Machine Intelligence - CO472

Automatic label prediction for GitHub issues

Tejas R
16CO148
B.Tech 8th semester
9845762892
tejas1908@gmail.com

Sample execution results

1. First five rows of the raw data file.

[5]:
```
# Importing the data into a dataframe
issues=pd.read_csv('../input/curated-issues-100/Curated issues-100.csv')
issues.head()
```

Out[5]:

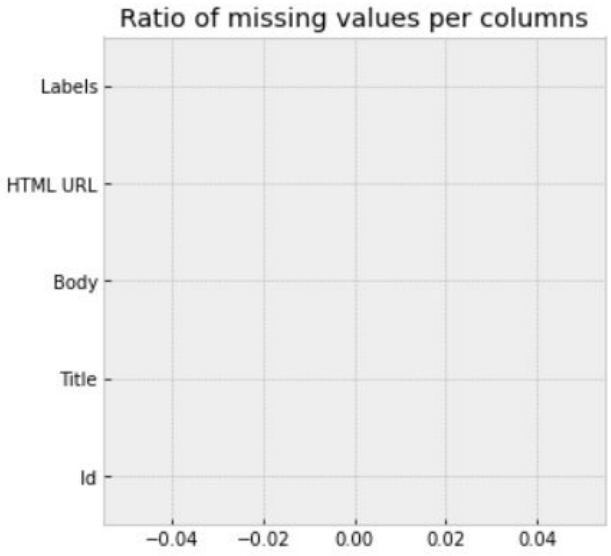
	Id	Title	Body	Created At	Updated At	Closed At	HTML URL	Status	is_locked	Role	Labels
0	603077341.0	b'[SPARK-31489] [SPARK-31488][SQL] Translate da...	b'### What changes were proposed in this pull ...	2020-04-20T09:16:13Z	2020-04-20T09:24:37Z	NaN	https://github.com/apache/spark/pull/28272	open	FALSE	CONTRIBUTOR	b'SQL'
1	603063137.0	b'[SPARK-31495][SQL] Support formatted explain...	b"<!--\r\nThanks for sending a pull request! ...	2020-04-20T08:56:30Z	2020-04-20T09:03:07Z	NaN	https://github.com/apache/spark/pull/28271	open	FALSE	CONTRIBUTOR	b'SQL'
2	602931061.0	b'[SPARK-31494][ML] flatten the result datafra...	b'### What changes were proposed in this pull ...	2020-04-20T04:32:21Z	2020-04-20T06:09:06Z	NaN	https://github.com/apache/spark/pull/28270	open	FALSE	CONTRIBUTOR	b'ML'
3	602921805.0	b'[SPARK-31493][SQL] Optimize InSet to In acco...	b"<!--\r\nThanks for sending a pull request! ...	2020-04-20T04:02:56Z	2020-04-20T07:39:06Z	NaN	https://github.com/apache/spark/pull/28269	open	FALSE	CONTRIBUTOR	b'SQL'
4	602917700.0	b'[SPARK-31492][ML] flatten the result datafra...	b'### What changes were proposed in this pull ...	2020-04-20T03:48:38Z	2020-04-20T05:31:07Z	NaN	https://github.com/apache/spark/pull/28268	open	FALSE	CONTRIBUTOR	b'ML'

+ Code + Markdown

2. A plot of the number of missing values of various columns.

[15]:
```
plt.figure(figsize=(5, 5))
issues.isnull().mean(axis=0).plot.barh()
plt.title("Ratio of missing values per columns")
# new_df.isnull().mean(axis=0).head()
```

Out[15]:
```
Text(0.5, 1.0, 'Ratio of missing values per columns')
```



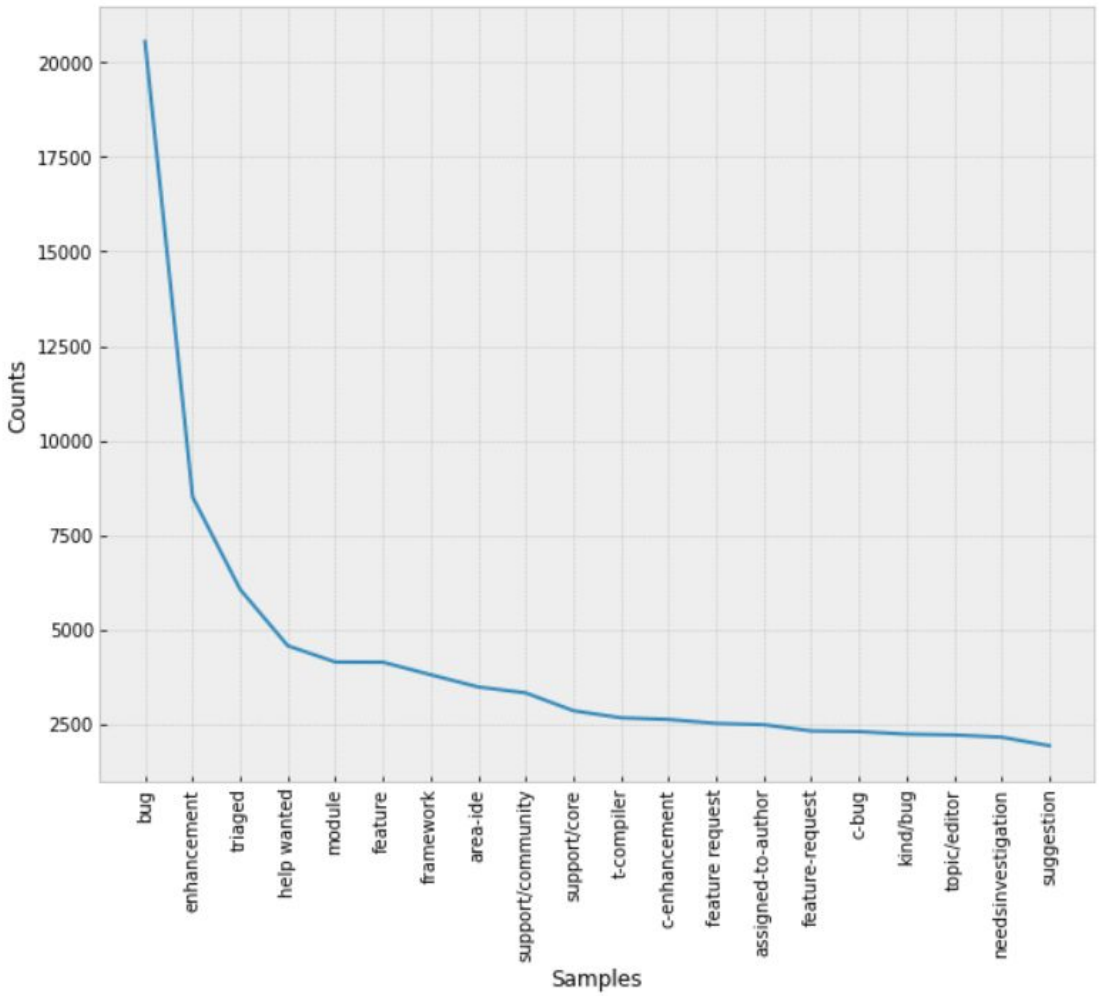
3. After pre-processing the labels.

[23]:
```
issues['Labels']=issues['Labels'].apply(lambda x: replace_char_with_char(x))
issues.head()
```

Out[23]:

	Id	Title	Body	HTML URL	Labels
0	603077341.0	[SPARK-31489][SPARK-31488][SQL] Translate date...	### What changes were proposed in this pull re...	https://github.com/apache/spark/pull/28272	[sql]
1	603063137.0	[SPARK-31495][SQL] Support formatted explain f...	"<!--\r\nThanks for sending a pull request! H...	https://github.com/apache/spark/pull/28271	[sql]
2	602931061.0	[SPARK-31494][ML] flatten the result dataframe...	### What changes were proposed in this pull re...	https://github.com/apache/spark/pull/28270	[ml]
3	602921805.0	[SPARK-31493][SQL] Optimize InSet to In accord...	"<!--\r\nThanks for sending a pull request! H...	https://github.com/apache/spark/pull/28269	[sql]
4	602917700.0	[SPARK-31492][ML] flatten the result dataframe...	### What changes were proposed in this pull re...	https://github.com/apache/spark/pull/28268	[ml]

4. **Frequency distribution of the top 20 most frequently occurring labels.**



5. Sample body text before and after pre-processing.

[45]:
```
issues['Body'].iloc[12345]
```

Out[45]:
```
'Given the following code:\\n\\n``` rust\\nextern crate mio;\\nuse mio::buf::RingBuf;\\nuse mio::buf::Buf;\\nuse std::io::Read;\\n\\nfn main() {\\n    let buf = RingBuf::new(10);\\n    let bytes = buf.bytes();\\n    println!("{:?}", bytes);\\n}\\n```\\n\\n`buf` is of type `RingBuf`. `RingBuf` does not provide `.bytes()`, but it implements both `Buf` and `Read`, which both provide a `.bytes()` implementation.\\n\\nAccording to https://doc.rust-lang.org/book/ufcs.html the compiler should complain. But it simply chooses to take the implementation of `Read`, which return the "wrong" result type.\\n\\n(Without the "use Read" line rust chooses the implementation of `Buf`, which return the "correct" type.)\\n\\n(I am using Rust 1.0.0.)\\n'
```

+ Code + Markdown

```
issues['Body'].iloc[12345]
```

Out[84]:
```
'give follow code rust extern crate mio use mio buf ringbuf use mio buf buf use std io read fn main let buf ringbuf new let bytes buf bytes println bytes buf type ringbuf ringbuf do not provide bytes but it implement buf read provide bytes implementation accord to compiler complain but it simply choose to take implementation read return wrong result type without use read line rust choose implementation buf return correct type use rust'
```

+ Code + Markdown

6. Tf-idf matrix for issue data.

[58]:
```
# print(TF_IDF_matrix.shape)
temp=TF_IDF_matrix
dense=temp.todense()
# print(dense)
denselist=dense.tolist()
# print(denselist)

tfidf=pd.DataFrame(denselist,columns=vectorizer_train.get_feature_names())
print(len(tfidf))
tfidf.head()
```

```
95188
```

Out[58]:

	aa	ab	able	ac	accept	access	account	action	active	actual	...	xef	xf	xml	xxx	yaml	yes	yml	zero	zip	zone
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.133680	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.053425	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.183145	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.136587	0.0	0.0	0.0	0.0

5 rows × 1000 columns

7. **Sample topics returned by the LDA algorithm.**

```
----------------------------------------
Topic 15:
microsoft azure github service content to xe id type ts
----------------------------------------
----------------------------------------
Topic 16:
pr fix sign merge issue pull address feature request component
----------------------------------------
----------------------------------------
Topic 17:
to in it version on behavior not godot reproduce issue
----------------------------------------
----------------------------------------
Topic 18:
istio task codeanalysis language microsoft at async system extension extensions
----------------------------------------
----------------------------------------
Topic 19:
foo bar symbol impl suggest aten www emit caffe arg
----------------------------------------
```

8. **Evaluation of classical classifiers.**

```
Clf:  MultinomialNB
Jacard score: 31.901179413695523
Precision: 36.28991129270308
Recall: 45.536907312435204
F score: 37.428235874456675
Hamming loss: 1.7100535770564134
---
Clf:  LinearSVC
Jacard score: 44.967274732481165
Precision: 53.23764439270269
Recall: 50.227823194652686
F score: 49.66095702119333
Hamming loss: 1.1760163882760795
---
Clf:  Perceptron
Jacard score: 41.04145901518262
Precision: 47.978901125371344
Recall: 53.215905661146756
F score: 47.65136038246365
Hamming loss: 1.6669292992961444
---
```

9. Evaluation of MLP.

```
[79]: print_score(y_pred_new,model)
```

```
Clf:  Sequential
Jacard score: 52.47224468840448
Precision: 56.87397970765446
Recall: 70.30190795594199
F score: 60.204616875087225
Hamming loss: 1.444479462128375
---
```

+ Code + Markdown

10. A Linear SVC classifier chain performance.