider nicht so gut performt.

**Wrong Predictions**

**Folie 1**

Wir haben uns dann mal die Wrong Predictions genauer angeschaut und festgestellt, das vorallem Business Artikel falsch vorhergesagt werden. Bei MultionmialNB, LSVC und LogistigRegression machen dies den grössten Teil aus und die vorhersage war immer entweder Politics oder Tech.

Vergleicht man dies mit den Word Emebddings so sieht man das die Business Artikel besser vorhergesagt werden aber immernoch den grössten Teil aus machen.

Innerhalb der Word embeddings werden die selbern Business Artikel falsch vorhergesagt.

Auch hier ist es entweder Politics oder Tech.

**Folie 2**

* André