



How to make use of
unstructured data – Claims
classification via Natural
Language Processing

Data Science & Data Ethics
e-Conference by EAA

29/30 June 2020

Antoine LY
SCOR

Antoine Ly is Head of Data Science at SCOR Global Life.

Doctor in mathematics on the application of machine learning to insurance, he is also graduated from [ENSAE Paris](#) and MSc "Data Learning and Knowledge" of Sorbonne University. He teaches in the Data Science program given by the French Institute of Actuaries in collaboration with Pr. Elie and Pr. Charpentier. Antoine also teaches "Machine Learning with python" and "Distributed computing" at ENSAE Paris.

Antoine is a certified Actuary, member of the French Institute of Actuaries and contribute to the AAE taskforce in Artificial Intelligence and Data Science.

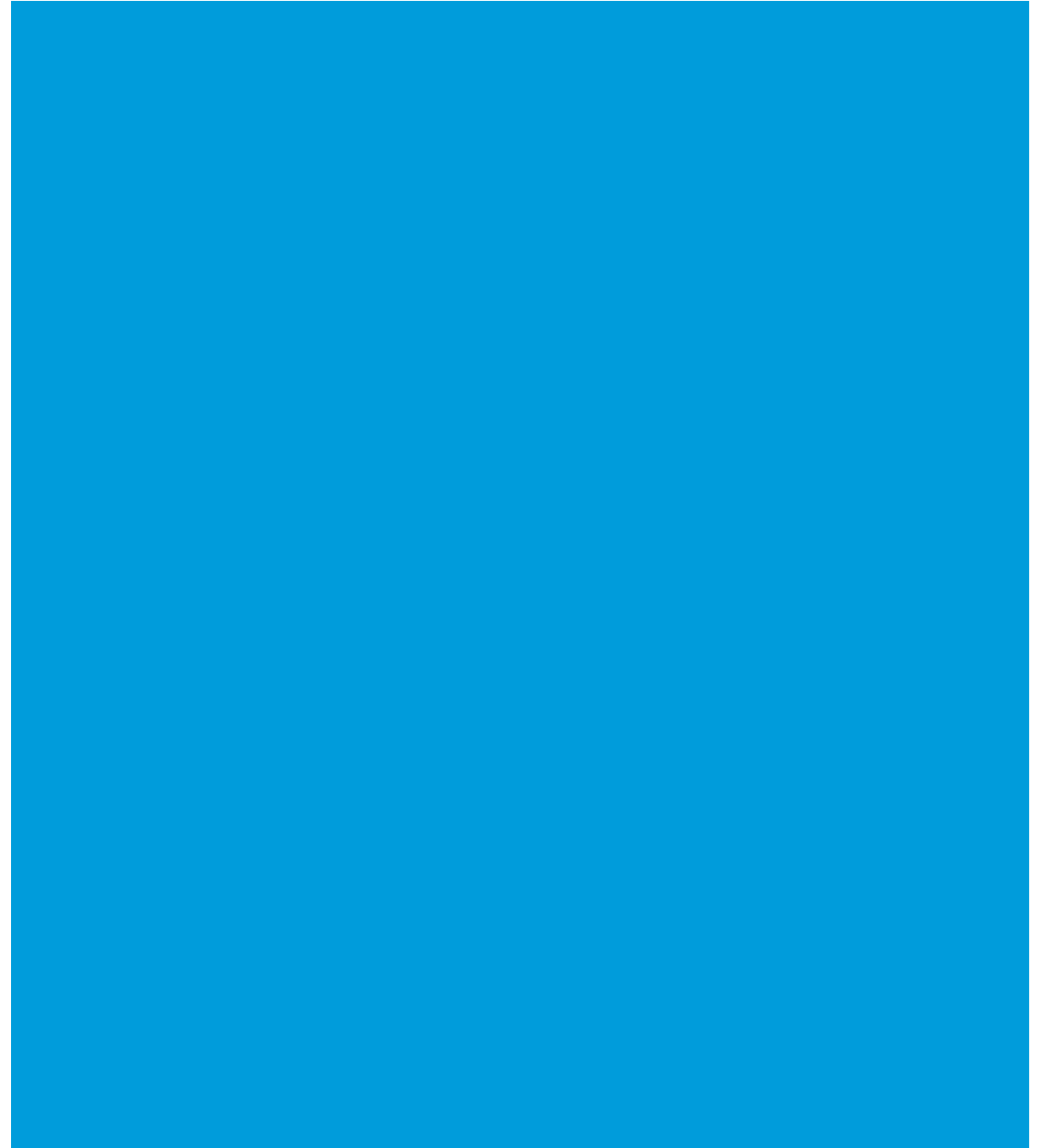
ABOUT ME



Antoine
LY

SCOR

INTRO- DUCTION



WHAT IS IT ABOUT?

Text mining is a field related to data analytics consisting in the analysis of textual data.

A textual data point can be a character, a word, a sentence or a paragraph.



“This is a sentence I can use to feed an algorithm”

Most of the time, text is analyzed by a human who can read the text and transform it into structured information. It takes a lot of time when it comes to analyze thousands or million of sentences...

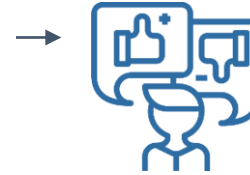


ID	Age of polycyholder	Description	Type of claim
1	27	« This is a description made by the claim manager »	1
2	67	« ... »	2

WHERE CAN WE FIND USEFUL TEXT INFORMATION?

Textual information are **almost everywhere**: reports, newspapers, comments, forms, websites, etc.

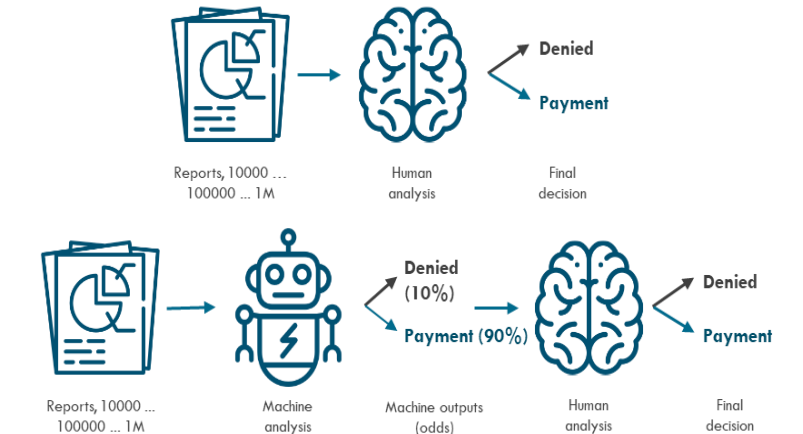
It contains a lot of information that our brain can easily interpret and exploit.



HOW TO USE TEXT INFORMATION?

In order to enhance the human capacity to deal with a lot of textual information, **machine learning/AI algorithms** can be used to automatize tasks.

To do so, **text representation** is an important field that will allow to fit proper algorithms dedicated to the task to be automatized.



- Text is considered as an unstructured data. To be analyzed by an algorithm text has to be converted into a numerical vector (sequence of numbers). Different methodologies exist in order to create an appropriate representation of text according to the problem we would like to solve.
- Different approaches exist:
 - **Grammatical:** try to detect the meaning and function of each word in the sentence (name, verb, attributes, etc.)
 - **Statistical:** try to focus only on occurrences and correlation with other variables (stemming needed for some languages)

The example of a statistical approach: Bag of Words (BoW)

“Bromwell High is nothing short of brilliant. Expertly scripted and perfectly delivered, this searing parody of a students and teachers at a South London Public School leaves you literally rolling with laughter...”



[Bromwell, High, nothing, short, brilliant, expertly, scripted, perfectly, delivered, this, searing, parody, students, teachers, South, London, Public, School, leaves, you, literally, rolling, with, laughter]



Bag of Words

Creation of D variables (**one for each word of the language vocabulary for example**). Each document is then represented by the counting of the present words in the document

hello good you morning left mining with thus brilliant
 $X = [0, 0, 1, 0, ..., 0, 0, 1, 0, 1]$

Vector of size D (potentially very huge!)

Simple but limited....

BAG OF WORDS / DATA PREPROCESSING

Pre-processing: Some words are not useful in term of predictive power because they are too frequent (so not very discriminant)
The goal of the pre-processing is to prepare text representation so it can feed a parametric models (e.g. linear regression)

Dictionary: The referential of words: list of words you consider as dimensions. (c.f. D variables of the previous slide)

In practice

Either a pre-defined list or the unique words observed in a corpus (set of documents)

Stopwords: List of words or character you want to exclude from your dictionary

In practice

Either a pre-defined list or custom list set manually

Stemming (tokenization): Consist in transforming words into invariant token (more or less easy according languages).

is, are → be

fly, flew, flown → fly

laptop, laptops → laptop

N-gram: considering in the dictionary word co-occurrence

“not bad”

“very neutral”

- **Libraries commonly used in python:**

- Preprocessing

- NLTK (very good for English)

- TreeTagger (stemming in French)

- Jieba (Chinese)

- Bag of Words

- Scikit-learn `sklearn.feature_extraction.text`

Variation of Bag of words: tf-idf instead of counting words, we weight them by the scalar: $tf(t) * idf(t)$

tf stands for the term frequency, the number of times a term occurs in one document

idf stands for inverse document frequency and is computed as for every term (word/character) t as

$$idf(t) = \frac{\log(1 + n)}{1 + df(t)} + 1$$

Where n is the number of documents in the corpus and $df(t)$ the number of documents containing the term t

Some variants of this weighting can exist. Some object also normalizes each resulting vector.

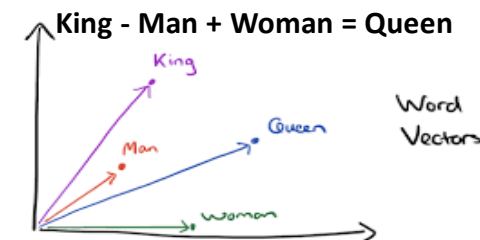
Deep learning in 2015 did some progress in the text representation to consider more the context. Instead of just counting the word, we pay attention to the co-occurrences and try to find a more concise dimension of the text representation. Each word is then represented by a numerical vector (which is not dummy).

We can use pre-trained models: text representation that is built on another dataset (e.g. Wikipedia)

The example of a contextual approach: Word embedding

“Bromwell High is nothing short of brilliant. Expertly scripted and perfectly delivered, this searing parody of a students and teachers at a South London Public School leaves you literally rolling with laughter...”

[Bromwell, High, nothing, short, brilliant, expertly, scripted, perfectly, delivered, this, searing, parody, students, teachers, South, London, Public, School, leaves, you, literally, rolling, with, laughter]



Word embedding

Creation of E variables (reduced dimension of the previous representation with D variables)
Each document is then represented a numerical vector which is supposed to translate the proximity between words

Bromwell High nothing short brilliant expertly ... with laughter

$$X = \begin{bmatrix} 0.34 & \dots & -1.34 \\ \vdots & \ddots & \vdots \\ 1.2 & \dots & 3.1 \end{bmatrix}$$

Column vector of size E (e.g. 300)

Embedding: numerical representation of text. The main goal is to find a dimension reduced representation that can take into account the meaning of the words.

How to build an embedding?

In your mind embedding can just be a matrix of size $D \times E$

Size $1 \times D$

$X = [0, 0, 1, 0, \dots, 0, 0, 1, 0, 1]$

Embedding matrix
Size $D \times E$

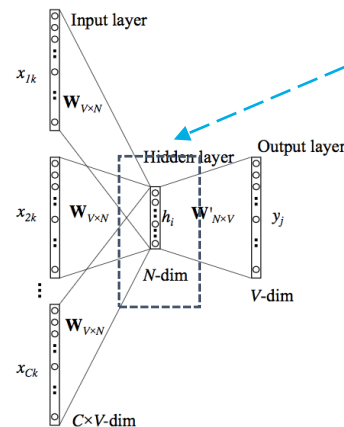
$X_{embedded} = [0, .34, \dots, 1.2]$

Size $1 \times E$

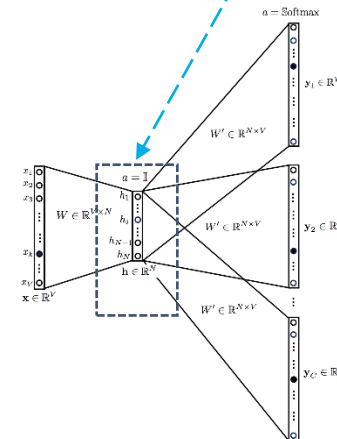
Word2Vec implementation

Historically 2 ways (2013 Mikalov et al.) to fit the embedding matrix from data. Both use neural networks

Common bag of Words: from context try to predict missing words



Skip-gram: from one word predict the context



Libraries commonly used in python:

Word2Vec

- Gensim
- Keras (tf or Theanos) + training by replicating Mikalov paper

Pre-trained embedding – matrix fitted on specific corpus

- Glove, Tencent, W2V

Example:

```
gensim.models.word2vec.Word2Vec(sentences=None, corpus_file=None, size=100, alpha=0.025, window=5, min_count=5,
max_vocab_size=None, sample=0.001, seed=1, workers=3, min_alpha=0.0001, sg=0, hs=0, negative=5, ns_exponent=0.75,
cbow_mean=1, hashfxn=<built-in function hash>, iter=5, null_word=0, trim_rule=None, sorted_vocab=1, batch_words=10000,
compute_loss=False, callbacks=(), max_final_vocab=None)
```

CBOW by default (1 =Skip gram)

TEXT MINING IN INSURANCE: USED METHODOLOGIES

In 2019, research did good progress in finding a word representation that can be generalize to most of the textmining issued. Mostly known through the BERT model, this representation can take into account not only the context but also the importance of the words inside the sentence (attention model). Those models are built with what is called **transformer**. It applies attention mechanisms to gather information about the relevant context of a given word, and then encode that context in a rich vector that smartly represents the word. The BERT model use a succession of transformers to build the best embedding of the words by considering the context.

The example of a contextual approach: BERT

“Bromwell High is nothing short of brilliant. Expertly scripted and perfectly delivered, this searing parody of a students and teachers at a South London Public School leaves you literally rolling with laughter...”

Contextual word embedding

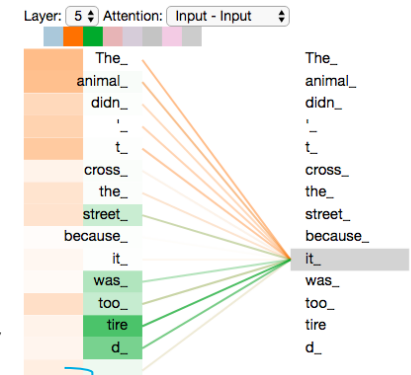
Creation of F variables (reduced dimension of the previous representation with D variables)
Each document is then represented a numerical vector which is supposed to translate the proximity between words

The vector representation of one word can change according the context and sentence.

[Bromwell, High, nothing, short, brilliant, expertly, scripted, perfectly, delivered, this, searing, parody, students, teachers, South, London, Public, School, leaves, you, literally, rolling, with, laughter]

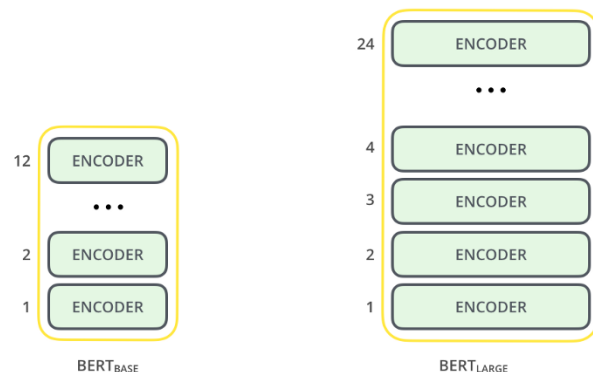
Bromwell nothing short brilliant expertly ... with laughter


$$X = \begin{bmatrix} 0.45 & \dots & -1.47 \\ \vdots & \ddots & \vdots \\ 3.56 & \dots & 0.1 \end{bmatrix}$$



Column vector of size F (e.g. 200)

BERT refers to the Google model. It stands for « Bidirectional Encoder Representations from **Transformers** » (published in 2019)




 We replace the embedding matrix by a
real deep learning model to produce
the embedding

“BERT’s clever language modeling task masks 15% of words in the input and asks the model to predict the missing word. To make BERT better at handling relationships between multiple sentences, the pre-training process also included an additional task: given two sentences (A and B), is B likely to be the sentence that follows A? Therefore we need to tell BERT what task we are solving by using the concept of attention mask and segment mask. In our case, all words in a query will be predicted and we do not have multiple sentences per query. ”

Implementation: Torch, Tensorflow (so available in keras and pytorch)

Really good high-level tutorial: <https://towardsdatascience.com/bert-for-dummies-step-by-step-tutorial-fb90890ffe03>

For deep dive: <https://arxiv.org/pdf/1810.04805.pdf>

TEXTMINING USAGE: ALGORITHM FITTING PIPELINE ILLUSTRATION

Dimension reduction

TF IDF

Vectorization

Embedding
Word2Vec,
BERT

Claim Cause	Words	乳腺癌	肺癌	宫颈癌	肝癌	...
01恶性肿瘤-生殖系统-女性生殖器官-乳腺癌		0.000	0.000	0.021	0.042	...
28重疾-颅脑手术-颅脑手术-颅脑手术		0.000	0.000	0.000	0.000	...
71重疾-冠心病-冠心病-冠心病		0.000	0.000	0.000	0.000	...
01恶性肿瘤-消化系统-胆、胆管癌		0.011	0.000	0.013	0.000	...
01恶性肿瘤-消化系统-胆、胆管癌		0.000	0.000	0.000	0.000	...
01恶性肿瘤-消化系统-胆、胆管癌		0.000	0.000	0.000	0.000	...
01恶性肿瘤-生殖系统-女性生殖器官-宫颈癌		0.000	0.000	0.049	0.000	...
01恶性肿瘤-生殖系统-女性生殖器官-乳腺癌		0.000	0.000	0.018	0.023	...
01恶性肿瘤-五官-鼻咽、鼻咽癌		0.000	0.000	0.049	0.065	...
01恶性肿瘤-五官-鼻咽、鼻咽癌		0.000	0.000	0.040	0.063	...
06重疾-终末期肾病-终末期肾病-终末期肾病		0.000	0.000	0.000	0.000	...
09重疾-急性脑卒中-急性脑卒中-急性脑卒中		0.000	0.000	0.000	0.000	...
01恶性肿瘤-五官-鼻、鼻窦癌		0.010	0.000	0.023	0.062	...
01恶性肿瘤-消化系统-胆、胆管癌		0.000	0.000	0.000	0.000	...
01恶性肿瘤-消化系统-胆、胆管癌		0.000	0.000	0.000	0.000	...

Machine Learning/AI

Logistic

SVM

GBM

XG Boost

Feed forward
Neural
Network

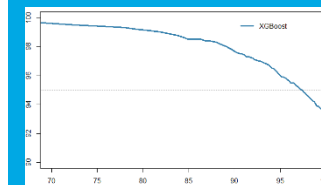
RNN

CNN

RCNN

Leaf-wise tree growth

Results



Prediction

Accident

Performance

Use cases 1: Claims adjudication

ZOOM ON USE CASES: CLAIMS CLASSIFICATION FOR CRITICAL ILLNESS*

❖ Original client data

- ~1M medical reports (digitalized) not analyzed
- 21964 medical reports manually classified to train the algorithm
- 175 cause of claims classes to be predicted – classification proposed by claim team

Model Output

10% of data is deliberately left to human classification

→ will be used for further learning going forward



2% of classified data is misclassified

→ Better than human practice**



Automating 90% of claims classification

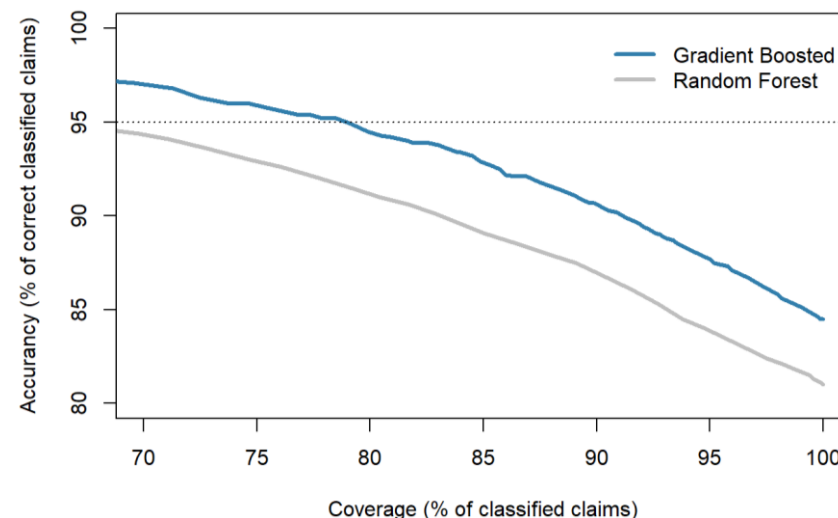
- Better performance as human
- absolute consistency
- leaves room for human expertise
- will naturally be further improved (sample size and/or success rate)

Medical report example

被保險人于2006年6月7日在家感觉不适，后到第五人民医院诊断为：心脏病。后又做冠状动脉支架植入手术。

The patient felt unwell at home on June 7 2006, and was then diagnosed with a heart disease at the Fifth People's Hospital of Later, coronary stent implantation was performed.

Accuracy Vs Coverage with XGboost



*SCOR used Text Mining on multiple line of business (MedEx, CI, PA)

**Human accuracy is close to 95% on these tasks

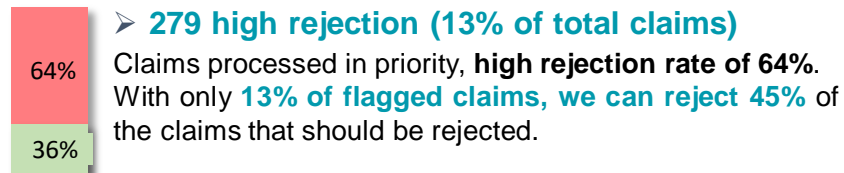
ZOOM ON USE CASES: CLAIMS ADJUDICATION FOR CRITICAL ILLNESS

❖ **Original client data**

- 10585 claims already processed by the claim department
- 2120 claims left for validation with **19% claims to decline** (rebalanced).
- Variables used for the adjudication: Medical report, Age, Sex, Sum insured, Type of claim, incurred date, report date, ...

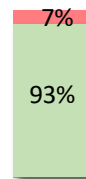
Model Output

Our final algorithm classified claims into 3 rejection categories (**High**, **Medium** and **Low**) to prioritize claims processing by the claim managers



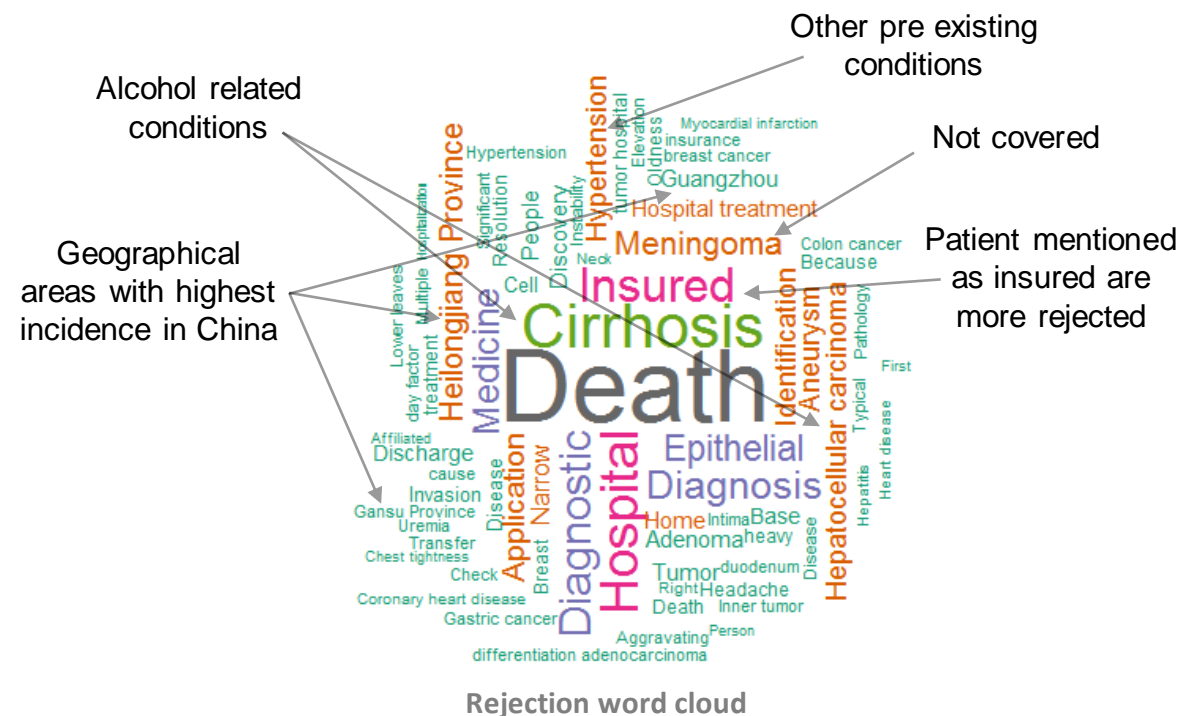
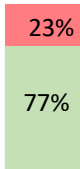
➤ **1218 low rejection (58% of total claims)**

79 rejected claims were wrongly accepted by the algorithm. After detailed review, the majority of these claims **could be rejected with the usage of the policy wording** regarding the conditions covered.



➤ **623 medium rejection (29% of total claims)**



Same situation as the accepted claims, the usage of the policy wording could improve the unknown with a greater % of claims being directly rejected



Use cases 2: Claims classification

ALL OUR SOLUTIONS ARE CLOUD BASED AND CAN DEPLOYED TO BETTER SUPPORT OUR CLIENTS

With a solid cloud strategy, SCOR Data Analytics can provide secure full stack AI services for insurance.



Sign in

Email Address *

Password *

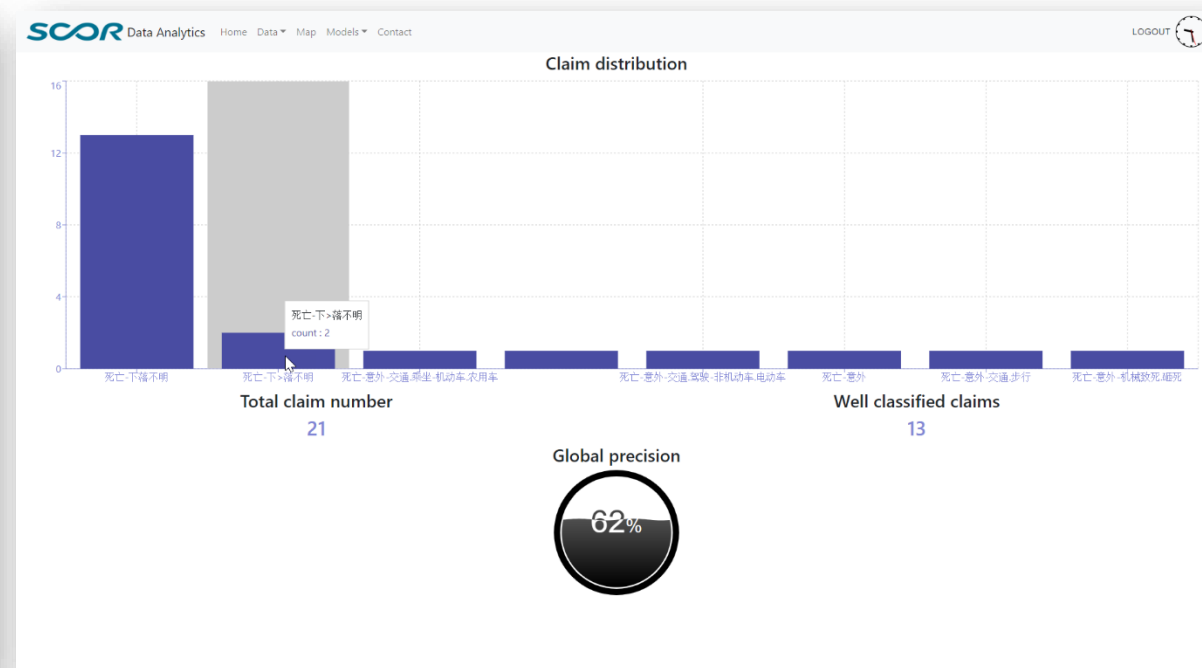
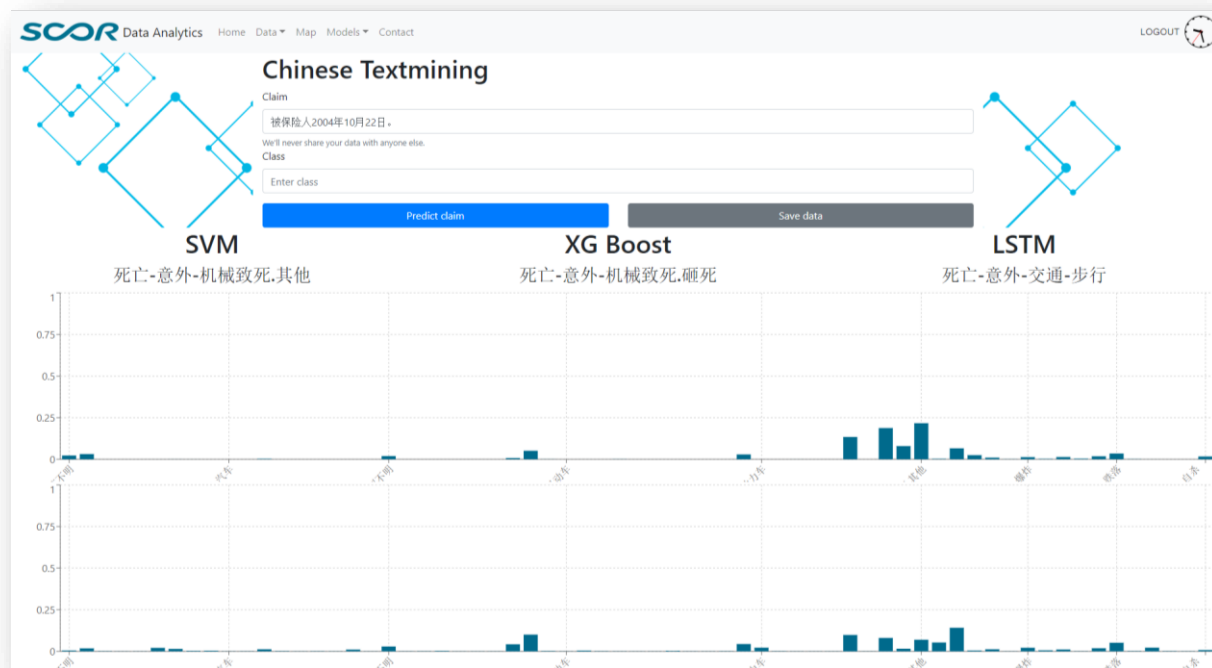
SIGN IN

[Forgot password?](#)

Copyright © Scor 2019

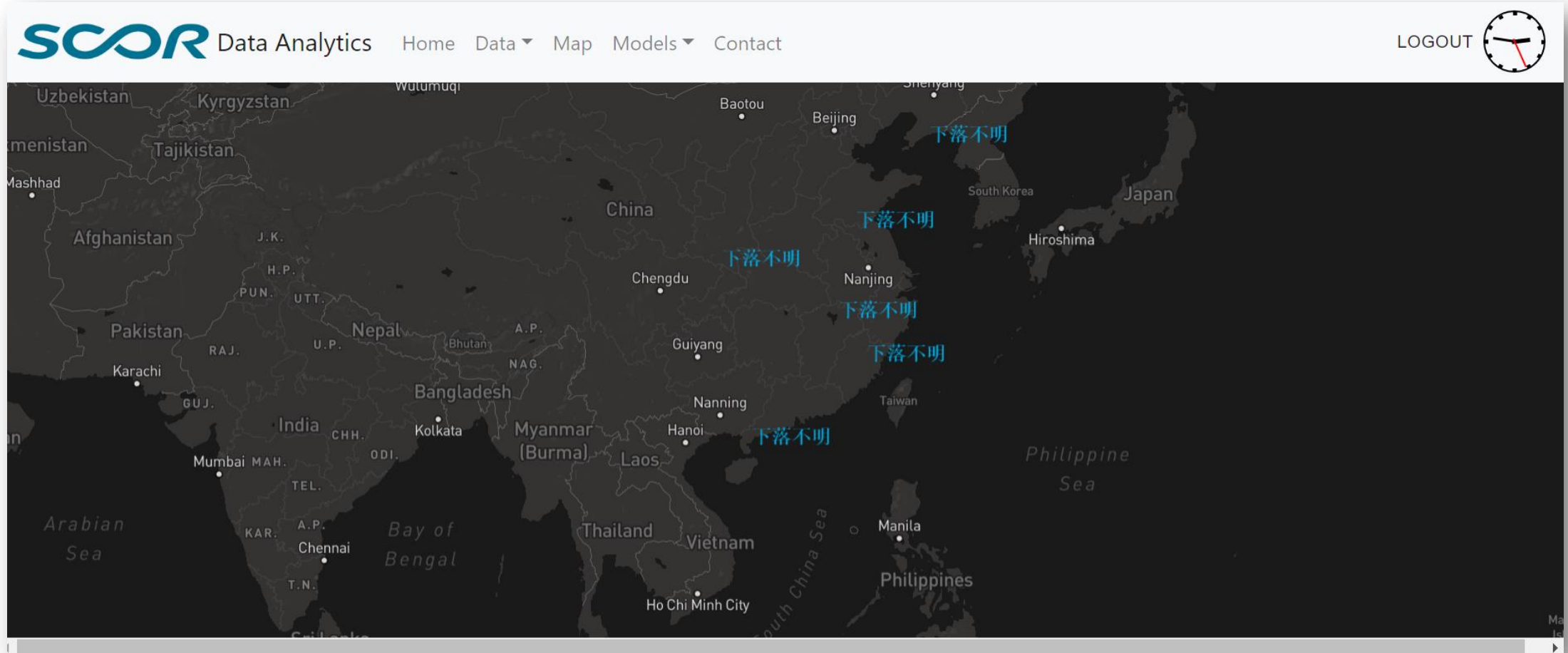
ALL OUR SOLUTIONS ARE CLOUD BASED AND CAN DEPLOYED TO BETTER SUPPORT OUR CLIENTS

Claim managers assisted with Artificial Intelligence work better.
 Solutions at SCOR are developed and tailored to suit our clients needs.



ALL OUR SOLUTIONS ARE CLOUD BASED AND CAN DEPLOYED TO BETTER SUPPORT OUR CLIENTS

Web scrapping enables SCOR to go beyond traditional data visualization.





Data Science & Data Ethics
e-Conference by EAA

29/30 June 2020

Contact

Antoine Ly

Head of Data Science – SCOR Global Life

aly@scor.com