Interpretation of Models

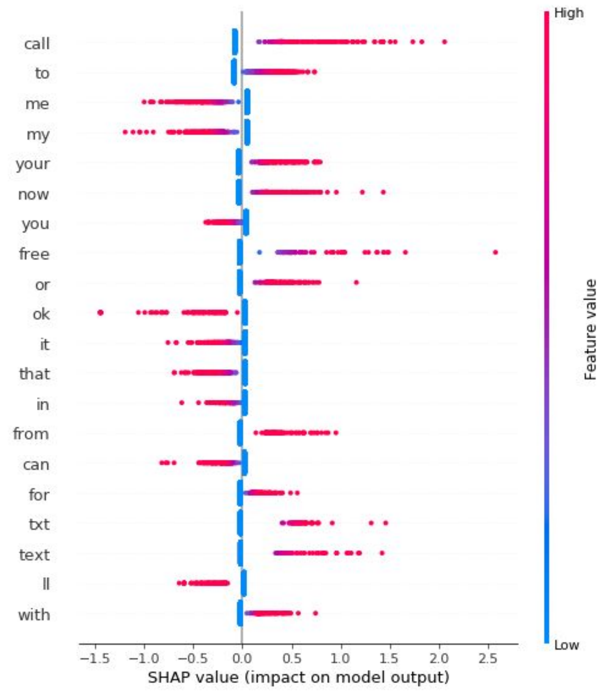
1 . Shap (SHapley Additive exPlanations) :

It illustrates the model dependencies with the help of values called **shapley_values.** These values describe the feature's impact on the model. (either collectively or Independently).

Pros:

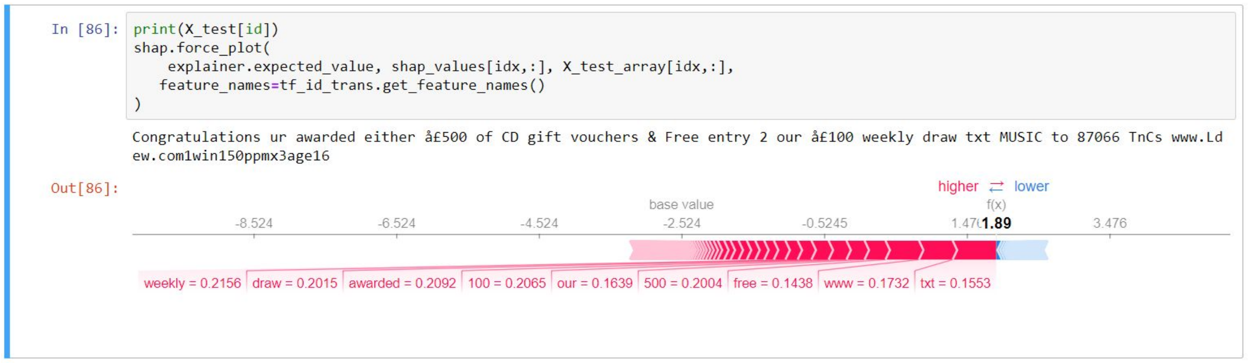
- **Summary_plot :** Shap has very informative plot called summary plot which gives the overall picture of features and their impact on model

```
In [85]: shap.summary_plot(shap_values, X_test_tfid.toarray(), feature_names=tf_id_trans.get_feature_names())
```



The words `call` `to` `me` `my` `your` ... has the highest SHAP values. That means they have the highest impact on model to choose between spam ar
ham

- The shap values generated help to interpret the values easily and identify the cost of impact
- Force plot provides the flexibility for the user to plot the feature impact on models for required predictions



```
In [86]: print(X_test[id])
         shap.force_plot(
             explainer.expected_value, shap_values[idx,:], X_test_array[idx,:],
            feature_names=tf_id_trans.get_feature_names()
         )

         Congratulations ur awarded either â£500 of CD gift vouchers & Free entry 2 our â£100 weekly draw txt MUSIC to 87066 TnCs www.Ld
         ew.com1win150ppmx3age16
Out[86]:
```

- LinearExplainer is fast when compared to other Shap models

Cons :

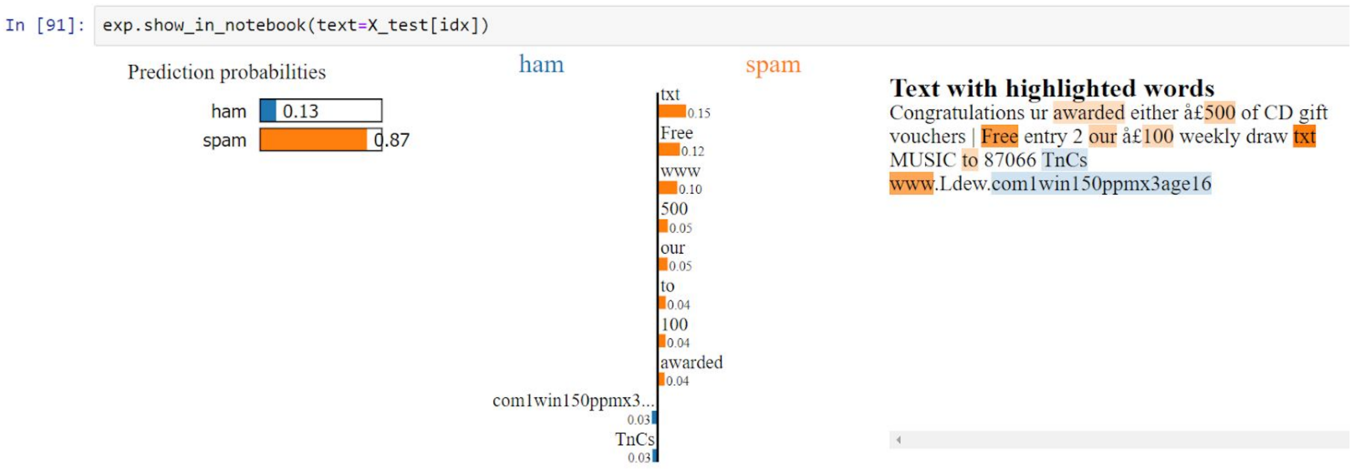
- KernalShap is computationally very slow

2 . Lime (Local Interpretable Model-Agnostic Explanations) :

Lime takes a classifier and raw text or numpy array into a function and explains the reason behind predicting the output. At present lime works on the classifiers with two or more classes.

Pros:

- Lime works very well on text classifiers
- It is very simple to implement and very quick in response, even for huge datasets
- For textual data Lime highlights each word which is responsible for the prediction



```
In [91]: exp.show_in_notebook(text=X_test[idx])
```

Congratulations ur awarded either â£500 of CD gift vouchers | Free entry 2 our â£100 weekly draw txt MUSIC to 87066 TnCs www.Ldew.com1win150ppmx3age16

Cons :

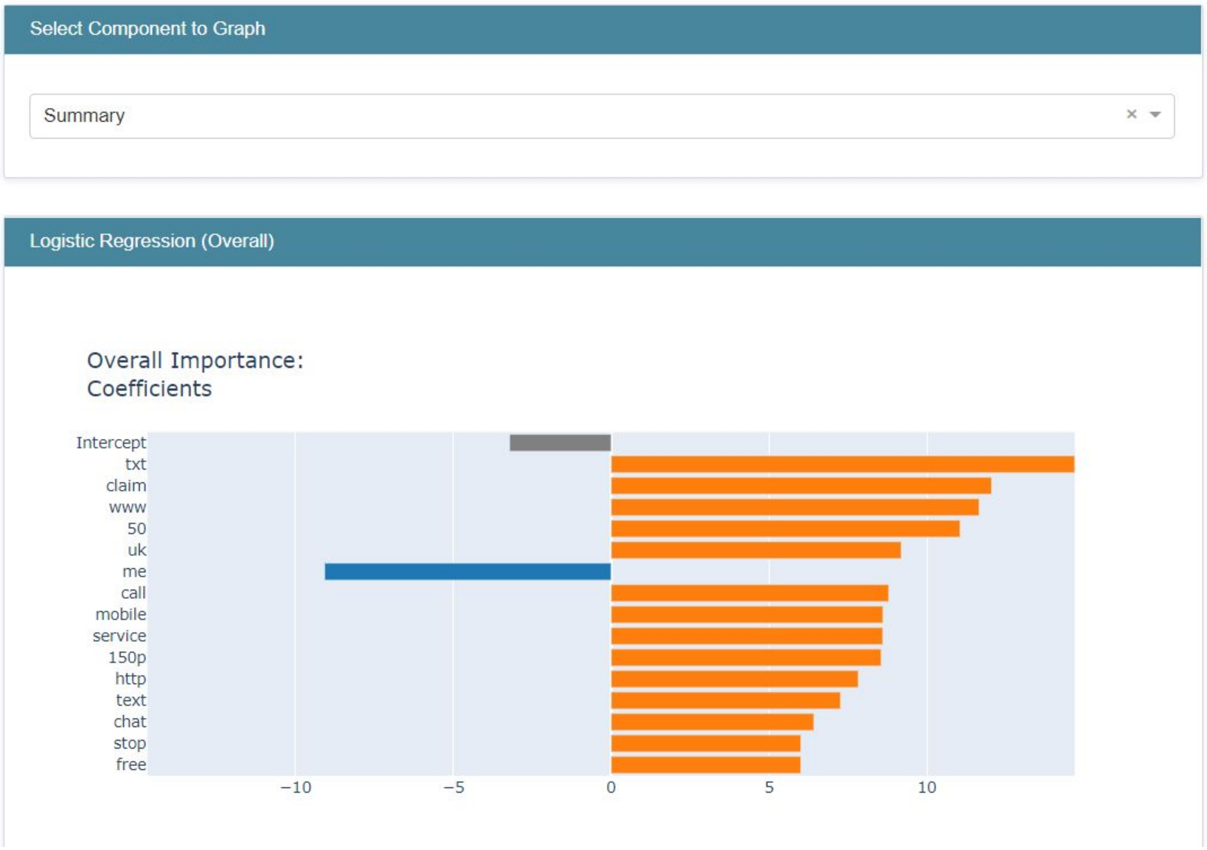
- Lime is not consistent when compared to Shap

3 . InterpretML :

InterpretML helps find the model's global behaviour and also individual predictions. It brings all the interpretable techniques such as Shap, Lime into one roof.

Pros:

- It provides a dashboard to play with around the features and its impact on model



- It brings all the techniques in to one place

Cons :

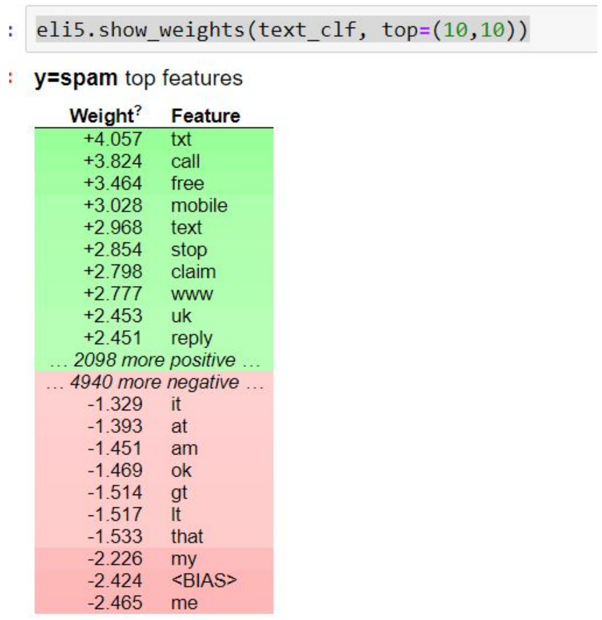
- It consumes more resources as it extracts all the possibilities of feature relations and its coefficients.

4 . ELI5:

It inspects model parameters based on weights and tries to figure out how the model works globally. It inspects an individual prediction of a model, and identifies the reason behind the prediction.

Pros :

- We can see the most weighted features which has the impact on the model globally

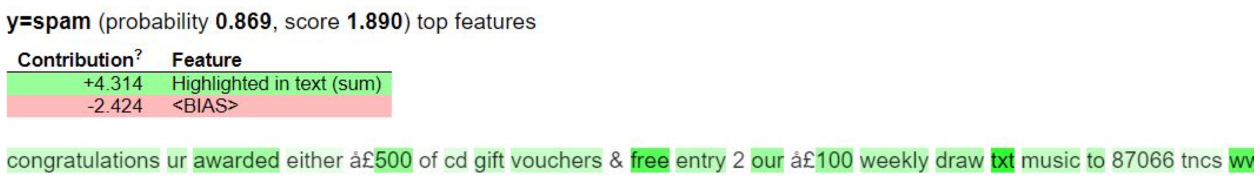


```
: eli5.show_weights(text_clf, top=(10,10))
```

y=spam top features

Weight?	Feature
+4.057	txt
+3.824	call
+3.464	free
+3.028	mobile
+2.968	text
+2.854	stop
+2.798	claim
+2.777	www
+2.453	uk
+2.451	reply
… 2098 more positive …	
… 4940 more negative …	
-1.329	it
-1.393	at
-1.451	am
-1.469	ok
-1.514	gt
-1.517	lt
-1.533	that
-2.226	my
-2.424	<BIAS>
-2.465	me

- We can view the individual prediction along with the highlighted words that are responsible for the prediction



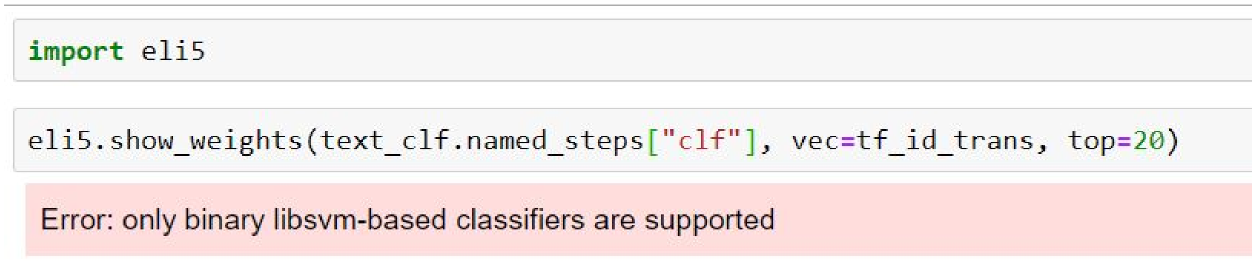
y=spam (probability **0.869**, score **1.890**) top features

Contribution?	Feature
+4.314	Highlighted in text (sum)
-2.424	<BIAS>

congratulations ur awarded either å£500 of cd gift vouchers & free entry 2 our å£100 weekly draw txt music to 87066 tncs wv

Cons :

- Multi class availability is limited



```
import eli5
```

```
eli5.show_weights(text_clf.named_steps["clf"], vec=tf_id_trans, top=20)
```

```
Error: only binary libsvm-based classifiers are supported
```