Explaining ‘Explainable AI’

Sahibpreet Singh

Why Explainable AI

1. Cross Validation can Fail

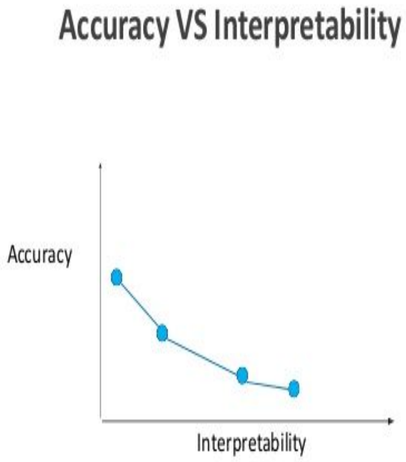
Although we can rely on cross validation for testing our model before putting in production
But the problem with cross validation is we never know what kind of testing data we'll get in production.

2. A/B Testing not the solution For everyone

Although A/B testing is a golden standard but the problem is
2.1 Companies will have to expose Not the best quality product to public testers.

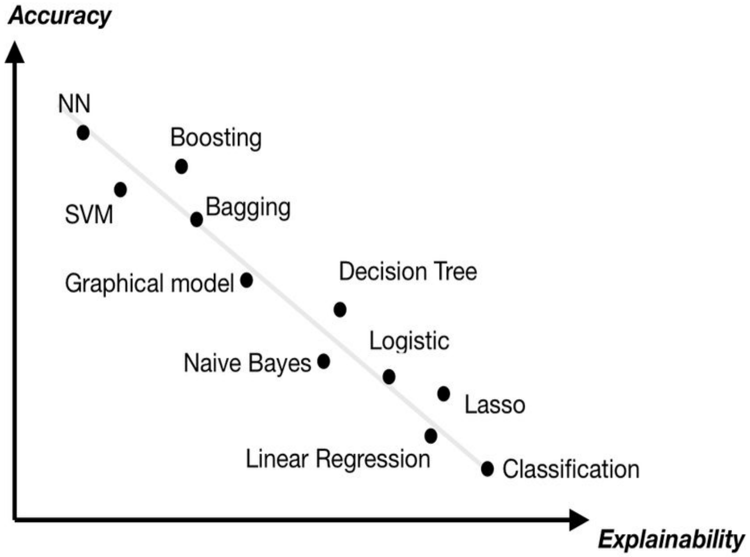
2.2 It can be expensive

WHY I AM STRESSING UPON IT



Source: https://www.youtube.com/watch?v=LAm4QmVaf0E&t=3658s

AND



INFLUENTIAL RESEARCH PAPERS

1. LIME =Local Model Agnostic Explainability -----> https://arxiv.org/abs/1602.04938
2. SHAPLEY = Shapley Additive Explainations ---->https://arxiv.org/abs/1911.11888

FRAMEWORKS THAT CAN BE USED

1. Lime = https://lime-ml.readthedocs.io/en/latest/
2. Eli5 = https://eli5.readthedocs.io/en/latest/overview.html
3. AIX 360= https://github.com/IBM/AIX360/
   This Framework AIX is an open source package by IBM
4. Deeplift= https://github.com/kundajelab/deeplift

What I am using ELI5

ELI5 is a python framework that supports interpretability for models like

1. Sklearn-crfsuite
2. Keras
3. Scikit-learn
4. LightGBM

VERY IMPORTANT RESOURCE **https://github.com/kundajelab/deeplift**

CREATE INTERPRETABLE TEXT CLASSIFICATION SYSTEM

For the purpose of creating such system we have used algorithms like

1. Tf/Idf + Logistic regression
2. Tf/Idf + SVC
3. Hashingvectorizer + Logistic regression

And For Interpretability we are using
LIME
(Local Interpretable Model Agnostic Explanations)

NOW WHAT IS LIME
‘Let’s break it’

1. **LOCAL in LIME means creating Interpretability only for one prediction**

   Why did the model make a certain prediction for an instance?

   We will zoom in on a single instance and examine what the model predicts for this input, and explain why. If you look at an individual prediction, the behavior of the otherwise complex model might behave more pleasantly. Locally, the prediction might only depend linearly or monotonically on some features, rather than having a complex dependence on them. For example, the value of a house may depend nonlinearly on its size. But if you are looking at only one particular 100 square meters house, there is a possibility that for that data subset, your model prediction depends linearly on the size. You can find this out by simulating how the predicted price changes when you increase or decrease the size by 10 square meters. Local explanations can therefore be more accurate than global explanations

2. Model Agnostic

It means any model till date can be applied to LIME and Lime can create local explanations for them
By treating each model as **Black box model**
should not make any assumptions about model while providing explanations.

Explanation is just an explanation 😊😊

TYPES OF INTERPRETABILITY

There are two main ways to look at a classification or a regression model:

1. Inspect model parameters and try to figure out how the model works **globally**;
2. Inspect an **individual prediction** of a model, try to figure out why the model makes the decision it makes.

AND LIME WORKS on SECOND POINT

HOW LIME WORKS 😉

1. Permute Data (can be based on different distance metrics like **euclidean distance,manhattan distance** etc.)

2. Calculate distance bw permutation and original distance.

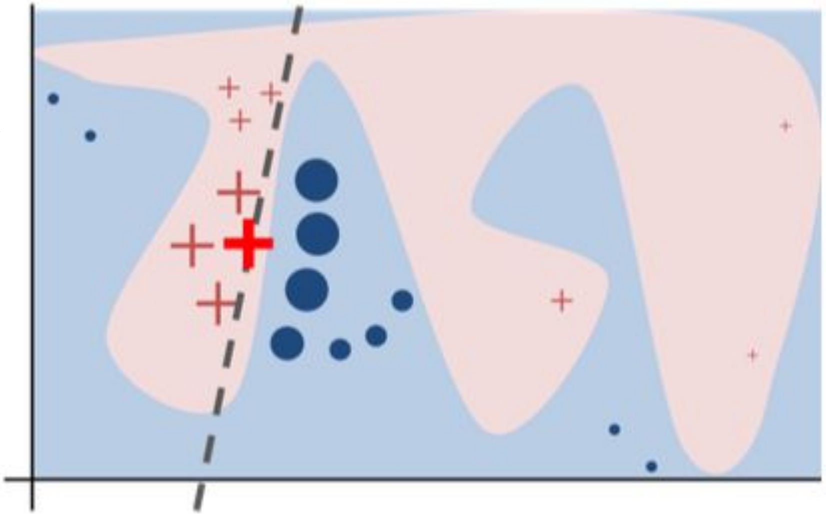
3. Make predictions on this fake data created.

4. Pick **M** features that contributed to predictions.

5. Fit a simple linear model on these M features.

6. And the weights derived for the process act as explanations for local prediction

What I wanted to say? →



The **black-box model’s** complex decision function f (unknown to LIME) is represented by the blue/pink background, which cannot be approximated well by a linear model. The bold red cross is the instance being explained. LIME samples instances, gets predictions using f, and weighs them by the proximity to the instance being explained (represented here by size). The dashed line is the learned explanation that is locally (but not globally) faithful.

What makes LIME a good model explainer?

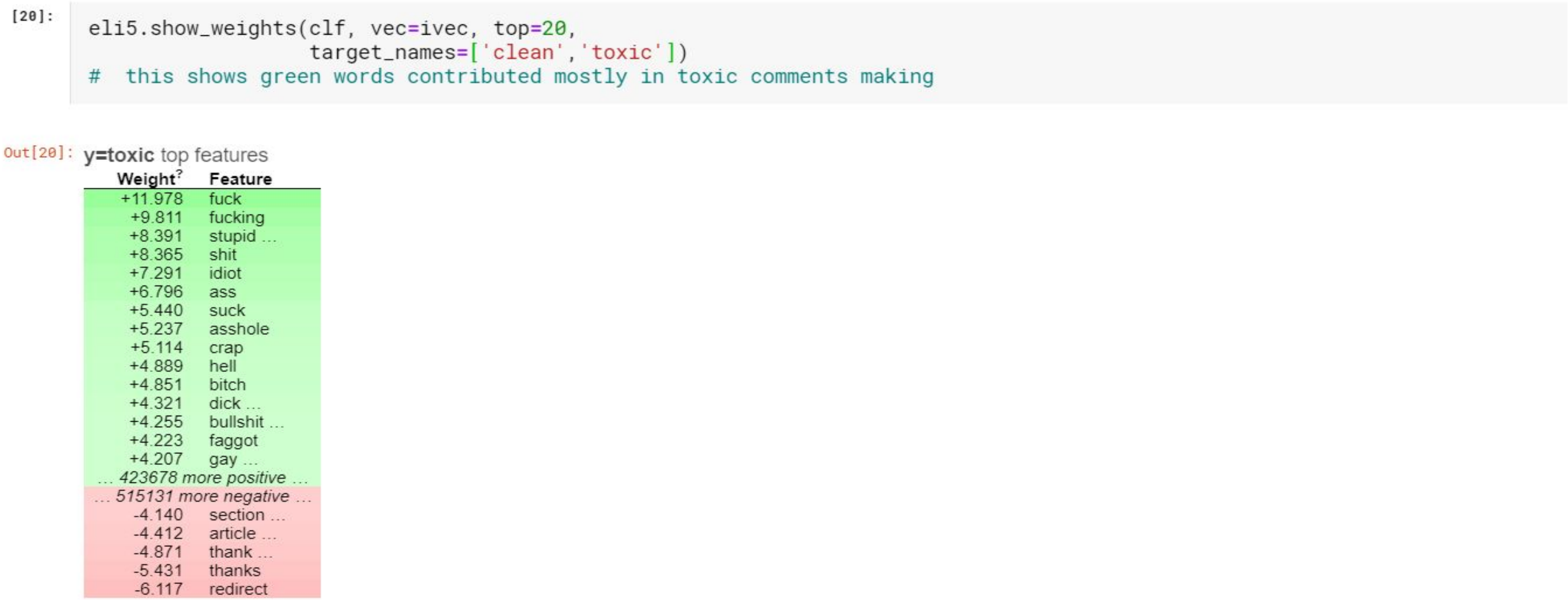
1. Interpretable Data Representations

LIME use a representation that is understood by the humans irrespective of the actual features used by the model. This is coined as interpretable representation. An interpretable representation would vary with the type of data that we are working with for example :

1. For **text** : It represents **presence/absence** of words.
2. For **image** : It represents presence/absence of **super pixels** (contiguous patch of similar pixels).
3. For **tabular data** : It is a **weighted combination** of columns

RESULTS

1. HASHINGVECTORIZER + LOGISTIC REGRESSION



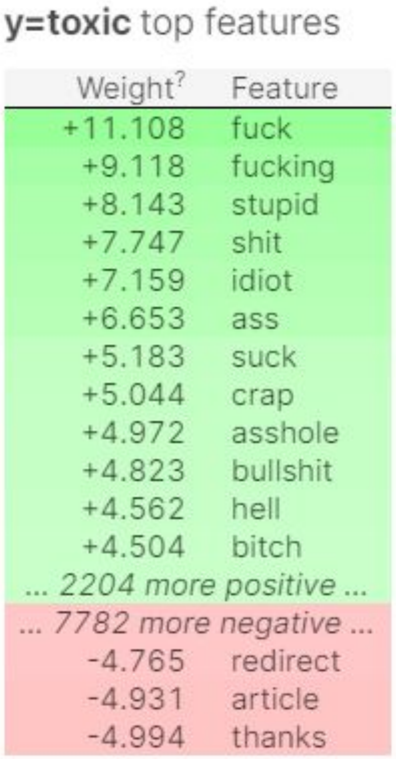
[20]:

```
eli5.show_weights(clf, vec=ivec, top=20,
                  target_names=['clean','toxic'])
#  this shows green words contributed mostly in toxic comments making
```

Out[20]: **y=toxic** top features

Weight?	Feature
+11.978	fuck
+9.811	fucking
+8.391	stupid …
+8.365	shit
+7.291	idiot
+6.796	ass
+5.440	suck
+5.237	asshole
+5.114	crap
+4.889	hell
+4.851	bitch
+4.321	dick …
+4.255	bullshit …
+4.223	faggot
+4.207	gay …
… 423678 more positive …	
… 515131 more negative …	
-4.140	section …
-4.412	article …
-4.871	thank …
-5.431	thanks
-6.117	redirect

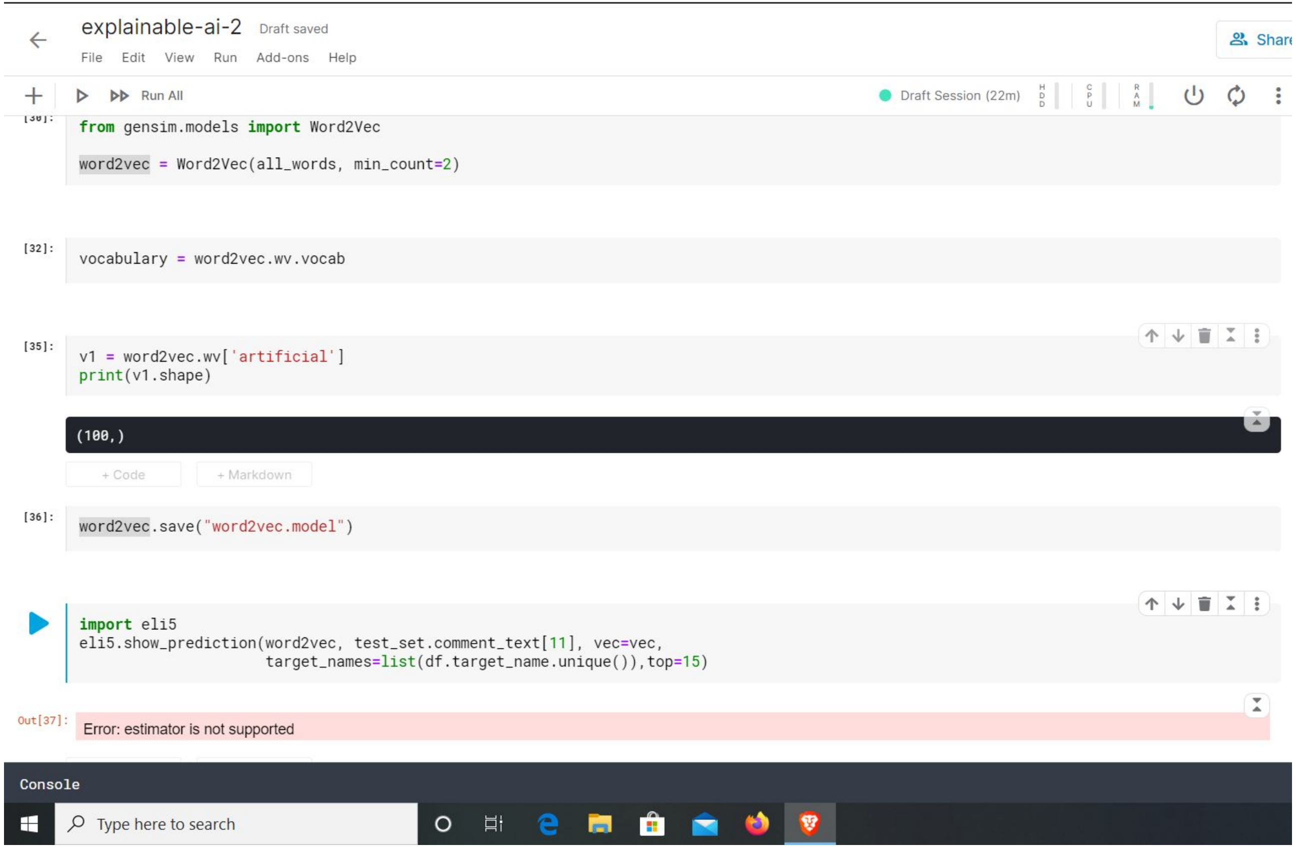
2. **TF/IDF + LOGISTIC REGRESSION** GAVE US:-



y=toxic top features

Weight?	Feature
+11.108	fuck
+9.118	fucking
+8.143	stupid
+7.747	shit
+7.159	idiot
+6.653	ass
+5.183	suck
+5.044	crap
+4.972	asshole
+4.823	bullshit
+4.562	hell
+4.504	bitch
... 2204 more positive ...	
... 7782 more negative ...	
-4.765	redirect
-4.931	article
-4.994	thanks

3. Word2vec Not supported by present version of LIME and ELI5



[30]:
```
from gensim.models import Word2Vec

word2vec = Word2Vec(all_words, min_count=2)
```

[32]:
```
vocabulary = word2vec.wv.vocab
```

[35]:
```
v1 = word2vec.wv['artificial']
print(v1.shape)
```

```
(100,)
```

[36]:
```
word2vec.save("word2vec.model")
```

```
import eli5
eli5.show_prediction(word2vec, test_set.comment_text[11], vec=vec,
                     target_names=list(df.target_name.unique()),top=15)
```

Out[37]:
```
Error: estimator is not supported
```

Future Scope

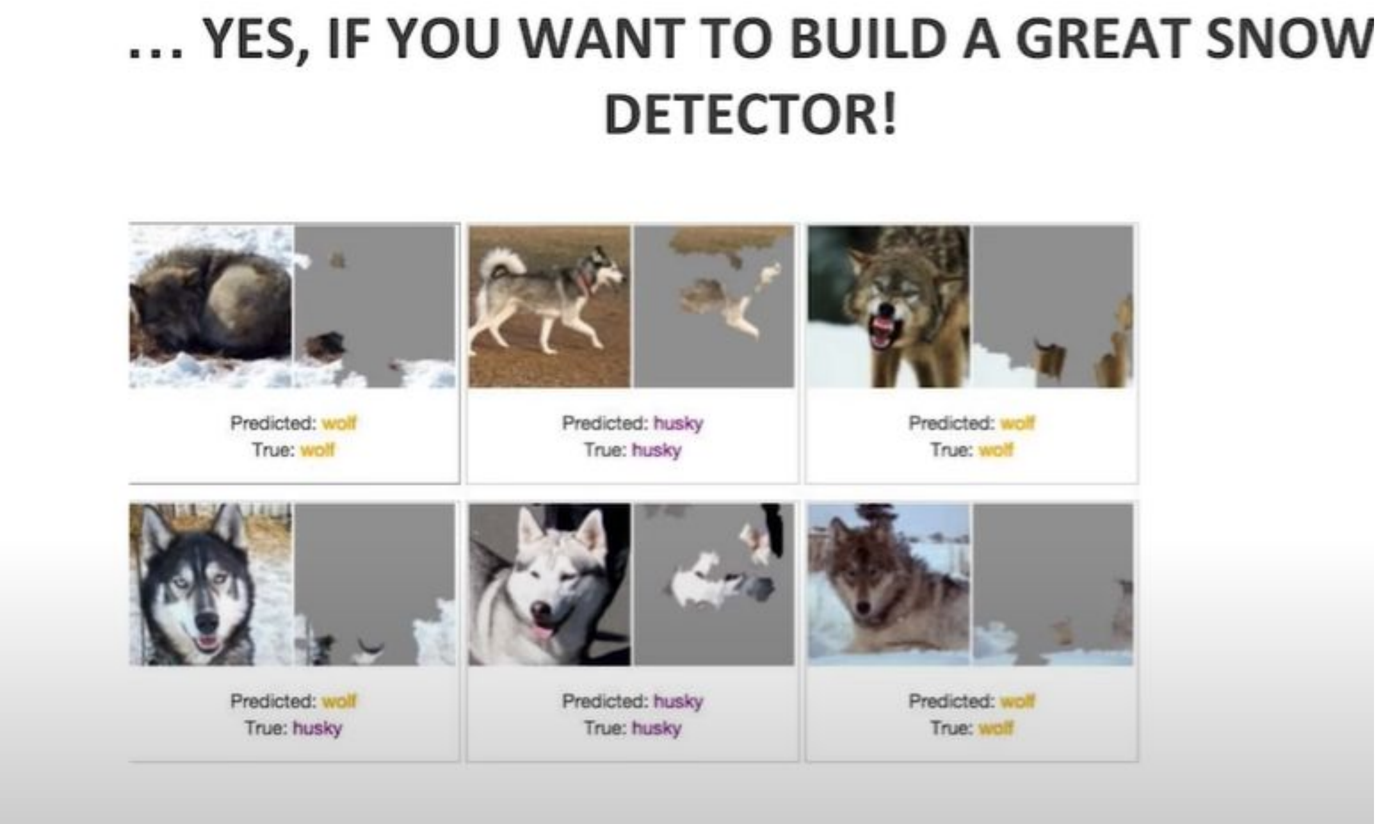
1. Have to understand how Interpretable Machine learning works for **Reinforcement Learning models**.
2. Have to improve existing frameworks because they don’t support majority of present models.
3. How **LIME** works for **Ensemble Models**.

LIMITATIONS

Limitations with ELI5

1. It Does not support **Multinomial Naive Baye**s.
2. It does not support **SVM** with kernels other than **Linear.**
3. It does not support **Keras** models based on **Tensorflow 2.0**
4. It does not support **Pytorch** models.

HOW LIME WORKS FOR IMAGES



… YES, IF YOU WANT TO BUILD A GREAT SNOW DETECTOR!

Predicted: wolf
True: wolf

Predicted: husky
True: husky

Predicted: wolf
True: wolf

Predicted: wolf
True: husky

Predicted: husky
True: husky

Predicted: wolf
True: wolf

THANK YOU