PyCaret Using Google Drive

```
# installations
!pip install -U tensorflow-gpu==2.0.0 grpcio
!pip install pycaret
!pip install -U -q PyDrive
```

```
# imports
import numpy as np
import pandas as pd
from pycaret.classification import *

# Code to read csv file into Colaboratory:
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials
```

```
# Authenticate and create the PyDrive client.
auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)
```

Generate Data

```
# Generating Dataframe for taxonomic level MANUAL
link = "https://drive.google.com/file/d/1AEsr95ktU2AxfEQenfvX_qqaFNBWzgNB/view?usp=s

# to get the id part of the file
id = link.split("/")[-2]

downloaded = drive.CreateFile({'id':id})
downloaded.GetContentFile("training.csv")

training_df = pd.read_csv('training.csv')
#df = df.drop(columns = 'Unnamed: 0')
print(training_df)
```

```
          Sublevel Name  pp_magtropy  pp_avg_magnitude   entropy
   0    Kitrinoviricota    66.423098         91.567643  1.378551
   1    Kitrinoviricota    66.028342         90.635067  1.372669
```

```
2    Kitrinoviricota    65.510853         89.786372  1.370557
3    Kitrinoviricota    65.716921         90.206633  1.372655
4    Kitrinoviricota    48.519912         67.045521  1.381815
..               ...          ...               ...       ...
495  Negarnaviricota    71.185677         97.253325  1.366192
496  Negarnaviricota    78.482111        108.415361  1.381402
497  Negarnaviricota    30.172801         41.396704  1.371987
498  Negarnaviricota    26.894834         36.865421  1.370725
499  Negarnaviricota    24.674267         33.825194  1.370869

[500 rows x 4 columns]
```

```
# Generating Dataframe for COVID-19 Sequences
testing_link = "https://drive.google.com/file/d/1_SxcTlA9dDIergs__seb-DbnifluBQF6/vi

sublevel = input("Sublevel of Testing Data: ")
# to get the id part of the file
id = testing_link.split("/")[-2]

downloaded = drive.CreateFile({'id':id})
downloaded.GetContentFile('testing.csv')

testing_df = pd.read_csv('testing.csv')
testing_df = testing_df.drop(columns = 'Unnamed: 0')
testing_df = testing_df[testing_df['Sublevel Name'] == sublevel]
print(testing_df)
```

```
Sublevel of Testing Data: Embecovirus
     Sublevel Name  pp_magtropy  pp_avg_magnitude   entropy
112   Embecovirus   114.269624        153.103733  1.339846
113   Embecovirus   114.111031        155.141480  1.359566
114   Embecovirus   114.987320        153.815693  1.337675
115   Embecovirus   114.226726        153.062393  1.339988
116   Embecovirus   114.320187        153.136267  1.339538
..            ...          ...               ...       ...
207   Embecovirus   112.497193        153.807531  1.367212
208   Embecovirus   114.288491        153.117355  1.339744
209   Embecovirus   114.870606        153.996769  1.340611
210   Embecovirus   115.440977        150.518479  1.303857
211   Embecovirus   114.422743        153.317131  1.339918

[100 rows x 4 columns]
```

Magtropy

```
magtropy_df = training_df.drop(columns = ["pp_avg_magnitude", "entropy"])
print(magtropy_df)
```

```
       Sublevel Name  pp_magtropy
0    Kitrinoviricota    66.423098
1    Kitrinoviricota    66.028342
```

```
2    Kitrinoviricota    65.510853
3    Kitrinoviricota    65.716921
4    Kitrinoviricota    48.519912
..               ...          ...
495  Negarnaviricota    71.185677
496  Negarnaviricota    78.482111
497  Negarnaviricota    30.172801
498  Negarnaviricota    26.894834
499  Negarnaviricota    24.674267

[500 rows x 2 columns]
```

```
experiment = setup(data=magtropy_df, target='Sublevel Name')
# if the error states target is not defined, change from Sublevel_Name to Sublevel N
# label encodings alphabetical
```

	Description	Value
0	session_id	7097
1	Target	Sublevel Name
2	Target Type	Multiclass
3	Label Encoded	Duplornaviricota: 0, Kitrinoviricota: 1, Lenar...
4	Original Data	(500, 2)
5	Missing Values	False
6	Numeric Features	1
7	Categorical Features	0
8	Ordinal Features	False
9	High Cardinality Features	False
10	High Cardinality Method	None
11	Transformed Train Set	(349, 1)
12	Transformed Test Set	(151, 1)
13	Shuffle Train-Test	True
14	Stratify Train-Test	False
15	Fold Generator	StratifiedKFold
16	Fold Number	10
17	CPU Jobs	-1
18	Use GPU	False
19	Log Experiment	False
20	Experiment Name	clf-default-name
21	USI	54da
22	Imputation Type	simple
23	Iterative Imputation Iteration	None
24	Numeric Imputer	mean
25	Iterative Imputation Numeric Model	None
26	Categorical Imputer	constant
27	Iterative Imputation Categorical Model	None
28	Unknown Categoricals Handling	least_frequent
29	Normalize	False
30	Normalize Method	None

30	Normalize Method	None
31	Transformation	False
32	Transformation Method	None
33	PCA	False
34	PCA Method	None
35	PCA Components	None

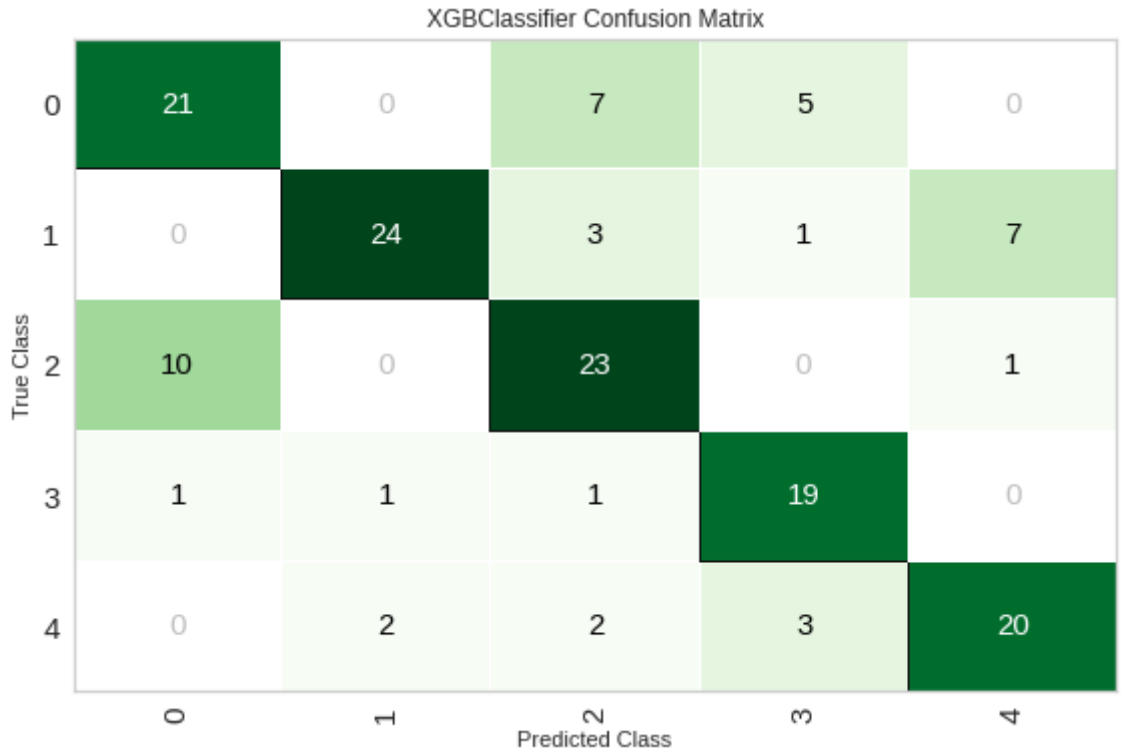
```
compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
xgboost	Extreme Gradient Boosting	0.7307	0.9184	0.7306	0.7460	0.7259	0.6627	0.6684	2.234
et	Extra Trees Classifier	0.7280	0.8802	0.7268	0.7404	0.7226	0.6594	0.6646	0.473
gbc	Gradient Boosting Classifier	0.7279	0.9150	0.7260	0.7428	0.7208	0.6591	0.6656	0.356
knn	K Neighbors Classifier	0.7253	0.9076	0.7288	0.7525	0.7206	0.6567	0.6649	0.120
dt	Decision Tree Classifier	0.7250	0.8278	0.7239	0.7395	0.7198	0.6556	0.6613	0.022
rf	Random Forest Classifier	0.7250	0.9169	0.7239	0.7395	0.7198	0.6556	0.6613	0.504
lightgbm	Light Gradient Boosting Machine	0.7193	0.9176	0.7227	0.7459	0.7158	0.6488	0.6556	0.112
catboost	CatBoost Classifier	0.7133	0.9273	0.7135	0.7309	0.7095	0.6408	0.6466	1.397
nb	Naive Bayes	0.5445	0.7734	0.5690	0.4623	0.4688	0.4364	0.4662	0.021
qda	Quadratic Discriminant Analysis	0.5445	0.7742	0.5690	0.4623	0.4688	0.4364	0.4662	0.021
ada	Ada Boost	0.4069	0.7657	0.4070	0.3280	0.3165	0.2514	0.3113	0.102

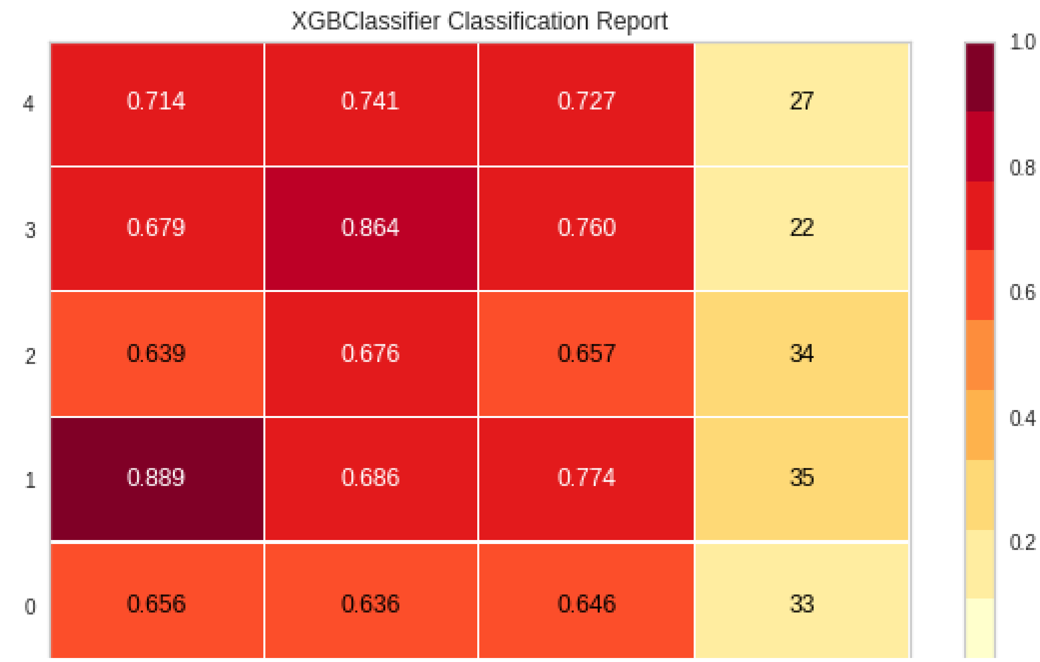
```
estimator = create_model('xgboost')
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.7143	0.8903	0.7167	0.7479	0.7169	0.6414	0.6461
1	0.5714	0.7847	0.5798	0.5848	0.5723	0.4643	0.4676
2	0.8286	0.9553	0.8238	0.8471	0.8239	0.7853	0.7901
3	0.8000	0.9786	0.8000	0.8245	0.7935	0.7500	0.7578
4	0.7143	0.9143	0.7143	0.7279	0.7076	0.6429	0.6488
5	0.7429	0.8973	0.7524	0.7571	0.7403	0.6782	0.6831
6	0.7143	0.9126	0.7250	0.7192	0.7105	0.6432	0.6465
7	0.8000	0.9508	0.7798	0.7810	0.7862	0.7479	0.7503
8	0.6857	0.9401	0.6881	0.6933	0.6747	0.6071	0.6147
9	0.7353	0.9596	0.7262	0.7770	0.7336	0.6667	0.6794
Mean	0.7307	0.9184	0.7306	0.7460	0.7259	0.6627	0.6684
SD	0.0690	0.0523	0.0644	0.0696	0.0670	0.0859	0.0863

```
plot_model(estimator, 'confusion_matrix')
```




```
plot_model(estimator, 'class_report')
```




```
magtropy_testing_df = testing_df.drop(columns = ["pp_avg_magnitude", "entropy"])
print(magtropy_testing_df)

       Sublevel Name  pp_magtropy
   112   Embecovirus   114.269624
   113   Embecovirus   114.111031
   114   Embecovirus   114.987320
   115   Embecovirus   114.226726
   116   Embecovirus   114.320187
   ..            ...          ...
   207   Embecovirus   112.497193
   208   Embecovirus   114.288491
   209   Embecovirus   114.870606
   210   Embecovirus   115.440977
   211   Embecovirus   114.422743

   [100 rows x 2 columns]


X_test = magtropy_testing_df.drop(columns = ["Sublevel Name"])
predict = estimator.predict(X_test)
print(predict)
print(len(predict))

   [4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
    4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
    4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4]
   100


unique_elements, count_elements = np.unique(predict, return_counts = "True")
results = np.asarray((unique_elements, count_elements))
print(results)

   [[  4]
    [100]]
```

Magnitude avg

```
avg_magnitude_df = training_df.drop(columns = ["pp_magtropy", "entropy"])
print(avg_magnitude_df)
```

```
         Sublevel Name  pp_avg_magnitude
   0    Kitrinoviricota         91.567643
   1    Kitrinoviricota         90.635067
   2    Kitrinoviricota         89.786372
   3    Kitrinoviricota         90.206633
   4    Kitrinoviricota         67.045521
   ..               ...               ...
   495  Negarnaviricota         97.253325
   496  Negarnaviricota        108.415361
   497  Negarnaviricota         41.396704
   498  Negarnaviricota         36.865421
   499  Negarnaviricota         33.825194

   [500 rows x 2 columns]
```

```
experiment = setup(data=avg_magnitude_df, target='Sublevel Name')
```

26	Categorical Imputer	constant
27	Iterative Imputation Categorical Model	None
28	Unknown Categoricals Handling	least_frequent
29	Normalize	False
30	Normalize Method	None
31	Transformation	False
32	Transformation Method	None
33	PCA	False
34	PCA Method	None
35	PCA Components	None
36	Ignore Low Variance	False
37	Combine Rare Levels	False
38	Rare Level Threshold	None
39	Numeric Binning	False
40	Remove Outliers	False
41	Outliers Threshold	None
42	Remove Multicollinearity	False
43	Multicollinearity Threshold	None
44	Clustering	False
45	Clustering Iteration	None
46	Polynomial Features	False
47	Polynomial Degree	None
48	Trignometry Features	False
49	Polynomial Threshold	None
50	Group Features	False
51	Feature Selection	False
52	Features Selection Threshold	None
53	Feature Interaction	False
54	Feature Ratio	False
55	Interaction Threshold	None
56	Fix Imbalance	False
57	Fix Imbalance Method	SMOTE

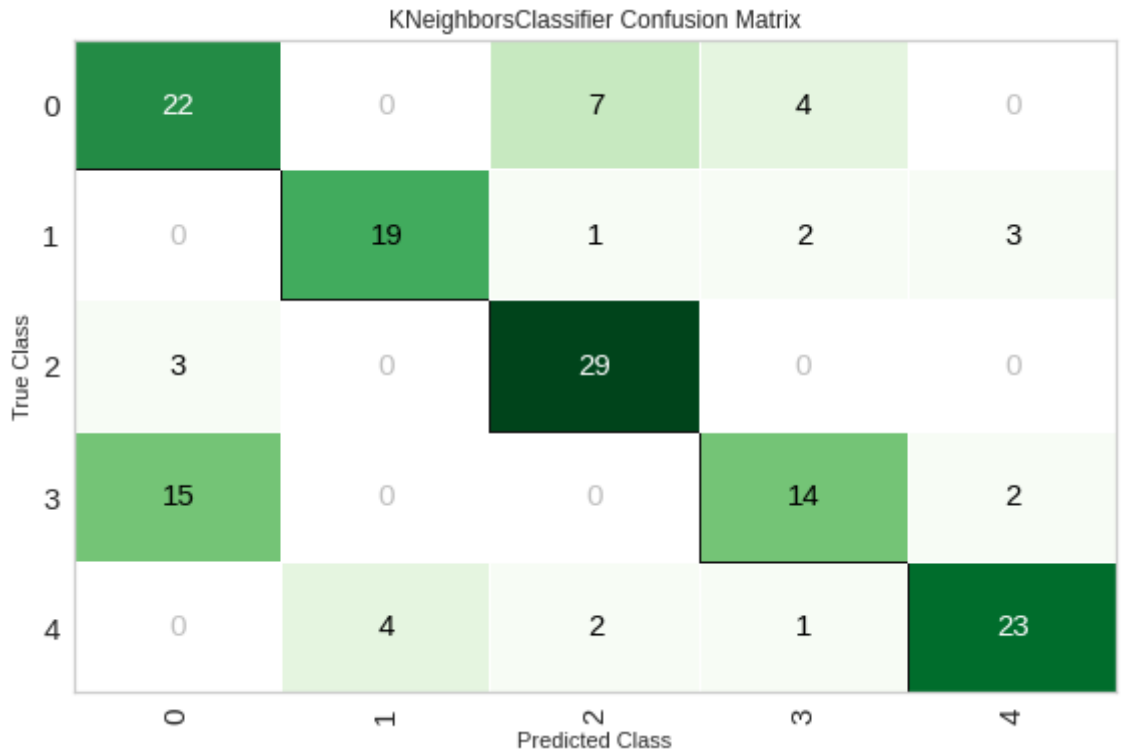
```
compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
knn	K Neighbors Classifier	0.7472	0.8947	0.7449	0.7757	0.7420	0.6837	0.6924	0.120
lightgbm	Light Gradient Boosting Machine	0.7103	0.9091	0.7088	0.7349	0.7053	0.6378	0.6455	0.104
catboost	CatBoost Classifier	0.7101	0.9122	0.7094	0.7289	0.7050	0.6375	0.6446	1.413
et	Extra Trees Classifier	0.7073	0.8718	0.7056	0.7247	0.7020	0.6339	0.6405	0.471
dt	Decision Tree Classifier	0.7045	0.8154	0.7027	0.7240	0.6998	0.6303	0.6368	0.021
rf	Random Forest Classifier	0.7045	0.8983	0.7027	0.7240	0.6998	0.6303	0.6368	0.483
gbc	Gradient Boosting Classifier	0.7045	0.9040	0.7027	0.7219	0.6996	0.6304	0.6366	0.356
xgboost	Extreme Gradient Boosting	0.7045	0.9053	0.7027	0.7240	0.6998	0.6303	0.6368	1.875
nb	Naive	0.5556	0.7588	0.5508	0.4547	0.4739	0.4423	0.4693	0.020

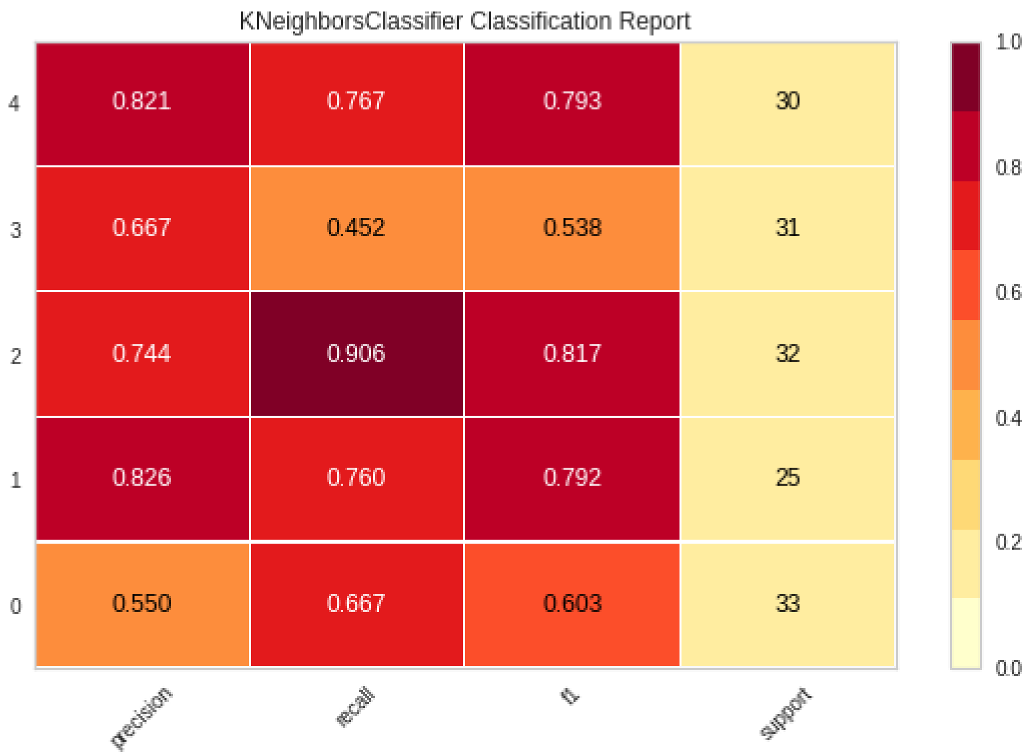
```
estimator = create_model('knn')
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.7429	0.8827	0.7429	0.7698	0.7328	0.6786	0.6878
1	0.7429	0.8500	0.7429	0.8077	0.7470	0.6786	0.6958
2	0.8000	0.9020	0.8000	0.8257	0.7959	0.7500	0.7570
3	0.7429	0.9158	0.7429	0.7481	0.7378	0.6786	0.6821
4	0.8000	0.9136	0.7905	0.8222	0.7963	0.7492	0.7547
5	0.8000	0.9160	0.8107	0.8226	0.7982	0.7503	0.7557
6	0.7429	0.9009	0.7429	0.7858	0.7294	0.6782	0.6940
7	0.8286	0.9596	0.8143	0.8329	0.8221	0.7844	0.7885
8	0.7429	0.9081	0.7381	0.8000	0.7351	0.6786	0.6928
9	0.5294	0.7984	0.5238	0.5426	0.5258	0.4106	0.4156
Mean	0.7472	0.8947	0.7449	0.7757	0.7420	0.6837	0.6924
SD	0.0791	0.0414	0.0795	0.0818	0.0790	0.0990	0.0990

```
plot_model(estimator, 'confusion_matrix')
```




```
plot_model(estimator, 'class_report')
```




```
magnitude_avg_testing_df = testing_df.drop(columns = ["pp_magtropy", "entropy"])
print(magnitude_avg_testing_df )
```

```
      Sublevel Name  pp_avg_magnitude
112    Embecovirus        153.103733
113    Embecovirus        155.141480
114    Embecovirus        153.815693
115    Embecovirus        153.062393
116    Embecovirus        153.136267
..             ...               ...
207    Embecovirus        153.807531
208    Embecovirus        153.117355
209    Embecovirus        153.996769
210    Embecovirus        150.518479
211    Embecovirus        153.317131

[100 rows x 2 columns]
```

```
X_test = magnitude_avg_testing_df.drop(columns = ["Sublevel Name"])
predict = estimator.predict(X_test)
print(predict)
print(len(predict))
```

```
[4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4]
100
```

```
unique_elements, count_elements = np.unique(predict, return_counts = "True")
results = np.asarray((unique_elements, count_elements))
print(results)
```

```
[[  4]
 [100]]
```

Entropy

```
entropy_df = training_df.drop(columns = ["pp_magtropy","pp_avg_magnitude"])
print(entropy_df)
```

```
        Sublevel Name   entropy
0    Kitrinoviricota  1.378551
1    Kitrinoviricota  1.372669
2    Kitrinoviricota  1.370557
3    Kitrinoviricota  1.372655
4    Kitrinoviricota  1.381815
..               ...       ...
495  Negarnaviricota  1.366192
496  Negarnaviricota  1.381402
497  Negarnaviricota  1.371987
498  Negarnaviricota  1.370725
499  Negarnaviricota  1.370869

[500 rows x 2 columns]
```

```
experiment = setup(data=entropy_df, target='Sublevel Name')
```

	Description	Value
0	session_id	3238
1	Target	Sublevel Name
2	Target Type	Multiclass
3	Label Encoded	Duplornaviricota: 0, Kitrinoviricota: 1, Lenar...
4	Original Data	(500, 2)
5	Missing Values	False
6	Numeric Features	1
7	Categorical Features	0
8	Ordinal Features	False
9	High Cardinality Features	False
10	High Cardinality Method	None
11	Transformed Train Set	(349, 1)
12	Transformed Test Set	(151, 1)
13	Shuffle Train-Test	True
14	Stratify Train-Test	False
15	Fold Generator	StratifiedKFold
16	Fold Number	10
17	CPU Jobs	-1
18	Use GPU	False
19	Log Experiment	False
20	Experiment Name	clf-default-name
21	USI	438e
22	Imputation Type	simple
23	Iterative Imputation Iteration	None
24	Numeric Imputer	mean
25	Iterative Imputation Numeric Model	None
26	Categorical Imputer	constant
27	Iterative Imputation Categorical Model	None
28	Unknown Categoricals Handling	least_frequent
29	Normalize	False

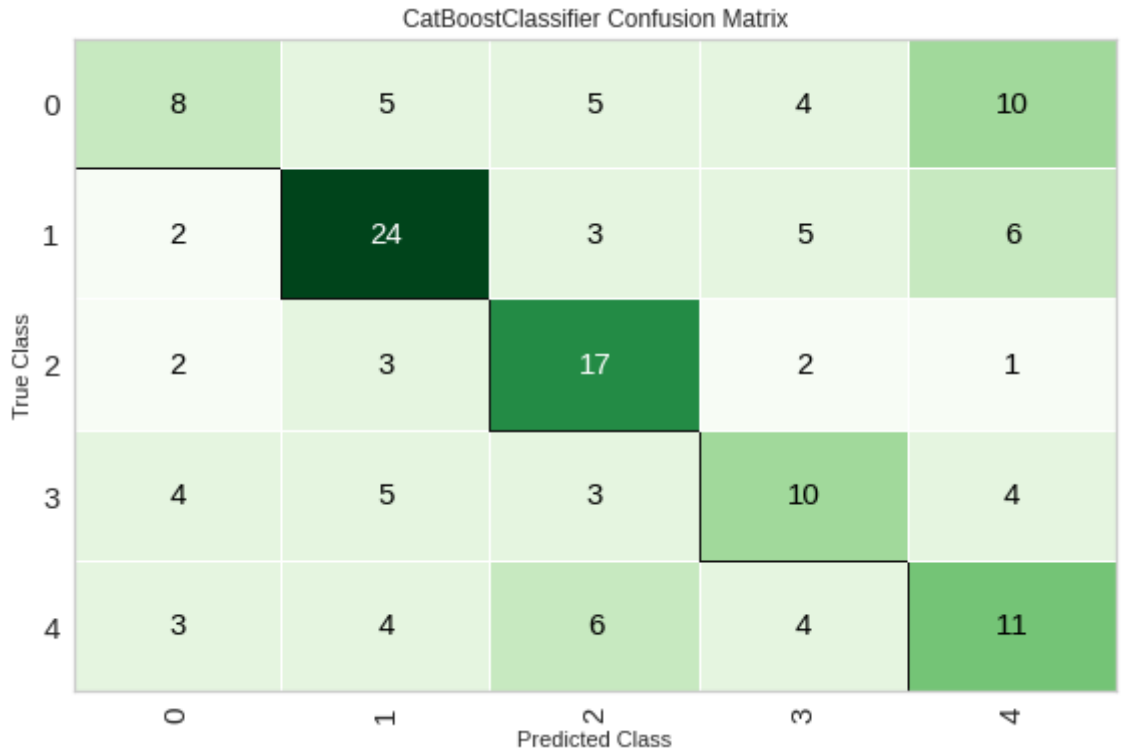
30	Normalize Method	None
31	Transformation	False
32	Transformation Method	None
33	PCA	False
34	PCA Method	None
35	PCA Components	None
36	Ignore Low Variance	False
37	Combine Rare Levels	False
38	Rare Level Threshold	None
39	Numeric Binning	False
40	Remove Outliers	False
41	Outliers Threshold	None
42	Remove Multicollinearity	False
43	Multicollinearity Threshold	None

```
compare_models()
```

```
estimator = create_model('catboost')
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.3714	0.7574	0.3810	0.3943	0.3692	0.2119	0.2152
1	0.4000	0.6614	0.3952	0.3781	0.3826	0.2469	0.2492
2	0.4571	0.7423	0.4512	0.4737	0.4402	0.3214	0.3292
3	0.4000	0.7467	0.4036	0.3929	0.3749	0.2500	0.2603
4	0.4000	0.7037	0.4083	0.4069	0.3883	0.2515	0.2568
5	0.4571	0.6868	0.4702	0.4481	0.4427	0.3228	0.3265
6	0.3143	0.6733	0.3190	0.3036	0.3038	0.1429	0.1439
7	0.3429	0.7096	0.3369	0.3690	0.3368	0.1735	0.1774
8	0.4286	0.6794	0.4357	0.4072	0.3981	0.2872	0.2957
9	0.3824	0.6645	0.3857	0.3833	0.3738	0.2281	0.2319
Mean	0.3954	0.7025	0.3987	0.3957	0.3810	0.2436	0.2486
SD	0.0434	0.0337	0.0447	0.0433	0.0397	0.0554	0.0569

```
plot_model(estimator, 'confusion_matrix')
```




```
plot_model(estimator, 'class_report')
```