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/1AnlNZGfc_wHchdUU4ojk8Skx6bLGhy-B/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')
training_df = training_df.drop(columns = 'Unnamed: 0')
print(training_df)
```

```
          Sublevel Name  pp_magtropy  pp_avg_magnitude   entropy
   0     Stelpaviricetes    63.395274         86.688259  1.367425
   1     Stelpaviricetes    51.680550         71.074139  1.375259
```

```
2     Stelpaviricetes    61.679557         84.457897  1.369301
3     Stelpaviricetes    54.650512         74.886937  1.370288
4     Stelpaviricetes    63.488241         86.883770  1.368502
..                ...          ...               ...       ...
295  Duplopiviricetes    27.813665         37.860965  1.361236
296  Duplopiviricetes    25.793086         35.661495  1.382599
297  Duplopiviricetes    37.391497         51.711052  1.382963
298  Duplopiviricetes    25.139549         34.708882  1.380649
299  Duplopiviricetes    31.456221         42.574051  1.353438

[300 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     Stelpaviricetes    63.395274
1     Stelpaviricetes    51.680550
```

```
2     Stelpaviricetes    61.679557
3     Stelpaviricetes    54.650512
4     Stelpaviricetes    63.488241
..                ...          ...
295  Duplopiviricetes    27.813665
296  Duplopiviricetes    25.793086
297  Duplopiviricetes    37.391497
298  Duplopiviricetes    25.139549
299  Duplopiviricetes    31.456221

[300 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	5303
1	Target	Sublevel Name
2	Target Type	Multiclass
3	Label Encoded	Duplopiviricetes: 0, Pisoniviricetes: 1, Stelp...
4	Original Data	(300, 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	(209, 1)
12	Transformed Test Set	(91, 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	2367
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

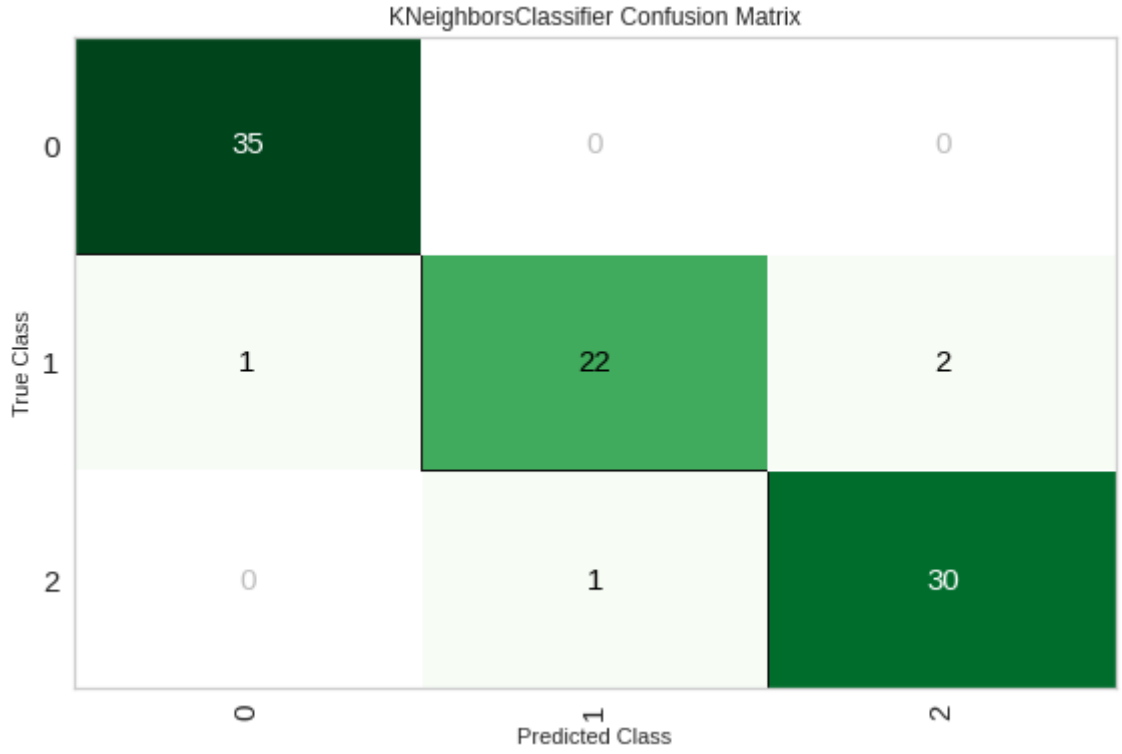
```
compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
knn	K Neighbors Classifier	0.8902	0.9581	0.8921	0.9018	0.8880	0.8349	0.8416	0.118
xgboost	Extreme Gradient Boosting	0.8900	0.9609	0.8911	0.9043	0.8900	0.8346	0.8412	0.448
dt	Decision Tree Classifier	0.8852	0.9138	0.8863	0.9009	0.8856	0.8275	0.8343	0.019
rf	Random Forest Classifier	0.8852	0.9576	0.8863	0.9009	0.8856	0.8275	0.8343	0.470
gbc	Gradient Boosting Classifier	0.8852	0.9553	0.8863	0.9009	0.8856	0.8275	0.8343	0.200
et	Extra Trees Classifier	0.8852	0.9495	0.8863	0.9021	0.8844	0.8273	0.8357	0.469
catboost	CatBoost Classifier	0.8852	0.9656	0.8863	0.9009	0.8856	0.8275	0.8343	0.796
lightgbm	Light Gradient Boosting Machine	0.8802	0.9552	0.8831	0.8935	0.8784	0.8203	0.8276	0.058
nb	Naive	0.7269	0.8427	0.7415	0.7988	0.6986	0.5932	0.6418	0.018

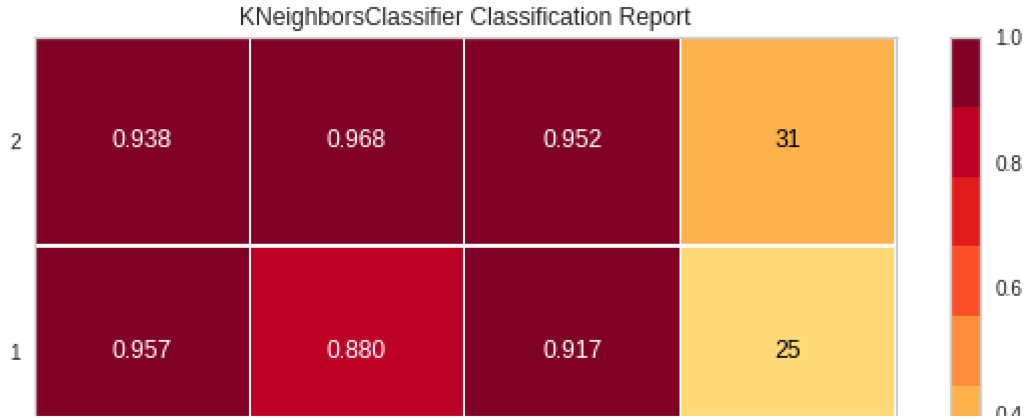
```
estimator = create_model('knn')
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.8095	0.9377	0.8016	0.8485	0.8013	0.7093	0.7328
1	0.7619	0.8874	0.7738	0.8095	0.7567	0.6441	0.6645
2	0.9048	0.9929	0.9167	0.9259	0.9039	0.8571	0.8690
3	0.9524	0.9528	0.9524	0.9577	0.9520	0.9278	0.9311
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
5	0.8095	0.9473	0.8095	0.8075	0.8056	0.7143	0.7167
6	0.9048	0.9473	0.9048	0.9048	0.9048	0.8571	0.8571
7	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

```
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))
```

```
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
100
```

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

```
[[  1]
 [100]]
```

Magnitude avg

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

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

```
          Sublevel Name  pp_avg_magnitude
0     Stelpaviricetes         86.688259
1     Stelpaviricetes         71.074139
2     Stelpaviricetes         84.457897
3     Stelpaviricetes         74.886937
4     Stelpaviricetes         86.883770
..                ...               ...
295  Duplopiviricetes         37.860965
296  Duplopiviricetes         35.661495
297  Duplopiviricetes         51.711052
298  Duplopiviricetes         34.708882
299  Duplopiviricetes         42.574051

[300 rows x 2 columns]
```

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

	Description	Value
0	session_id	4002
1	Target	Sublevel Name
2	Target Type	Multiclass
3	Label Encoded	Duplopiviricetes: 0, Pisoniviricetes: 1, Stelp...
4	Original Data	(300, 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	(209, 1)
12	Transformed Test Set	(91, 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	b126
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

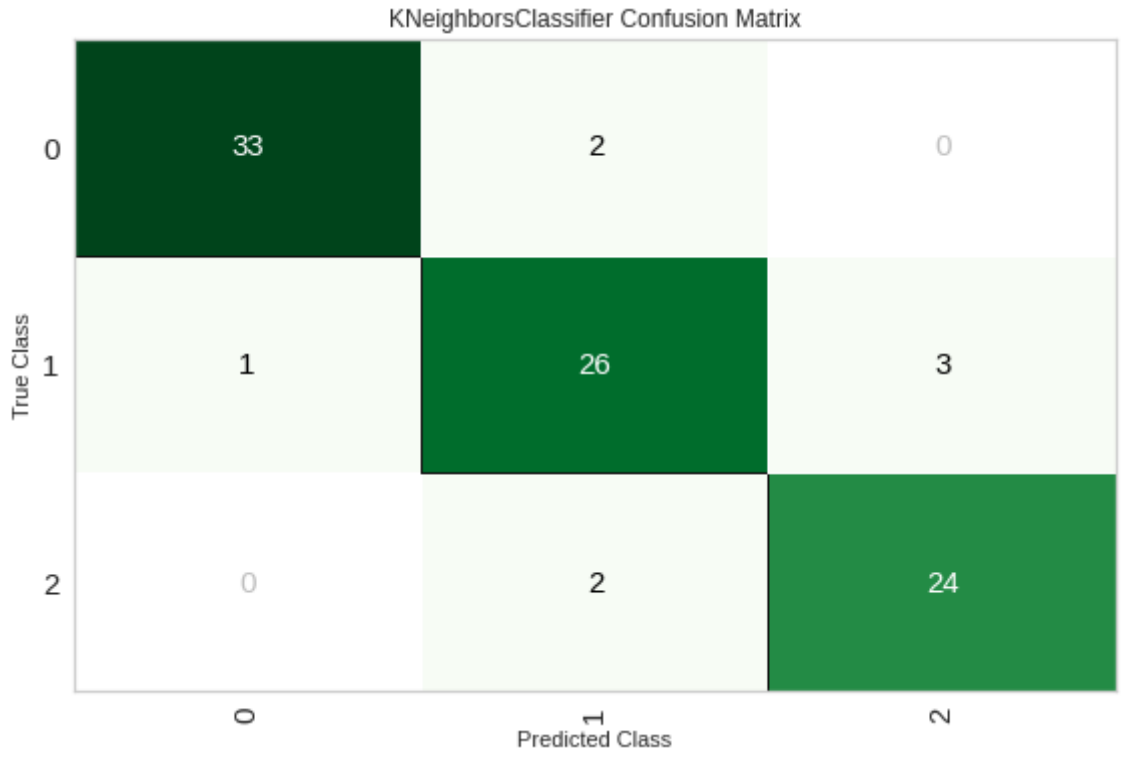
```
compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
knn	K Neighbors Classifier	0.9236	0.9553	0.9250	0.9359	0.9222	0.8853	0.8925	0.122
lightgbm	Light Gradient Boosting Machine	0.9045	0.9589	0.9077	0.9190	0.9026	0.8566	0.8648	0.053
rf	Random Forest Classifier	0.9043	0.9555	0.9063	0.9171	0.9037	0.8561	0.8627	0.475
et	Extra Trees Classifier	0.9043	0.9583	0.9063	0.9171	0.9037	0.8561	0.8627	0.472
dt	Decision Tree Classifier	0.8995	0.9241	0.9008	0.9124	0.8989	0.8488	0.8554	0.020
gbc	Gradient Boosting Classifier	0.8995	0.9586	0.9008	0.9124	0.8989	0.8488	0.8554	0.203
xgboost	Extreme Gradient Boosting	0.8995	0.9568	0.9008	0.9124	0.8989	0.8488	0.8554	0.392
catboost	CatBoost Classifier	0.8995	0.9635	0.9008	0.9124	0.8989	0.8488	0.8554	0.812
ada	Ada Boost	0.8617	0.9254	0.8599	0.8906	0.8509	0.7914	0.8083	0.095

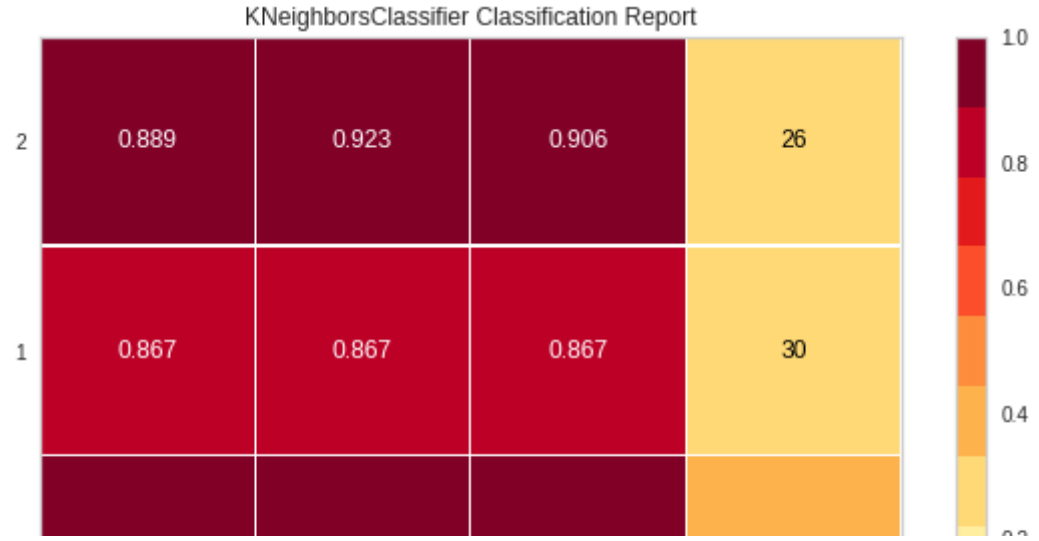
```
estimator = create_model('knn')
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
1	0.9048	0.9286	0.9048	0.9167	0.9000	0.8571	0.8660
2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
3	0.9048	0.9609	0.9048	0.9259	0.9028	0.8571	0.8690
4	0.9048	0.9184	0.9048	0.9259	0.9028	0.8571	0.8690
5	0.9524	0.9771	0.9524	0.9577	0.9520	0.9278	0.9311
6	0.8095	0.9103	0.8155	0.8143	0.8005	0.7143	0.7242
7	0.9524	0.9843	0.9524	0.9577	0.9520	0.9278	0.9311
8	0.8571	0.9218	0.8631	0.9048	0.8619	0.7872	0.8063
9	0.9500	0.9519	0.9524	0.9562	0.9497	0.9248	0.9283

```
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))
```

```
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
100
```

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

```
[[  1]
 [100]]
```

Entropy

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

```
          Sublevel Name   entropy
   0     Stelpaviricetes  1.367425
   1     Stelpaviricetes  1.375259
   2     Stelpaviricetes  1.369301
   3     Stelpaviricetes  1.370288
   4     Stelpaviricetes  1.368502
   ..                ...       ...
   295  Duplopiviricetes  1.361236
   296  Duplopiviricetes  1.382599
   297  Duplopiviricetes  1.382963
   298  Duplopiviricetes  1.380649
   299  Duplopiviricetes  1.353438

   [300 rows x 2 columns]
```

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

	Description	Value
0	session_id	3628
1	Target	Sublevel Name
2	Target Type	Multiclass
3	Label Encoded	Duplopiviricetes: 0, Pisoniviricetes: 1, Stelp...
4	Original Data	(300, 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	(209, 1)
12	Transformed Test Set	(91, 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	978a
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

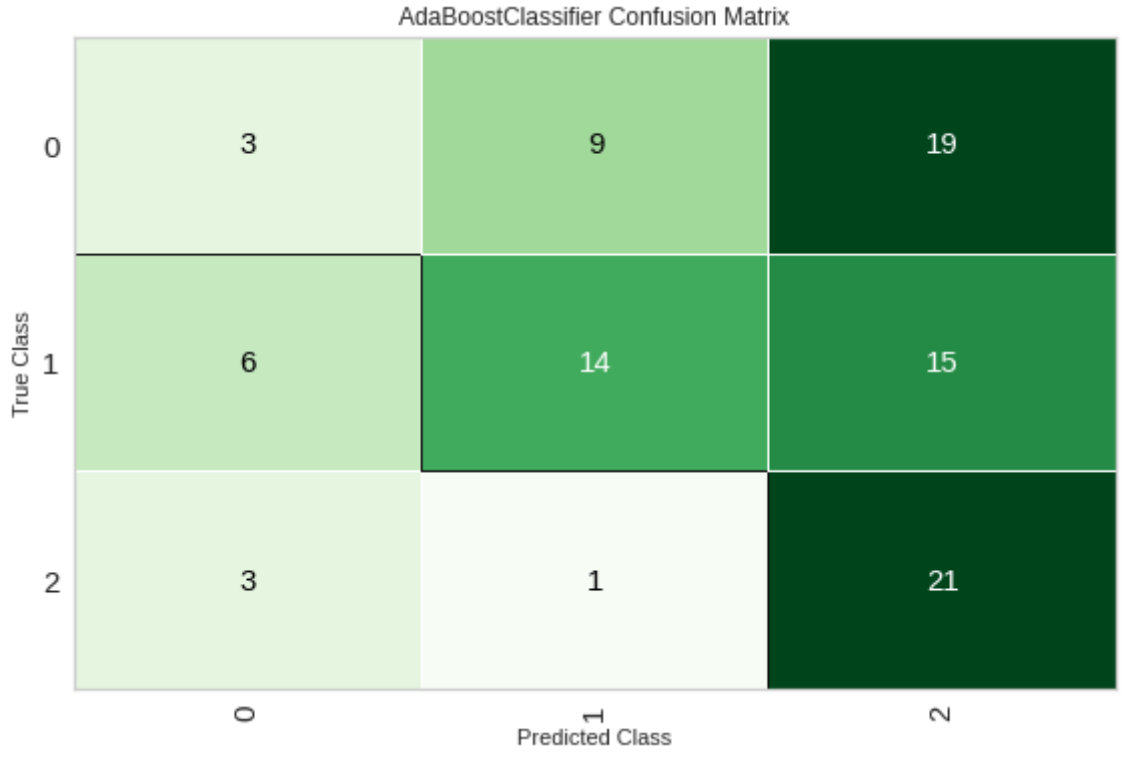
```
compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
ada	Ada Boost Classifier	0.4779	0.6445	0.4641	0.4544	0.4410	0.2016	0.2140	0.095
knn	K Neighbors Classifier	0.4736	0.6488	0.4754	0.4956	0.4635	0.2088	0.2153	0.122
lightgbm	Light Gradient Boosting Machine	0.4640	0.6639	0.4605	0.4598	0.4503	0.1915	0.1974	0.052
catboost	CatBoost Classifier	0.4112	0.6355	0.4127	0.4003	0.3981	0.1146	0.1168	0.784
gbc	Gradient Boosting Classifier	0.4062	0.6058	0.4044	0.3964	0.3926	0.1043	0.1064	0.203
xgboost	Extreme Gradient Boosting	0.3971	0.6283	0.3992	0.3733	0.3757	0.0964	0.0985	2.502
et	Extra Trees Classifier	0.3969	0.5805	0.3978	0.3706	0.3744	0.0956	0.0976	0.467
dt	Decision Tree Classifier	0.3921	0.5441	0.3937	0.3652	0.3690	0.0887	0.0909	0.021
rf	Random Forest	0.3921	0.6320	0.3937	0.3652	0.3690	0.0887	0.0909	0.472

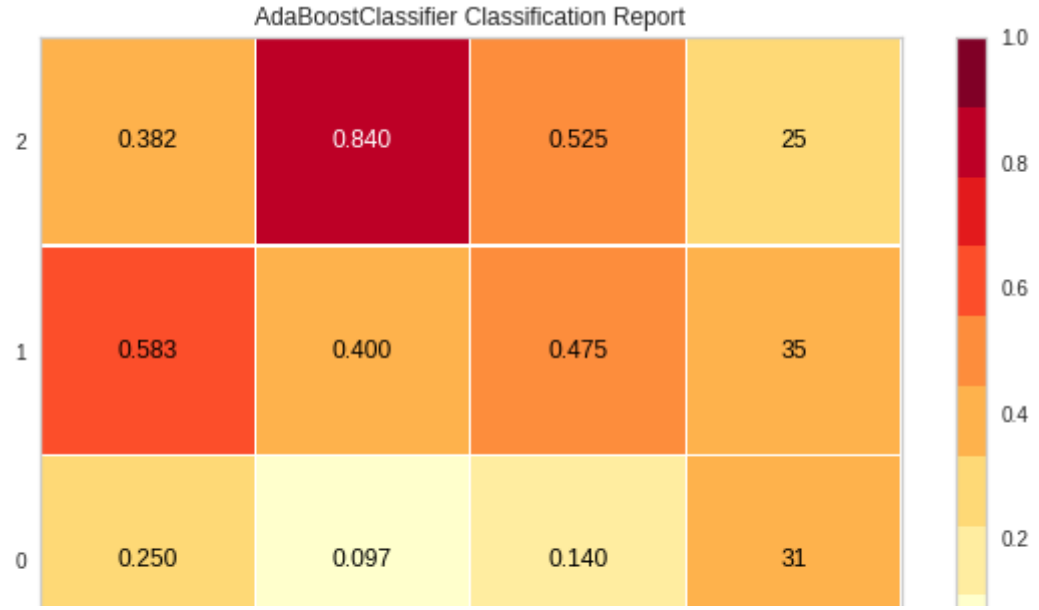
```
estimator = create_model('ada')
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.4286	0.6565	0.4286	0.4286	0.4125	0.1429	0.1474
1	0.3810	0.6395	0.3810	0.3694	0.3569	0.0714	0.0748
2	0.6667	0.6769	0.6667	0.6944	0.6354	0.5000	0.5401
3	0.6190	0.7551	0.6190	0.6389	0.6096	0.4286	0.4392
4	0.4762	0.6701	0.4762	0.4667	0.4428	0.2143	0.2296
5	0.3810	0.4770	0.3591	0.3751	0.3560	0.0387	0.0417
6	0.4286	0.6314	0.3929	0.3333	0.3714	0.1127	0.1199
7	0.5238	0.7788	0.5060	0.5147	0.4709	0.2683	0.2958
8	0.5238	0.6713	0.5060	0.4762	0.4762	0.2708	0.2875
9	0.3500	0.4884	0.3056	0.2464	0.2782	-0.0317	-0.0363
Mean	0.4779	0.6445	0.4641	0.4544	0.4410	0.2016	0.2140

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




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




```
entropy_testing_df = testing_df.drop(columns = ["pp_avg_magnitude", "pp_magtropy"])
print(entropy_testing_df)
```

```
     Sublevel Name   entropy
112   Embecovirus  1.339846
113   Embecovirus  1.359566
114   Embecovirus  1.337675
115   Embecovirus  1.339988
116   Embecovirus  1.339538
..            ...       ...
207   Embecovirus  1.367212
208   Embecovirus  1.339744
209   Embecovirus  1.340611
210   Embecovirus  1.303857
211   Embecovirus  1.339918

[100 rows x 2 columns]
```

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

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 2 0 0 0 2 0 0 0 0 0 0 0 0
 0 0 0 2 0 0 0 2 0 0 0 2 0 0 0 0 2 0 2 2 0 2 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 2 0 0 0 2 0 0 0 0]
100
```

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

```
[[ 0  2]
 [87 13]]
```