Math 569 Project

AI IN DATA PRIVACY HACKATHON

Jayaprakash Kutala: jkutala@hawk.iit.edu

April 30, 2019

Contents

Abstract

In this hackathon, corpus of privacy documents is shared with the contestants to rate these documents, using multi-classifier. First, exploratory analysis is performed to identify features, trends, correlations and derive other interesting insights from the data. Later, test privacy notices are rated based on factors such as number of words per sentence, clarity of sentences and availability of relevant privacy content.

Machine learning models are built to classify the documents, and the performance of the models are measured. Model selection is performed based on the F1-Score to identify the best performing model and validate the accuracy of the predictions on new data. We conclude that tree-based XGBoost model classified the documents most accurately. We also observe some important predictor variables, including "is minor", "geo-location", "email", "share" and "sell" have significantly higher weights compared to other predictors.

Introduction

The EU General Data Protection Regulation (GDPR) is the most important change in data privacy regulation in 20 years. As the result, all the privacy notices on the web need to be compliant with these changes. In this competition, data set of approximately 500 General Data Protection Regulation notices from various websites are given to the contestants. Using this data, the teams need to build supervised natural language learning models to identify weaker and stronger GDPR notices.

In this hackathon, data, which describes the various features of a compliance privacy document, was leveraged to make predictions on other notices. External factors which influence the classification of policy documents are observed.

The data is first subjected to cleaning where missing values and erroneous data is imputed or dropped. Then variable transformations and variable selection was performed. Exploratory analysis was then performed to gain insights from the data and to determine whether feature engineering needed to be performed. This data is then split into two subsets for training the machine learning model and to test the performance of the model.

Finally, various machine learning techniques are applied, such as KNN methods, tree-based methods to accurately predict the model. The performance of the model was measured using the F1 scores of the testing data.

As in any machine learning problem, there is a trade-off between complexity and prediction accuracy, and that was also encountered in this contest. The models that are easy to interpret in our case tends not to give us good predictions. Currently our models use many predictor variables to obtain high prediction accuracy. However, this could present a challenge if we want to provide a standard model that can help classify the notices.

This project consists of many sections, including data preparation, exploratory data analysis, modeling, and conclusion, which explain in detail the steps performed at each stage of the project.

Feature Description

The dataset is comprised of many privacy notices that were acquired from multiple sources. Below categorical functional features are extracted using regular expression. For more details, refer the code.

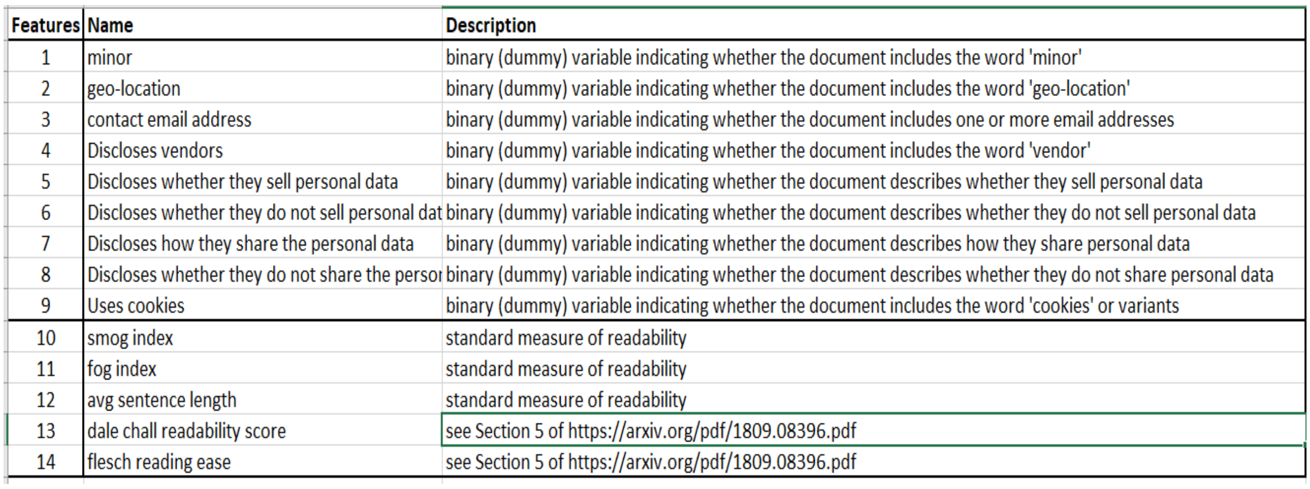
In order to achieve our goal, below features are extracted from privacy text file.

- **Readability Metrics**: The privacy document is broken down to sentences and words to understand readable readiness of the document. Below features are included as predictors to judge whether given privacy notice is readable ready.
  - *flesch_reading_ease* - Implements Flesch Formula: (below is an example)
    Reading Ease score = 206.835 - (1.015 × ASL) - (84.6 × ASW)
    Here, ASL = average sentence length (number of words divided by number of sentences) and ASW = average word length in syllables (syllables divided by number of words)
  - *smog_index* - Implements SMOG Formula / Grading
    SMOG grading = 3 + polysyllable count.
    Here, polysyllable count = number of words of more than two syllables in a sample of 30 sentences.
  - *dale_chall_readability_score* – Implements Dale Challe Formula:
    Raw score = 0.1579*(PDW) + 0.0496*(ASL) + 3.6365
    Here, PDW = Percentage of difficult words. ASL = Average sentence length.
  - *gunning_fog* - It is a weighted average of the number of words per sentence, and the number of long words per word. An interpretation is that the text can be understood by someone who left full-time education at a later age than the index.

- **Functional Features:** Further, a privacy document is considered better if it describes more about below factors.
  - *Is_minors* – did privacy notice mention whether site can be used my minors?
  - *Is_geo-location* - did it mention about geo-location of the user?
  - *Is_email* – did it mention any contact email id?
  - *Is_Share / Is_Not_Share* – did the policy mention about share data or not?

- *Is_Sell / Is_Not_Sell* – whether the user data is sold or not?
- *Is_cookies* – whether cookies are used on the site?

From our analysis on GDPR documents, we concluded that all above features are critical in deciding whether a given policy is compatible with GDPR rules or not. Most of the functional features such as is_minor, is_sell are categorical variables. Rest of the readability features are values. Moreover, all the above features can be represented in the below single table [Fig-1].



Features	Name	Description
1	minor	binary (dummy) variable indicating whether the document includes the word 'minor'
2	geo-location	binary (dummy) variable indicating whether the document includes the word 'geo-location'
3	contact email address	binary (dummy) variable indicating whether the document includes one or more email addresses
4	Discloses vendors	binary (dummy) variable indicating whether the document includes the word 'vendor'
5	Discloses whether they sell personal data	binary (dummy) variable indicating whether the document describes whether they sell personal data
6	Discloses whether they do not sell personal dat	binary (dummy) variable indicating whether the document describes whether they do not sell personal data
7	Discloses how they share the personal data	binary (dummy) variable indicating whether the document describes how they share personal data
8	Discloses whether they do not share the perso	binary (dummy) variable indicating whether the document describes whether they do not share personal data
9	Uses cookies	binary (dummy) variable indicating whether the document includes the word 'cookies' or variants
10	smog index	standard measure of readability
11	fog index	standard measure of readability
12	avg sentence length	standard measure of readability
13	dale chall readability score	see Section 5 of https://arxiv.org/pdf/1809.08396.pdf
14	flesch reading ease	see Section 5 of https://arxiv.org/pdf/1809.08396.pdf

Figure 1: Features extracted from privacy documents

Feature Selection and Standardization

Feature importance function ranks the value of each feature as the average importance across all the decision trees in the sequence. This importance function can be used for feature selection. From below importance plot [Fig-2], we can only run algorithms on important features and ignore all other features because removed features don't affect output of the algorithm. Finally, in this project, algorithms were run twice. First, by including all features and second, by excluding features that were relevant and were less correlated.

Since the range of values of raw data varies widely, in some machine learning algorithms, objective functions will not work properly without normalization. For example, most classifiers calculate the distance between two points by the Euclidean distance. If one of the features has a broad range of values, the distance will be governed by this feature. Therefore, the range of all features are normalized so that each feature contributes approximately proportionately to the final distance. Another reason why feature scaling is applied is that gradient descent converges much faster with feature scaling than without it. Hence, all the features are normalized in this project post extraction.

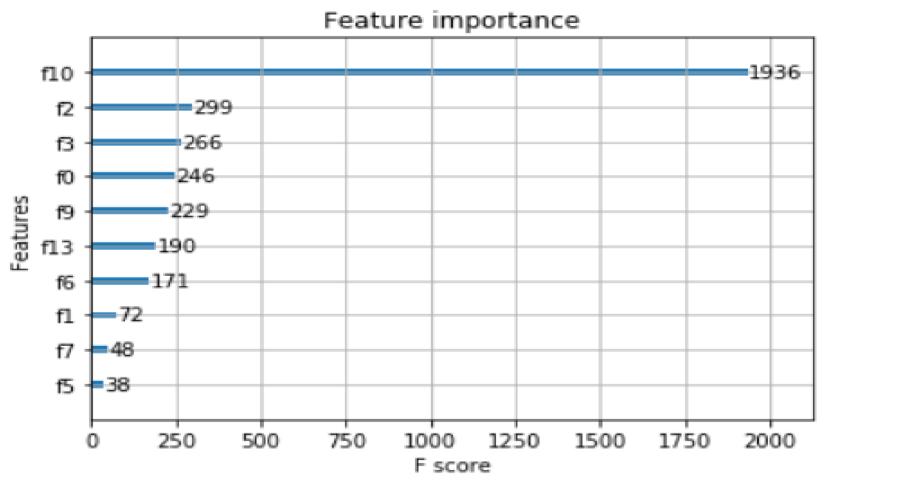


Figure 2: Feature Importance plot

Exploratory Data Analysis

Score has a high positive correlation with *is_email*(0.73). In addition, some of the explanatory variables were observed collinear. For example, *avg_sentence_length* and *dale_chall_readability_score* have a strong correlation of 1; *gunning_fog* and *flesch_reading_ease* have a negative correlation of -1 and high correlations with other variables. Analysis for multicollinearity should be performed [Fig-2].

From [Fig-2], it can also be concluded that the presence of functional variable such as *is_minor, is_email, is_vendor, is_cookies* will result in better compliance document compared to the privacy policies that don't have these variables. However, we can also observe that readability parameters don't have any visible correlation with scores. But, these features play significant role in predicting the scores because these features are considered critical as per the feature section algorithm in the previous section.

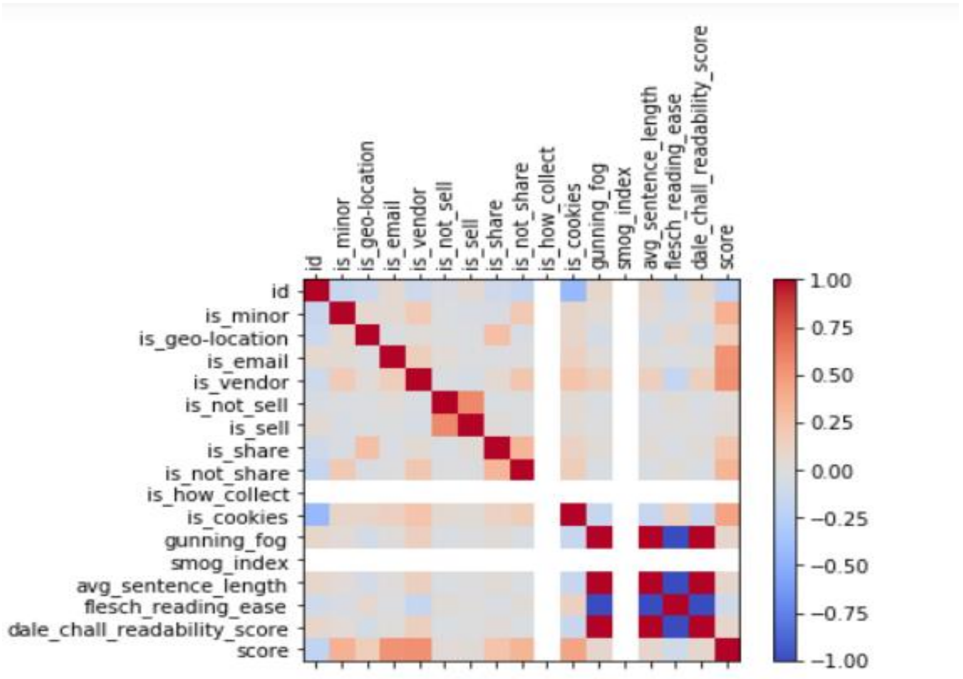


Figure 2: Correlation plot of numeric variables

Modeling

1. KNN Methods

1.1 Methodology

The K- nearest neighbor (KNN) model is intuitive and simple. As it is a non-parametric model, it has no assumptions and very easy to implement for multi-classifiers. However, found below issues with these features.

- Curse of Dimensionality: KNN works well with small number of input variables.
- KNN algorithm doesn't work well with categorical features since it is difficult to find the distance between dimensions with categorical features.
- K-NN needs homogeneous features: it is completely necessary that features have the same scale.
- Imbalanced data causes problems: If we consider two classes, A and B, and most of the training data is labeled as A, then the model will ultimately give a lot of preference to A. This might result in getting the less common class B wrongly

1.2 Results

For evaluating an algorithm, confusion matrix, precision, recall and f1 score are the most commonly used metrics. The confusion_matrix and classification_report methods of the sklearn.metrics can be used to calculate these metrics. We can that F1 scores and precision for this model are low.

Iteration	precision	recall	f1-score	support
1	0.73	0.95	0.83	20
2	0.7	0.75	0.72	63
3	0.46	0.45	0.46	40
4	0.73	0.19	0.64	16
5	1.00	0.6	0.75	5
Avg/total	0.68	0.6	0.67	144

Table 1: Results of KNN Models

Comparing Error Rate with the K Value:

From the output we can see that the mean error is less when the value of the K is between 1 and 3. So, we played around with the value of K to see how it impacts the accuracy of the predictions. It predicted above best results when K is at 3.

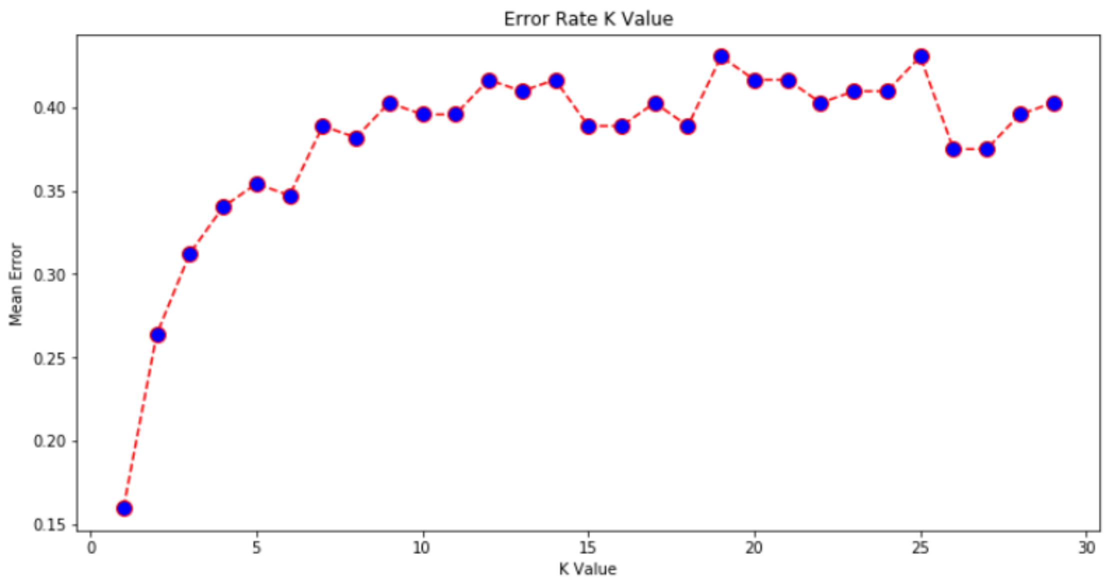


Figure 3: Error rate VS K-value

2.Tree-based Methods

Methodology

Tree-based methods have been favorite techniques in many industries with proven successful cases for prediction. These methods are considered non-parametric, making no assumption on the distribution of data and the structure of the true model. They require less data cleaning and are not influenced by outliers and multicollinearity to some fair extend. The simplest method is decision trees, which can be used for both regression and classification problems and provide a useful and simple tool for interpretation. However, a simple model like Decision Tree tends to not have good predicting power. More complicated methods, such as random forests and boosting, usually yield better results, though there is a trade-off between interpretability and prediction accuracy. As the purpose of this project is accurately predicting scores, random forest and XGBoost algorithms were chosen.

Random forests involve building many decision trees on a bootstrapped training set. When building decision trees, a random selection of m predictors among the full set of p predictors is chosen in each considered split. The split can only use one of those m selected predictors. This method of bagging is great to overcome the problem of choosing only a few strong candidates in the splits. It decorrelates the trees, making the results of averaging trees more robust.

XGBoost stands for eXtreme Gradient Boosting, implementing the gradient boosting techniques but being optimized for speed and performance. The boosting idea involves growing trees sequentially, meaning that each tree is built based on the information from previously grown trees.[5] XGBoost, however, applies more penal- ties in the boosting equation when updating trees and residuals compared to the traditional boosting method, while leveraging the structure of hardware to speed up computing time.[5] Therefore, it has been considered the best model among all tree-based models.

Results

After running different models with tuning parameters, a model for random forest and 2 models for XGBoost are chosen to compare the predicting power with the test data set. The random forest model built 500 trees and 8 predictors were considered at each split. This model was fitted using all described variables in our training set (15 variables in total). All the XGBoost models were trained with cross validation of 5 folds, using the classification report to evaluate the model. The first XGBoost model considered all available predictors, while the second model excluded some variables that were highly correlated to each other. The results of these models are displayed in the table 2 below. The XGBoost Log loss and classification error helped us to analyze early stopping technique to stop model training before the model overfits the training data, to monitor the performance of XGBoost models during training and to plot learning curves and to configure early stopping when training XGBoost models.

Model	F1-Scores	Model description
Random Forest	0.71425	Fit using all variables
XGBoost 1	0.75943	Fit using all variables
XGBoost 2	0.80562	Drop variables based on feature selection and correlation

Table 2: Results of Tree-based Models

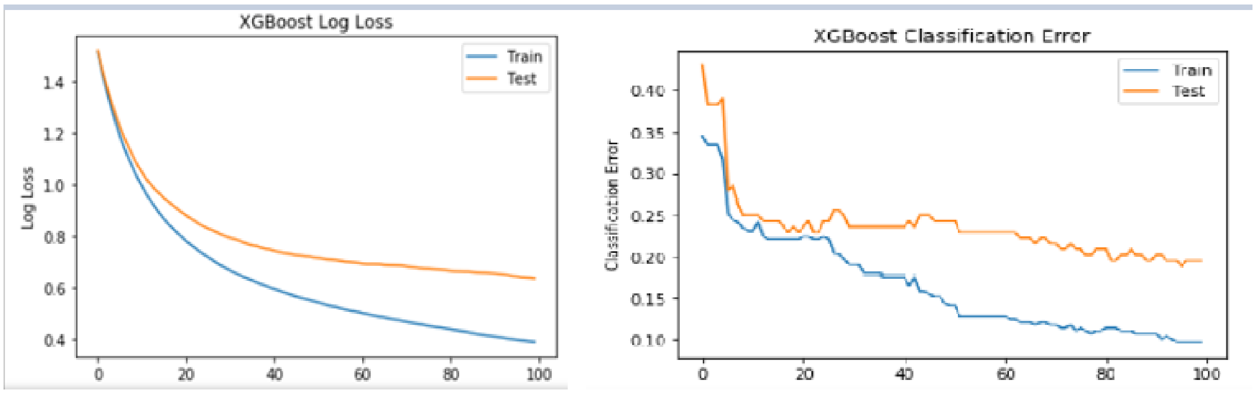


Figure 4: XGBoost LOG Loss Learning curve & Classification Error

Conclusions

The KNN models achieve good performance in explaining our target variable. However, due to curse of dimensionality and presence of categorical variable has resulted in slightly lower performance as compared to other boosting models. The model that could accurately predict the scores of the policies is XGBoost using all selected predictor variables. The key insights from the analysis is that the factors that most influence the scores include presence of words geo-location, share/not share data and sell/not sell data. As for future improvement, principal component analysis (PCA) can be performed on the collinear terms, among the predictor variables, this can be then used in the KNN to boost its performance. Additional future work could involve transforming the entire process of the project from selecting more appropriate words to building models and making predictions in production application where a user can input the document and the output being an approximate score of the privacy documents.

Appendix

Outcome of the XGBoost model:

```
model.fit(X_train, y_train, eval_metric=["merror", "mlogloss"],
eval_set=eval_set, verbose=True)
# make predictions for test data
y_pred = model.predict(X_test)
predictions = [round(value) for value in y_pred]
# evaluate predictions
accuracy = accuracy_score(y_test, predictions)
print ("Accuracy: %.2f%%" % (accuracy * 100.0))
Accuracy: 80.56%
```

Reference

[1] The EU General Data Protection Regulation (GDPR) compliance. Retrieved from https://en.wikipedia.org/wiki/General_Data_Protection_Regulation

[2] Introduction to Natural Language Tool Kit (NLTK). Retrieved from https://www.nltk.org/book/ch01.html

[3] Hackers guide to python spaCy Library. Retrieved from https://nlpforhackers.io/complete-guide-to-spacy/

[4] The Gunning's Fog Index (or FOG) Readability Formula. Retrieved from http://www.readabilityformulas.com/gunning-fog-readability-formula.php

[5] Introduction to Boosted Trees — xgboost 0.81 documentation. (n.d.). Retrieved from https://xgboost.readthedocs.io/en/latest/tutorials/model.html

[6] Avoid Overfitting by Early Stopping With XGBoost In Python. Retrieved from https://machinelearningmastery.com/avoid-overfitting-by-early-stopping-with-xgboost-in-python/

[7] The EU General Data Protection Regulation (GDPR) is the most important change in data privacy regulation in 20 years. Retrieved from https://eugdpr.org/

[8] James,G., Witten,D.,Hastie,T.,Tibshirani, R. (2017). An introduction to statistical learning with applications in R. New York: Springer.

[9] American Mathematical Society. (n.d.). Retrieved from http://www.ams.org/publicoutreach/feature-column/fcarc-taxi

[10] Introduction to Boosted Trees Python — xgboost packages documentation.Retrieved from https://xgboost.readthedocs.io/en/latest/python/python_intro.html