Hallo zusammen,

Zunächst möchte ich mich für die Einladung zur Präsentation bedanken. Die heutige Präsentation trägt den Titel eines Datenfalls: AI Solution for Help Desk Inc. Die Präsentation findet auf englischer und auf deutscher Sprache statt / gehalten.

Und ich möchte mich kurz vorstellen. Mein Name ist Alvin Rindra Fazrie, ich habe gerade mein Masterstudium abgeschlossen und ich habe die Erfahrung gemacht, sowohl AI- als auch Nicht-AI-Apps zu entwickeln, und ein Enthusiast von Data Science.

Business Case

The web contains more and more user genereated data. One huge source of such information can be found in the myriads of customer written complaints and reviews.

The aim of this data challenge is to show that there is value in this unstructured text data. Can you convince the C-Suite of Helpdesk Inc. a helpdesk and support company that there is value in working with the unstructured text data in their tickets and support request by applying state-of-the-art deep learning models? In a first meeeting you want to convince them, that a neural net can learn to distinguish between the relevant topics and problems talked about in a customer complaint.

Is it possible to train a neural net which can be used to predict the Product or Issue category? Feel free to focus on the categories with the highest impact. Another helpful information would be an accurate prediction if a complaint is resolved in a timely manner and if the consumer will dispute the decision. Depending on that prediction, tickets with a high probability of disputes or late resolves could be handled with specialized processes.

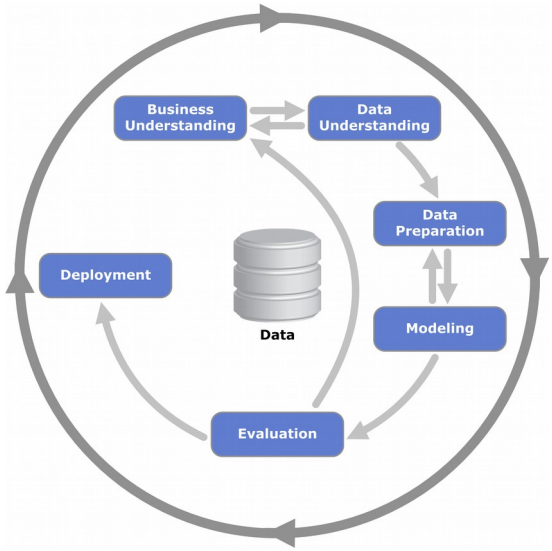
Grundsätzlich müssen wir die C-Suite von Helpdesk Inc. überzeugen, um Einblicke aus unstrukturierten Daten zu erhalten und die "State-of-the-Art-Deep-Learning" -Modelle zu implementieren.

Und Ja, das ist möglich, um aus diesen unstrukturierten Textdaten "Deep Learning" zu trainieren, also wir können um die Produkt- oder Problemkategorie vorherzusagen/ prognosen. Und dann wenn eine complaints rechzeitig aufgelöst/entschlossen wird und die ‘consumer’/Konsument die Entscheidung anficht/disputieren. Mit Deep Learning, wir können diese Problem/Prozess automatisieren.

Basically, we need to convince the C-suite of Helpdesk Inc. to get the insight from unstructured data and implement the 'state-of-the-art deep learning' models.

And Yes, that is possible to train "deep-learning" from this unstructured text data to predict Product or Issue category. And then if a complaint is resolved in a timely manner and if the consumer will dispute the decision.

CRISP-DM Methodology



*https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining

Bevor wir weiter gehen und die Implementierung besprechen. Ich möchte Ihnen die CRISP-DM-Methode zeigen. CRISP-DM stands for 'Cross-industry-standard-for-data-mining'.

Es gibt viele Ansätze/Methodik im Data Science-Prozess. Aber sie sind jedoch dieser CRISP-DM-Methode ziemlich ähnlich/fast gleich.

Es beginnt mit … danach, weiter, endlich..

Und wir wissen, dass dieser Data Science-Prozess iterativ ist. Andrew Ng sagt dass auch. ML is ein iterative prozess und Don't expect it to work first time. Erwarten / glauben wir nicht, dass es das erste Mal funktioniert.

Laut/ according Andrew ng, es gibt 3 Hauptprozess in ML.
1. (Idea → Code → Experiment) → Idea, zum, ….

Okay, fangen wir mit Business Understanding.

Business Understanding

- **Task:**

  To convince the C-Suite of Helpdesk Inc. to apply state-of-the-art deep learning models to solve the problems from unstructured text data.

- **Plan:**

  To enable a simpler ticket submission system which does not have the consumer to click through a complicated form and fill several fields.

- **Analytic Approach:**

  Multi-class Classification for the Product Classifier and Binary Classification for the Consumer Dispute Classifier

Geschäftsverständnis

Dies ist der Beginn / der Anfang des Prozesses. Business Understanding, wir müssen über Task, Plan und Analytic Approach verstehen.

Die Aufgabe / Task ist, C-suite des Helpdesk Inc. überzeugen um Deep learning zu implementieren als die als Lösung des Problems unstrukturierter Textdaten.

Das Konzept Plan ist, um eine einfachere Ticketübergabe System zu ermöglichen / implementieren. Also, die konzument muss sich keiner komplizierten Form stellen. Viele input füllen.

Die Analytic Approach: ...

Data Understanding

- **Data Collection**
- **Exploratory Data Analysis**

```
# Data Collection
df = pd.read_csv('../input/Consumer_Complaints.csv')
print(df.shape)
df.head()
```

```
(972147, 18)
```

	Date received	Product	Sub-product	Issue	Sub-issue	Consumer complaint narrative	Company public response	Company	State	ZIP code	Tags	Consumer consent provided?	Submitted via	Date sent to company	Company response to consumer	Timely response?	Consumer disputed?	Complaint ID
0	03/12/2014	Mortgage	Other mortgage	Loan modification,collection,foreclosure	NaN	NaN	NaN	M&T BANK CORPORATION	MI	48382	NaN	NaN	Referral	03/17/2014	Closed with explanation	Yes	No	759217
1	10/01/2016	Credit reporting	NaN	Incorrect information on credit report	Account status	I have outdated information on my credit repor...	Company has responded to the consumer and the ...	TRANSUNION INTERMEDIATE HOLDINGS, INC.	AL	352XX	NaN	Consent provided	Web	10/05/2016	Closed with explanation	Yes	No	2141773
2	10/17/2016	Consumer Loan	Vehicle loan	Managing the loan or lease	NaN	I purchased a new car on XXXX XXXX. The car de...	NaN	CITIZENS FINANCIAL GROUP, INC.	PA	177XX	Older American	Consent provided	Web	10/20/2016	Closed with explanation	Yes	No	2163100
3	06/08/2014	Credit card	NaN	Bankruptcy	NaN	NaN	NaN	AMERICAN EXPRESS COMPANY	ID	83854	Older American	NaN	Web	06/10/2014	Closed with explanation	Yes	Yes	885638
4	09/13/2014	Debt collection	Credit card	Communication tactics	Frequent or repeated calls	NaN	NaN	CITIBANK, N.A.	VA	23233	NaN	NaN	Web	09/13/2014	Closed with explanation	Yes	Yes	1027760

Und jezt sprechen wir über Datenverständnis, wir gücken die shape des Daten ist fast 1 Million records/rows and 18 Features/variables.

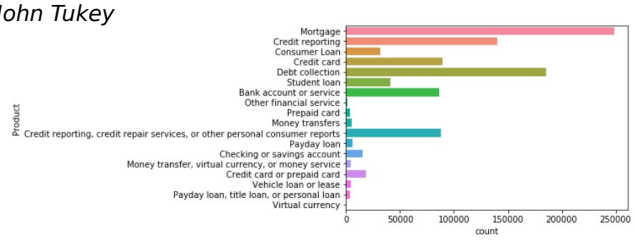
Es gibt, …., ..

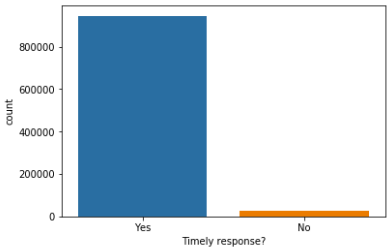
Und wählen wir ‘consumer complaints narrative’ feature aus als predictor variables/features und dann Product, timely response, und consumer disputeds werden ausgewählt als target variables.

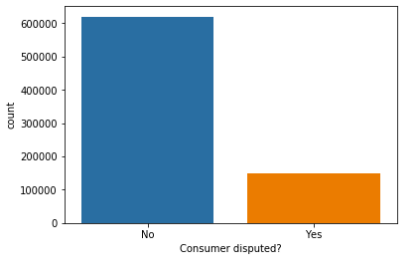
Nachdem Datensammlung, haben wir eine wichtige Sache zu tun, nämlich EDA.

Exploratory Data Analysis

„EDA can never be the whole story, but nothing else can serve as the foundation stone“
- John Tukey







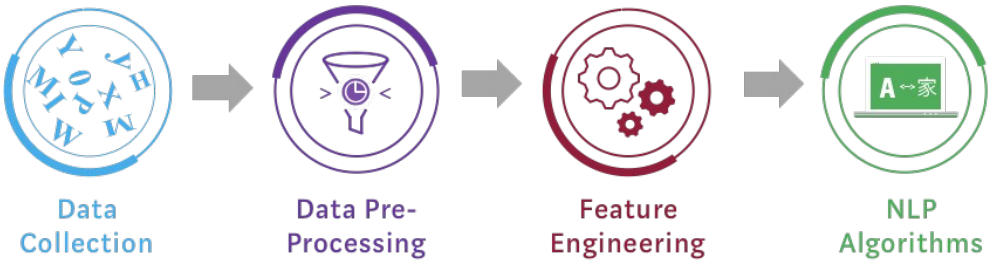
20.02.2019 https://alvinrindra.github.io 6

EDA …. John Tukey… Es bedeutet, EDA ist wichtig, um das Datenverständnis zu beginnen. Wir können target variables plotten mit graphical EDA. In diesem Fall benutze Ich seaborn packages.

Wir können sehen, dass es einige Majoritätsklassen und Minderheitenklassen gibt. Mortgage class ist fast … in gesamt.

Am Anfang habe ich das gesamte / Alle Label benutzt. Aber sie haben nicht gut funktioniert. Sie werden oft falsch klassifiziert. also habe ich mich entschlossen, entschieden, einige von labeln loszuwerden.

NLP Process Flow



*https://towardsdatascience.com/https-medium-com-vishalmorde-humanizing-customer-complaints-using-nlp-algorithms-64a820cef373

Da wir die NLP-Aufgabe lösen werden, verwenden wir den NLP-Prozess Flow.
Also, zwischen Data Preparation und Modelling in CRISP-DM. Es liegt NLP Process Flow.

Es könnte RNN, LSTM sein …

Alle Algorithmen, die sich auf die NLP-Task beziehen.

Data Preprocessing

GINKGO ANALYTICS

Better Data > Better Model

Data processing involves the following steps:

- Label Selection
- Remove NA values in every observation

- NLP and Regex Approaches:
    - Remove numeric and empty texts
    - Remove punctuation from texts
    - Convert words to lower case
    - Remove stop words
    - Remove unnecessary words
    - Stemming

    Other approaches: Tokenization, Lemmatization, POS tagging

Datenvorverarbeitung.

Bessere und sauberere Daten sind wichtiger als bessere Model. Dieser Satz habe ich oft in ML artikel gefunden und gesehen. Und Ich glaube das auch. Better Data for the win!

Datenvorverarbeitung beinhaltet den folgenden Schritt:

…. Entfernen

…..

Lemmatization verwandelt worter zum Grund/Basicform und es dauert. Auf der anderen Seite ist das Stemming sehr schnell.

Warum stemming, converts words to lower case, weil wir den Wortschatz/Vocab Size so klein wie möglich halten wollen.

Data Preprocessing

Word Cloud as Narrative Visualization:

Word Cloud ist wichtig, um einen Blick auf die Narrative zu gucken. welche worte sind die meistens, mehrheit.

Wir müssen sicherstellen, dass die Wörter unseren Erwartungen entsprechen. According to what we expect.

Ich kann euch ein beispiel geben. Bevor verwende Ich Word Cloud, Ich dachte, dass die Datenreinigung gut funktioniert. Aber wenn ich es benutze und die text visualisieren. The majority texts are XXXX und XXXX. Verstehsen sie?

Und dann Ich weiß was ist die problem.

Data Preprocessing

sel_df_cat['Consumer complaint narrative'][2]

"I purchased a new car on XXXX XXXX. The car dealer called Citizens Bank to get a 10 day payoff on my loan, good till XXXX XXXX. The dealer sent the check the next day. When I balanced my checkbook on XXXX XXXX. I noticed that Citizens bank had taken the automatic payment out of my checking account at XXXX XXXX XXXX Bank. I called Citizens and they stated that they did not close the loan until XXXX XXXX. (stating that they did not receive the check until XXXX. XXXX.). I told them that I did not believe that the check took that long to arrive. XXXX told me a check was issued to me for the amount overpaid, they deducted additional interest. Today (XXXX XXXX,) I called Citizens Bank again and talked to a supervisor named XXXX, because on XXXX XXXX. I received a letter that the loan had been paid in full (dated XXXX, XXXX) but no refund check was included. XXXX stated that they hold any over payment for 10 business days after the loan was satisfied and that my check would be mailed out on Wed. the XX/XX/XXXX.. I questioned her about the delay in posting the dealer payment and she first stated that sometimes it takes 3 or 4 business days to post, then she said they did not receive the check till XXXX XXXX I again told her that I did not believe this and asked where is my money. She then stated that they hold the over payment for 10 business days. I asked her why, and she simply said that is their policy. I asked her if I would receive interest on my money and she stated no. I believe that Citizens bank is deliberately delaying the posting of payment and the return of consumer 's money to make additional interest for the bank. If this is not illegal it should be, it does hurt the consumer and is not ethical. My amount of money lost is minimal but if they are doing this on thousands of car loans a month, then the additional interest earned for them could be staggering. I still have another car loan from Citizens Bank and I am afraid when I trade that car in another year I will run into the same problem again."

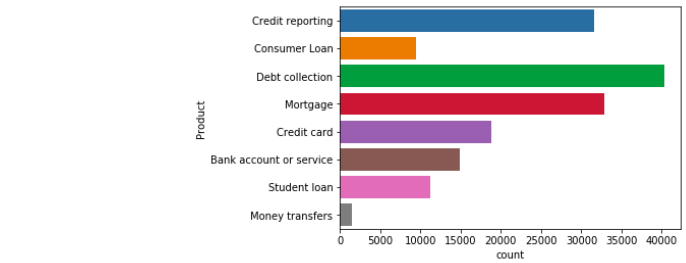
X[2]

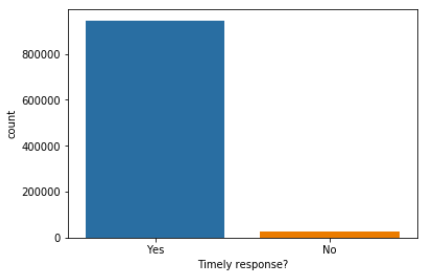
'purchas new car car dealer call citizen bank get day payoff loan good till dealer sent check next day balanc checkbook notic citizen bank taken automat payment check account bank call citizen state close loan state receiv check told believ check took long arriv told check issu amount overpaid deduct addit interest today call citizen bank talk supervisor name receiv letter loan paid full date refund check includ state hold payment busi day loan satisfi check would mail wed question delay post dealer payment first state sometim take busi day post said receiv check till told believ ask money state hold payment busi day ask whi simpli said polici ask would receiv interest money state no believ citizen bank deliber delay post payment return consum money make addit interest bank illeg be hurt consum ethic amount money lost minim thousand car loan month addit interest earn could stagger still anoth car loan citizen bank afraid trade car anoth year run problem again'

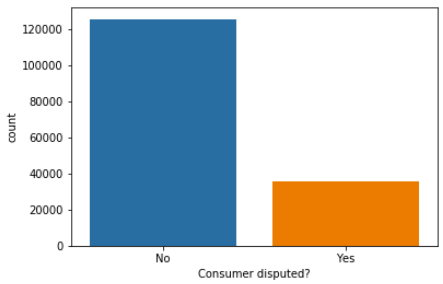
Das ist das Ergebnis der Datenbereinigung.

Es gibt keine ...

Data Preprocessing







20.02.2019 https://alvinrindra.github.io 11

wie wir schon besprochen haben, in Datenvorverarbeitung es gibt Schritte:
Label Selection, NA values entfernen.

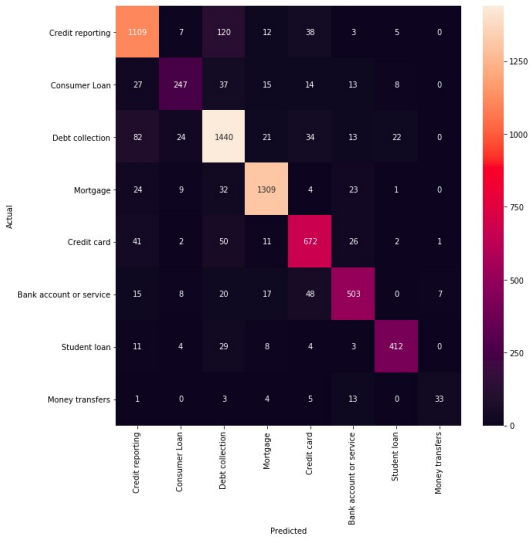
Also das ist die gewählte label, gewählte classes. Warum wähle ich diese labeln aus? Es gibt eine geschichte, eine Historie. Dafür gibt es einen Grund.

Ich habe mir die Confusion Matrix angesehen. Dass diese Classes/Katgories besser accuracy haben. Ich entferne manche ambiguos/vieldeutig kategorie wie …

It’s hard to classify, even we as a human. Und Ich entferne auch the minority classes, weil sie oft falsch klassifiziert werden. Many are misclassified.

Intermezzo:

Classification Matrix for choosing training labels



	precision	recall	f1-score	support
Credit reporting	0.85	0.86	0.85	1294
Consumer Loan	0.82	0.68	0.75	361
Debt collection	0.83	0.88	0.86	1636
Mortgage	0.94	0.93	0.94	1402
Credit card	0.82	0.83	0.83	805
Bank account or service	0.84	0.81	0.83	618
Student loan	0.92	0.87	0.89	471
Money transfers	0.80	0.56	0.66	59
micro avg	0.86	0.86	0.86	6646
macro avg	0.85	0.80	0.82	6646
weighted avg	0.86	0.86	0.86	6646

Ich habe eine kurze Intermezzo bevor wir über deep learning reden. Das is die confusion matrix mit classification report. Where I can find which classes are classified correctly, richtig klassifiziert, und welche Klasse ist falsch klassifiziert.

Wir können sehen, dass die majorität labeln wie ‘Credit reporting’, ‘Debt Collection’, und ‘Mortgage’ immer eine bessere accuracy haben und Meisten sind richtig klassifiziert.

Intermezzo:

Dealing with imbalanced classes

- TfidfVectorizer
- Finding unigrams and bigrams with chi2

```
# 'Bank account or service':
  . Most correlated unigrams:
. overdraft
. deposit
  . Most correlated bigrams:
. overdraft fee
. check account
# 'Consumer Loan':
  . Most correlated unigrams:
. vehicl
. car
  . Most correlated bigrams:
. car loan
. auto loan
# 'Credit card':
  . Most correlated unigrams:
. reward
. card
  . Most correlated bigrams:
. american express
. credit card
```

```
# 'Credit reporting':
  . Most correlated unigrams:
. experian
. equifax
  . Most correlated bigrams:
. tran union
. credit report
# 'Debt collection':
  . Most correlated unigrams:
. collect
. debt
  . Most correlated bigrams:
. collect debt
. collect agenc
# 'Money transfers':
  . Most correlated unigrams:
. moneygram
. western
  . Most correlated bigrams:
. money transfer
. western union
```

```
# 'Mortgage':
  . Most correlated unigrams:
. modif
. mortgag
  . Most correlated bigrams:
. mortgag payment
. loan modif
# 'Student loan':
  . Most correlated unigrams:
. student
. navient
  . Most correlated bigrams:
. privat loan
. student loan
```


Hier ist noch Intermezzo, Dealing with imbalanced classes.

Am Anfang habe ich TfidfVectorizer verwendet, also ich kann unigrams, bigrams, oder n-grams in jedes Narrative text finden. So, I want in each categories/class has their own unique and strong n-grams.

Es scheint gut zu funktionieren. Aber, leider funktioniert dieser Vectorizer nicht mit großen Datenmengen. Ich habe immer MemoryError bekommen. Es funktioniert nur mit dataset circa 50k rows.

Und verwende ich auch diese vectorizer für die word embeddings im deep learning model. Aber ich hatte eine schlechte performance. Ich weiß es nicht. und Ich muss später noch mal gucken.

Intermezzo:

Multinominal NB with limited data, nrows = 50000

```
selection = sel_df_cat.iloc[:, [0,1]]
debt_collection = selection[selection['Product'] == "Debt collection"].head(10)
print(debt_collection)
```

```
             Product                       Consumer complaint narrative
12   Debt collection  This company refuses to provide me verificatio...
16   Debt collection  This complaint is in regards to Square Two Fin...
49   Debt collection  I am writing to request your assistance in loo...
107  Debt collection  My identity was stolen. I filed a complaint, p...
190  Debt collection  The first communication that I received from t...
227  Debt collection  My complaint is n't about phone calls, but I b...
230  Debt collection  In a clearance interview a company was reporte...
235  Debt collection  i gave a vehicle away signed a bill of sale an...
249  Debt collection  Years ago when they first harassed me about th...
255  Debt collection  I have asked real time solutions several times...
```

```
print(clf.predict(count_vect.transform(debt_collection['Consumer complaint narrative'])))
```

```
['Debt collection' 'Debt collection' 'Debt collection' 'Debt collection'
 'Debt collection' 'Debt collection' 'Debt collection' 'Debt collection'
 'Debt collection' 'Debt collection']
```

```
mortgage = selection[selection['Product'] == "Mortgage"].head(10)
print(mortgage)
```

```
      Product                       Consumer complaint narrative
25   Mortgage  Started the refinance of home mortgage process...
26   Mortgage  In XXXX, I and my ex-husband applied for a ref...
29   Mortgage  Mortgage was transferred to Nationstar as of X...
69   Mortgage  Need to move into a XXXX facility. Can no long...
85   Mortgage  I had an FHA loan at US Bank that was paid off...
95   Mortgage  I went through a divorce several years ago and...
118  Mortgage  Select Portfolio Servicing has been deceptive ...
161  Mortgage  I got recent modification ( XXXX/XXXX/2015 ) f...
172  Mortgage  I was late on my mortgage payments and decided...
175  Mortgage  Requested a payoff quote by fax and certified ...
```

```
print(clf.predict(count_vect.transform(mortgage['Consumer complaint narrative'])))
```

```
['Mortgage' 'Mortgage' 'Debt collection' 'Debt collection' 'Mortgage'
 'Mortgage' 'Debt collection' 'Mortgage' 'Mortgage' 'Debt collection']
```


Also versuche ich, diesen Vektorisierer für den Vanilla Machine Learning-Algorithmus zu verwenden. In diesem Fall verwende ich Multinominal NB und es scheint gut zu funktionieren.

Es hat circa 86% accuracy und the majority class sind natürlich richtig klassifiziert.

Feature Engineering

GINKGO ANALYTICS

- Word Embeddings: Transforming text into a meaningful vector or array of numbers.

- N-grams : An unigram is a set of individual words within a document; bi-gram is a set of 2 adjacent words within a document.

- TF-IDF values: Term-Frequency-Inverse-Document-Frequency is a numerical statistic representing how important a word is to a document within a collection of documents.

```
# Feature Engineering to encode the text to sequences and to encode the label to categorical sequences
def sentences_to_sequences(X):
    token = Tokenizer(num_words=vocab_size, filters='!"#$%&()*+,-./:;<=>?@[\]^_`{|}~ ', lower=True, split=' ')
    token.fit_on_texts(X)
    X_seq = token.texts_to_sequences(X)
    X_seq = sequence.pad_sequences(X_seq, maxlen=max_doc_len)
    return X_seq

def label_encoding(y, num_cls):
    le = LabelEncoder()
    y_en = le.fit_transform(y)
    y_en = to_categorical(y_en, num_classes= num_cls)
    return y_en
```

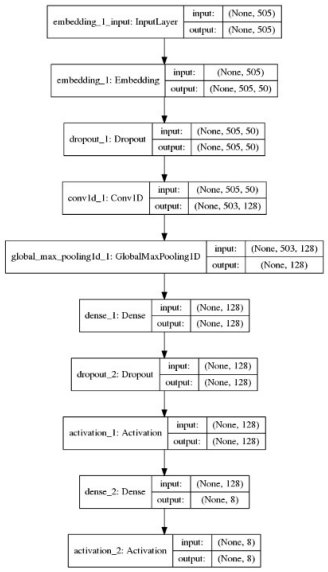

Jetzt sind wir im Feature Engineering. Es gibt einige Arten von Feature-Engineering.

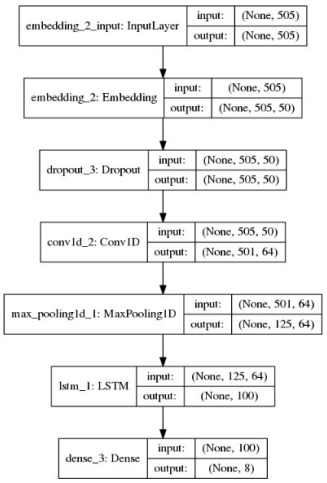
Es gibt ...

Wie wir schon gesehen haben, N-grams, TF-IDF.

Was funktioniert am besten für Deep Learning? Das ist Word Embeddings.

Modelling:
Deep Learning (CNN VS LSTM)





20.02.2019 https://alvinrindra.github.io 16

Nach dem Feature Engineering diskutieren wir nun über die Modellierung, NLP-Algorithmen. Welches ist das Beste für diesen Fall.

Ich habe zwei Modelle ausprobiert, CNN und LSTM.

Obwohl CNN für die Bildklassifizierung populär ist, funktioniert es auch für die Textklassifizierung. Und LSTM ist populär für many-to-many solutions wie sentiment anlyse, machine translation, chatbot, NER, und so weiter.

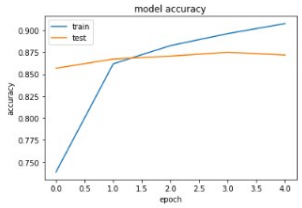
Aber diese LSTM Model ist auch populär für die Trainingszeit. We can train it like forever. Es braucht wirklich Zeit, deshalb verwenden wir noch Convolutiional Layer to make um training schneller zu machen.

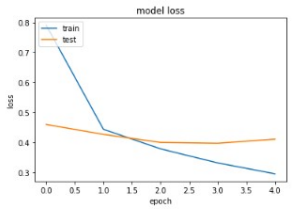
Training & Evaluation:
Deep Learning (CNN VS LSTM)

GINKGO ANALYTICS

CNN

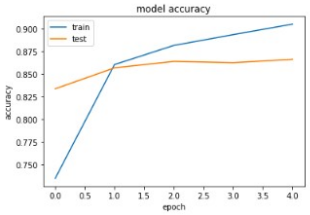
loss: 0.2956 - acc: 0.9074 - val_loss: 0.4111 - val_acc: 0.8716
Epoch 00005: val_acc did not improve from 0.87472
Training time: 89.4 s

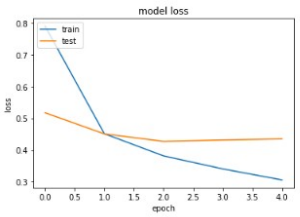




LSTM

loss: 0.3062 - acc: 0.9047 - val_loss: 0.4351 - val_acc: 0.8659
Epoch 00005: val_acc improved from 0.86365 to 0.86593
Training time: 995.3 s





Dies ist das Ergebnis von Training und Evaluation. Wir haben etwas die gleiche Performance zwischen CNN und LSTM.

CNN hat…. LSTM hat … mit 5 Epoch.

Aber was den Unterschied macht, ist die Trainingszeit. LSTM dauert 10x länger zum Trainieren.

Evaluation:
Deep Learning (CNN VS LSTM)

GINKGO ANALYTICS

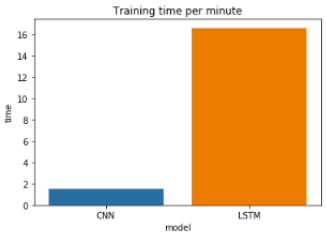
Observations:

- Based on the plots, both CNN and LSTM have similar accuracy and validation accuracy. Even CNN has a slightly higher accuracy.

- CNN model outperformed in terms of training time. I knew that training LSTM could be forever, that's why we still need convolutional layer to make it faster.

- However, LSTM has a positive and consistent trend in the validation accuracy. On the other hand, CNN has a negative trend / more overfitting in the next possible epochs.

- LSTM is popular for many-to-many solutions, but for this classification problem many-to-one solution. CNN, for the win!



Okay, das ist die Obervations.

- beide haben ein ähnliches Ergebnis, ähnliches accuracy. Obwohl CNN kurze besser performance hat.

- CNN model outperformed LSTM in training time. wie ich schon gesagt habe, training LSTM could be ...

- LSTM hat jedoch einen positiven und beständigen Trend. Auf der anderen Seite hat CNN einen negativen Trend und mehr overfitting trend für die nächste epoch.

- At the end, benutze Ich CNN.

Evaluation:
Deep Learning (CNN VS LSTM)

GINKGO ANALYTICS

Hyperparameters:

max_features: 40330
max_len: 505
epochs: 5
batch_size: 128
filters: 128
kernel_size: 3
hidden_dims: 128

Optimization Strategies:

Adam Optimization
Manual Hyperparameter Tuning

Regularization:

Dropout
EarlyStopping
ModelCheckpoint

Testing:

```
index = 60
x_test = np.array([sel_df_cat.iloc[index, 1]])
y_result = np.array([sel_df_cat.iloc[index, 0]])
X_test_indices = sentences_to_sequences(x_test)
le = LabelEncoder()
le.fit_transform(sel_df_cat['Product'])
print('Narrative: ' + x_test[0] + ', Expected Product: ' + y_result[0] + ', Prediction Product: '+  le.inverse_transform([np.argma
```

```
['Narrative: Winn Law group continues to pursue me on a debt that was charged off in XX/XX/2011.I have asked them to stop callin
g me & harassing me so now they have threatened to file a judgement against me and continue with collection efforts. I have let
them know that California state law prohibits them from collection activity after 4 years., Expected Product: Debt collection, P
rediction Product: Debt collection']
```

```
x_test = np.array(['I have a problem with my loan and I need the solution fast'])
X_test_indices = sentences_to_sequences(x_test)
le = LabelEncoder()
le.fit_transform(sel_df_cat['Product'])
print('Narrative: ' + x_test[0] + ', Prediction Product: '+  le.inverse_transform([np.argmax(dl_clf.predict(X_test_indices))]))
```

```
['Narrative: I have a problem with my loan and I need the solution fast, Prediction Product: Consumer Loan']
```


Das ist die Training Setup.

Es gibt ..

- Regularization für overfitting verhindern

Am anfang benutze Ich kleine max_features. Ich denke, dass es nur circa 2000 worter gibt in Narrative und zwar es gibt 40k worter. Ändern der Max_ features, Es verbessert die Performance.

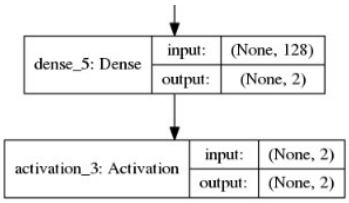
Viele Leute benutzen 64/128/256 als Hyperparameter nummer, vielleicht das ist die Best practice.

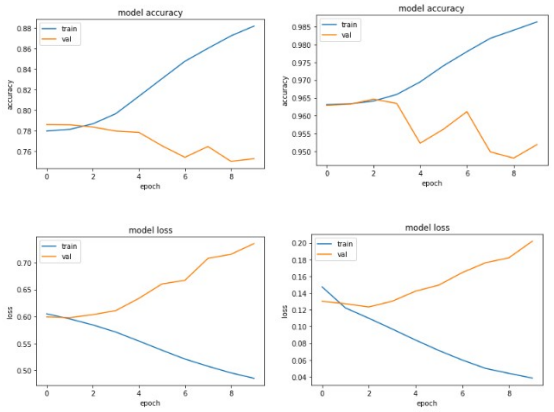
Und wir können die Testing sehen.

Disputed & Timely Response Classifier

GINKGO ANALYTICS

Multi-class Classification → Binary Classification
Softmax Layer → Sigmoid Layer
Categorical Crossentropy → Binary Crossentropy





Disputed | Timely Response

Und jetzt kommen wir zum Problem für ....

Aus dem narrativen Text können wir die Einsicht der Daten extrahieren. Zum Beispiel können wir die Sentiment-Analyse des Textes extrahieren, ob er positiv oder negativ ist. Damit können wir vorhersagen / forecasting, ob der Kunde streiten/dispute wird oder nicht.

Was habe ich getan, war, die Final layer und Loss function ändern. Von … zu …

Und das ist das Trainingsergebnis, accuracy und loss plotten.

Classifiers Testing

```
index = 60
x_test = np.array([sel_df_cat.iloc[index, 1]])
y_product = np.array([sel_df_cat.iloc[index, 0]])
y_time = np.array([sel_df_cat.iloc[index, 2]])
y_disputed = np.array([sel_df_cat.iloc[index, 3]])
X_test_indices = sentences_to_sequences(x_test)
le_disputed = LabelEncoder()
le_disputed.fit_transform(sel_df_concat['Consumer disputed?'])
le_product = LabelEncoder()
le_product.fit_transform(sel_df_concat['Product'])
le_time = LabelEncoder()
le_time.fit_transform(sel_df_concat['Timely response?'])

print('Narrative: ' + x_test[0])
print('Expected Product: ' + y_product[0] + ', Prediction Product: '+
      le_product.inverse_transform([np.argmax(dl_clf.predict(X_test_indices))]))
print('Expected Timely response: ' + y_time[0] + ', Prediction Timely response: '+
      le_time.inverse_transform([np.argmax(dl_clf_time.predict(X_test_indices))]))
print('Expected Disputed: ' + y_disputed[0] + ', Prediction Disputed: '+
      le_disputed.inverse_transform([np.argmax(dl_clf_disputed.predict(X_test_indices))]))
```

```
Narrative: Winn Law group continues to pursue me on a debt that was charged off in XX/XX/2011.I have asked them to stop calling me
& harassing me so now they have threatened to file a judgement against me and continue with collection efforts. I have let them kno
w that California state law prohibits them from collection activity after 4 years.
['Expected Product: Debt collection, Prediction Product: Debt collection']
['Expected Timely response: Yes, Prediction Timely response: Yes']
['Expected Disputed: No, Prediction Disputed: No']
```

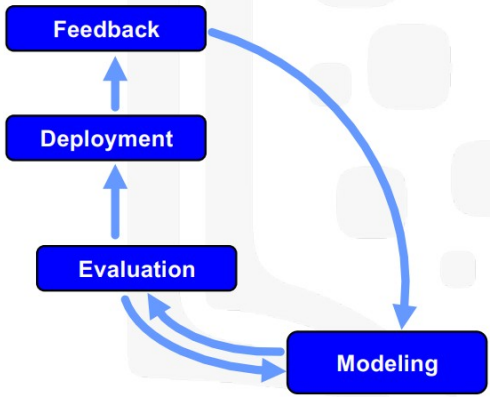
```
x_test = time_yes['Consumer complaint narrative']
X_test_indices = sentences_to_sequences(x_test)
le = LabelEncoder()
le.fit_transform(sel_df_cat['Timely response?'])
predicts = dl_clf_time.predict(X_test_indices)
for predict in predicts:
    print(le.inverse_transform([np.argmax(predict)]))
```

```
['Yes']
['Yes']
['Yes']
['No']
['Yes']
['Yes']
['Yes']
['Yes']
['Yes']
['Yes']
```


und jetzt können wir unseren Klassifikator testen.

Aus einer einzigen Narrative Text können wir Multiple targets vorhersagen. Predict multiple target.

Next:
Deployment? Feedback?



*https://www-01.ibm.com/events/wwe/grp/grp304.nsf/vLookupPDFs/Polong%20Lin%20Presentation/$file/Polong%20Lin%20Presentation.pdf

Nächster Schritt: Depolyment? Feedback?

Wie wir besprochen haben, Data Science/Machine Learning prozess ist ein iterativ prozess. Wenn es eine Feedback gibt, Wir können unseren Algorithmus umgestalten/remodellen.

Wir müssen erneut herausfinden, was das Problem ist und welche Faktoren die die Performance verbessern können.

Performance Improvements

- More Fine Tuning Hyperparameters: Grid Search, Random Search, ...
- Improve Text Preprocessing with NLP approaches: Lemmatization, NER for finding important entities, Topic Identification in the narratives, …
- Training other features
- Implement other feature engineering approaches (Tfidf, n-grams, factorization).
- Adding more data for minority classes and reducing data for majority classes.

Infrastructure Setup:

Kaggle provides Nvidia Tesla K80 GPU with limitation to 11GB GPU Memory.

Dies sind die Dinge, von denen Ich glaube, dass sie die Performance verbessern könnten.

Conclusion

- Vanilla ML such as SVM or NB could be an option for training not so large dataset.
- Deep Learning is very robust for the vast amount of data.
- From the consumer complaints narrative, we can build classifier for the Product, Timely response, consumer disputed, ...
- Other predictor variables such as ‘company public response’ could be useful for increasing the performance.
- Would like to try LSTM once having better computational resources.

Fazit,

Die schlussfolgerung

QuestIOns?

GINKGO ANALYTICS

Thank you!
Vielen Dank!
Terimakasih!

Gibt es eine Frage?