Feature engineering using Local Interpretable Model-agnostic Explanations (LIME)

Recap

A binary classification model may be complex and non-linear, as shown by the blue/pink boundary in the Fig 1. below. A specific instance that needs to be explained is shown as a bright red X. LIME samples instances in the vicinity of this specific instance and their actual class is used to fit a simple linear classifier, without dealing with the complications of global fit (local fidelity). The key feature values of the specific instance, and their contribution to its classification are shown in Fig 2. Details about LIME are published here.

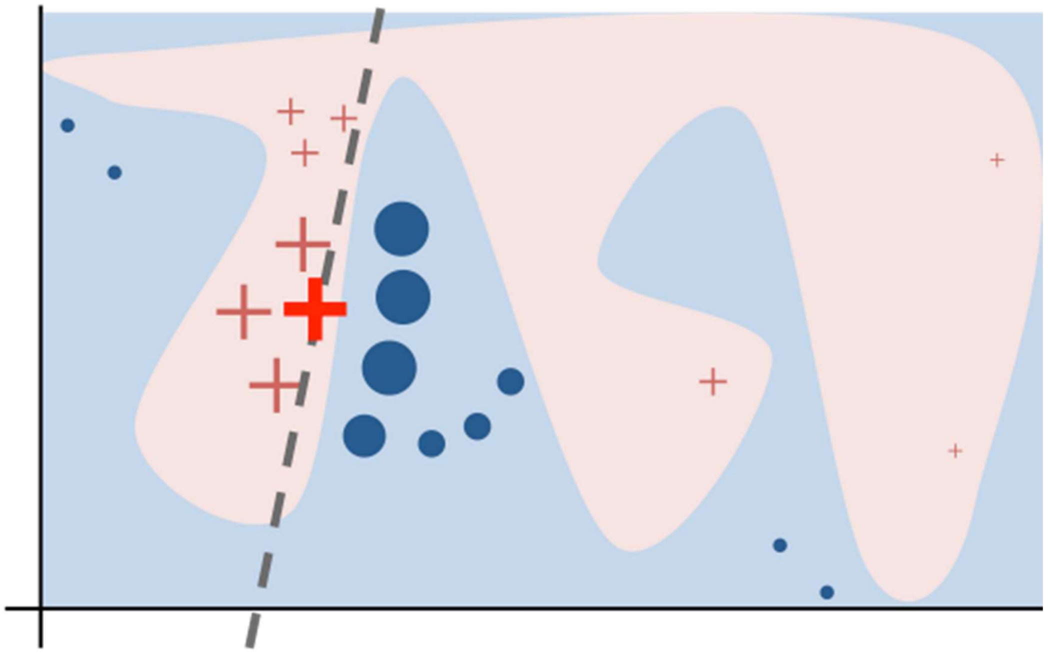
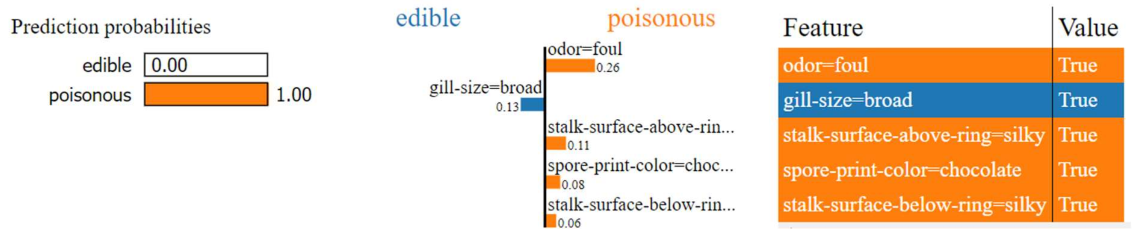


Fig 1: Simple local linear classifier by LIME



Feature	Value
odor=foul	True
gill-size=broad	True
stalk-surface-above-ring=silky	True
spore-print-color=chocolate	True
stalk-surface-below-ring=silky	True

Fig 2: Explaining a prediction as to why a mushroom is poisonous and not edible

While this local classifier can be used explain an individual prediction, it can also be used to improve features. Playing around with the LIME author’s work, here are some observations shared as learnings and not as any original work.

Feature Engineering

Taking the example of a Random Forest classifier that is trying to label TF-IDF vectors of text related to baseball or hockey and going by its metrics, the model appears to perform quite well.

Code:
https://colab.research.google.com/drive/1LtOEWFJ0PUbvGJgPXgWTZLgYpiGnFp5B#scrollTo=lkPnhvrdE2ZG

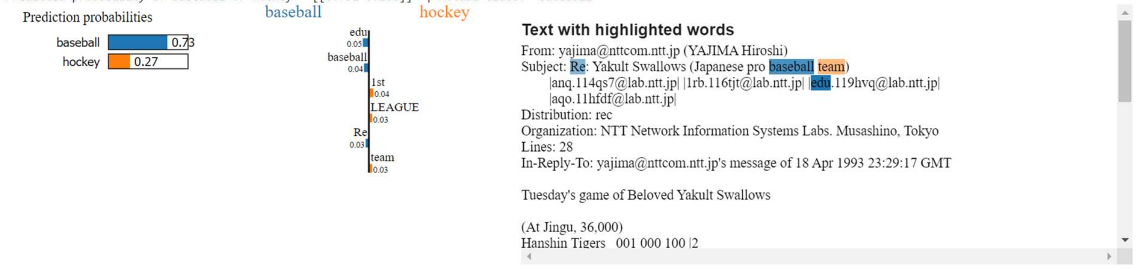
```
sklearn.metrics.precision_recall_fscore_support(df_test['target'], pred, average='binary')
```

```
(0.9536784741144414, 0.8771929824561403, 0.9138381201044387, None)
```

But when we examine the features used, irrelevant details like the sender's name and mail id are used as features (see comments in green). Clearly, the model cannot be generalized to unseen text and classify correctly. This is an important insight that LIME provides.

```
idx = 9 #idx 9 correctly identifies baseball, but the top feature is the irrelevant 'edu'
print('Predicted probability of baseball or hockey: ',c.predict_proba([df_test['text'][idx]]), ' | Actual class: ',class_names[df_test['target'][idx]])
exp = explainer.explain_instance(' '.join([df_test['text'][idx]]), c.predict_proba, num_features=6)
exp.show_in_notebook(text=True)
```

```
Predicted probability of baseball or hockey:  [[0.732 0.268]]  | Actual class:  baseball
```



Text with highlighted words

From: yajima@nttcom.ntt.jp (YAJIMA Hiroshi)
Subject: Re: Yakult Swallows (Japanese pro baseball team)
|anq.114qs7@lab.ntt.jp| |1rb.116tjt@lab.ntt.jp| |edu.119hvq@lab.ntt.jp|
|aqo.11hfdf@lab.ntt.jp|
Distribution: rec
Organization: NTT Network Information Systems Labs. Musashino, Tokyo
Lines: 28
In-Reply-To: yajima@nttcom.ntt.jp's message of 18 Apr 1993 23:29:17 GMT

Tuesday's game of Beloved Yakult Swallows

(At Jingu, 36,000)
Hanshin Tigers   001 000 100 |2

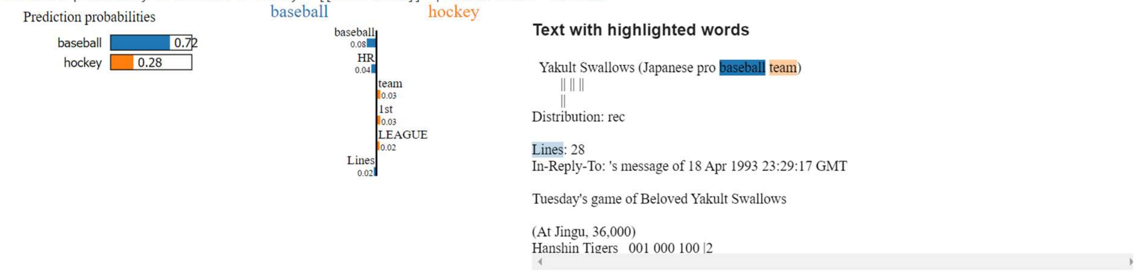
A simple regex function written to strip out irrelevant details, improving features fed to the model.

```
def remUninformative(text):
  import re
  next_text = re.sub(r'[A-Za-z0-9.]*@[A-Za-z0-9.]*', "", text)#email adress
  next_text = re.sub(r'^From:.*', "", next_text, flags=re.M)#From line completely
  next_text = re.sub(r'^Organization:.*', "", next_text, flags=re.M)#Organization line completely
  next_text = re.sub(r'Re:', "", next_text)#Just the tag
  next_text = re.sub(r'Subject:', "", next_text)#Just the tag
  return next_text
```

The retrained classifier provides greater confidence that it is using relevant features and that it can generalize better.

```
idx = 9 #idx 9 correctly identifies baseball, and the top feature is the much improved, if too easy 'baseball'
print('Predicted probability of baseball or hockey: ',c.predict_proba([df_test['text'][idx]]), ' | Actual class: ',class_names[df_test['target'][idx]])
exp = explainer.explain_instance(' '.join([df_test['text2'][idx]]), c.predict_proba, num_features=6)
exp.show_in_notebook(text=True)
```

```
Predicted probability of baseball or hockey:  [[0.744 0.256]]  | Actual class:  baseball
```



Text with highlighted words

Yakult Swallows (Japanese pro baseball team)
|| || ||
||
Distribution: rec

Lines: 28
In-Reply-To: 's message of 18 Apr 1993 23:29:17 GMT

Tuesday's game of Beloved Yakult Swallows

(At Jingu, 36,000)
Hanshin Tigers   001 000 100 |2

But the metrics haven't changed/ improved much.

```
sklearn.metrics.precision_recall_fscore_support(df_test['target'], pred, average='binary')
```

```
(0.9885057471264368, 0.8621553884711779, 0.9210174029451138, None)
```

Conclusion

The similarity in model metrics before and after feature engineering, demonstrates the need to go beyond metrics to choose features. Using LIME, the impact of features used can be understood better, enabling improved feature engineering and model generalize-ability.

Code:
https://colab.research.google.com/drive/1LtOEWFJ0PUbvGJgPXgWTZLgYpiGnFp5B#scrollTo=lkPnhvrdE2ZG