In [1]:
```
import os
import pandas as pd
import sklearn
import numpy as np
import eli5
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
import lime.lime_tabular
import itertools
from sklearn import model_selection
```

```
Using TensorFlow backend.
```

In [2]:
```
from sklearn import datasets
wine_data = datasets.load_wine()
df_wine = pd.DataFrame(wine_data.data,columns=wine_data.feature_names)
df_wine['target'] = pd.Series(wine_data.target)
```

In [3]:
```
from sklearn.model_selection import train_test_split
X = df_wine.drop(['target'], axis=1)
y = df_wine['target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

In [4]:
```
print(X)
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	\
0	14.23	1.71	2.43	15.6	127.0	2.80	
1	13.20	1.78	2.14	11.2	100.0	2.65	
2	13.16	2.36	2.67	18.6	101.0	2.80	
3	14.37	1.95	2.50	16.8	113.0	3.85	
4	13.24	2.59	2.87	21.0	118.0	2.80	
5	14.20	1.76	2.45	15.2	112.0	3.27	
6	14.39	1.87	2.45	14.6	96.0	2.50	
7	14.06	2.15	2.61	17.6	121.0	2.60	
8	14.83	1.64	2.17	14.0	97.0	2.80	
9	13.86	1.35	2.27	16.0	98.0	2.98	
10	14.10	2.16	2.30	18.0	105.0	2.95	
11	14.12	1.48	2.32	16.8	95.0	2.20	
12	13.75	1.73	2.41	16.0	89.0	2.60	
13	14.75	1.73	2.39	11.4	91.0	3.10	
14	14.38	1.87	2.38	12.0	102.0	3.30	
15	13.63	1.81	2.70	17.2	112.0	2.85	
16	14.30	1.92	2.72	20.0	120.0	2.80	
17	13.83	1.57	2.62	20.0	115.0	2.95	
18	14.19	1.59	2.48	16.5	108.0	3.30	
19	13.64	3.10	2.56	15.2	116.0	2.70	
20	14.06	1.63	2.28	16.0	126.0	3.00	
21	12.93	3.80	2.65	18.6	102.0	2.41	
22	13.71	1.86	2.36	16.6	101.0	2.61	
23	12.85	1.60	2.52	17.8	95.0	2.48	
24	13.50	1.81	2.61	20.0	96.0	2.53	
25	13.05	2.05	3.22	25.0	124.0	2.63	
26	13.39	1.77	2.62	16.1	93.0	2.85	
27	13.30	1.72	2.14	17.0	94.0	2.40	
28	13.87	1.90	2.80	19.4	107.0	2.95	
29	14.02	1.68	2.21	16.0	96.0	2.65	
..	...	...	...	...	...	...	
148	13.32	3.24	2.38	21.5	92.0	1.93	
149	13.08	3.90	2.36	21.5	113.0	1.41	
150	13.50	3.12	2.62	24.0	123.0	1.40	
151	12.79	2.67	2.48	22.0	112.0	1.48	
152	13.11	1.90	2.75	25.5	116.0	2.20	
153	13.23	3.30	2.28	18.5	98.0	1.80	
154	12.58	1.29	2.10	20.0	103.0	1.48	
155	13.17	5.19	2.32	22.0	93.0	1.74	
156	13.84	4.12	2.38	19.5	89.0	1.80	
157	12.45	3.03	2.64	27.0	97.0	1.90	
158	14.34	1.68	2.70	25.0	98.0	2.80	
159	13.48	1.67	2.64	22.5	89.0	2.60	
160	12.36	3.83	2.38	21.0	88.0	2.30	
161	13.69	3.26	2.54	20.0	107.0	1.83	
162	12.85	3.27	2.58	22.0	106.0	1.65	
163	12.96	3.45	2.35	18.5	106.0	1.39	
164	13.78	2.76	2.30	22.0	90.0	1.35	
165	13.73	4.36	2.26	22.5	88.0	1.28	
166	13.45	3.70	2.60	23.0	111.0	1.70	
167	12.82	3.37	2.30	19.5	88.0	1.48	

168	13.58	2.58	2.69	24.5	105.0	1.55
169	13.40	4.60	2.86	25.0	112.0	1.98
170	12.20	3.03	2.32	19.0	96.0	1.25
171	12.77	2.39	2.28	19.5	86.0	1.39
172	14.16	2.51	2.48	20.0	91.0	1.68
173	13.71	5.65	2.45	20.5	95.0	1.68
174	13.40	3.91	2.48	23.0	102.0	1.80
175	13.27	4.28	2.26	20.0	120.0	1.59
176	13.17	2.59	2.37	20.0	120.0	1.65
177	14.13	4.10	2.74	24.5	96.0	2.05

	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	\
0	3.06	0.28	2.29	5.640000	1.04	
1	2.76	0.26	1.28	4.380000	1.05	
2	3.24	0.30	2.81	5.680000	1.03	
3	3.49	0.24	2.18	7.800000	0.86	
4	2.69	0.39	1.82	4.320000	1.04	
5	3.39	0.34	1.97	6.750000	1.05	
6	2.52	0.30	1.98	5.250000	1.02	
7	2.51	0.31	1.25	5.050000	1.06	
8	2.98	0.29	1.98	5.200000	1.08	
9	3.15	0.22	1.85	7.220000	1.01	
10	3.32	0.22	2.38	5.750000	1.25	
11	2.43	0.26	1.57	5.000000	1.17	
12	2.76	0.29	1.81	5.600000	1.15	
13	3.69	0.43	2.81	5.400000	1.25	
14	3.64	0.29	2.96	7.500000	1.20	
15	2.91	0.30	1.46	7.300000	1.28	
16	3.14	0.33	1.97	6.200000	1.07	
17	3.40	0.40	1.72	6.600000	1.13	
18	3.93	0.32	1.86	8.700000	1.23	
19	3.03	0.17	1.66	5.100000	0.96	
20	3.17	0.24	2.10	5.650000	1.09	
21	2.41	0.25	1.98	4.500000	1.03	
22	2.88	0.27	1.69	3.800000	1.11	
23	2.37	0.26	1.46	3.930000	1.09	
24	2.61	0.28	1.66	3.520000	1.12	
25	2.68	0.47	1.92	3.580000	1.13	
26	2.94	0.34	1.45	4.800000	0.92	
27	2.19	0.27	1.35	3.950000	1.02	
28	2.97	0.37	1.76	4.500000	1.25	
29	2.33	0.26	1.98	4.700000	1.04	
..	...	...	...	...	...	
148	0.76	0.45	1.25	8.420000	0.55	
149	1.39	0.34	1.14	9.400000	0.57	
150	1.57	0.22	1.25	8.600000	0.59	
151	1.36	0.24	1.26	10.800000	0.48	
152	1.28	0.26	1.56	7.100000	0.61	
153	0.83	0.61	1.87	10.520000	0.56	
154	0.58	0.53	1.40	7.600000	0.58	
155	0.63	0.61	1.55	7.900000	0.60	
156	0.83	0.48	1.56	9.010000	0.57	
157	0.58	0.63	1.14	7.500000	0.67	

158	1.31	0.53	2.70	13.000000	0.57
159	1.10	0.52	2.29	11.750000	0.57
160	0.92	0.50	1.04	7.650000	0.56
161	0.56	0.50	0.80	5.880000	0.96
162	0.60	0.60	0.96	5.580000	0.87
163	0.70	0.40	0.94	5.280000	0.68
164	0.68	0.41	1.03	9.580000	0.70
165	0.47	0.52	1.15	6.620000	0.78
166	0.92	0.43	1.46	10.680000	0.85
167	0.66	0.40	0.97	10.260000	0.72
168	0.84	0.39	1.54	8.660000	0.74
169	0.96	0.27	1.11	8.500000	0.67
170	0.49	0.40	0.73	5.500000	0.66
171	0.51	0.48	0.64	9.899999	0.57
172	0.70	0.44	1.24	9.700000	0.62
173	0.61	0.52	1.06	7.700000	0.64
174	0.75	0.43	1.41	7.300000	0.70
175	0.69	0.43	1.35	10.200000	0.59
176	0.68	0.53	1.46	9.300000	0.60
177	0.76	0.56	1.35	9.200000	0.61

	od280/od315_of_diluted_wines	proline
0	3.92	1065.0
1	3.40	1050.0
2	3.17	1185.0
3	3.45	1480.0
4	2.93	735.0
5	2.85	1450.0
6	3.58	1290.0
7	3.58	1295.0
8	2.85	1045.0
9	3.55	1045.0
10	3.17	1510.0
11	2.82	1280.0
12	2.90	1320.0
13	2.73	1150.0
14	3.00	1547.0
15	2.88	1310.0
16	2.65	1280.0
17	2.57	1130.0
18	2.82	1680.0
19	3.36	845.0
20	3.71	780.0
21	3.52	770.0
22	4.00	1035.0
23	3.63	1015.0
24	3.82	845.0
25	3.20	830.0
26	3.22	1195.0
27	2.77	1285.0
28	3.40	915.0
29	3.59	1035.0
..	...	...

148	1.62	650.0
149	1.33	550.0
150	1.30	500.0
151	1.47	480.0
152	1.33	425.0
153	1.51	675.0
154	1.55	640.0
155	1.48	725.0
156	1.64	480.0
157	1.73	880.0
158	1.96	660.0
159	1.78	620.0
160	1.58	520.0
161	1.82	680.0
162	2.11	570.0
163	1.75	675.0
164	1.68	615.0
165	1.75	520.0
166	1.56	695.0
167	1.75	685.0
168	1.80	750.0
169	1.92	630.0
170	1.83	510.0
171	1.63	470.0
172	1.71	660.0
173	1.74	740.0
174	1.56	750.0
175	1.56	835.0
176	1.62	840.0
177	1.60	560.0

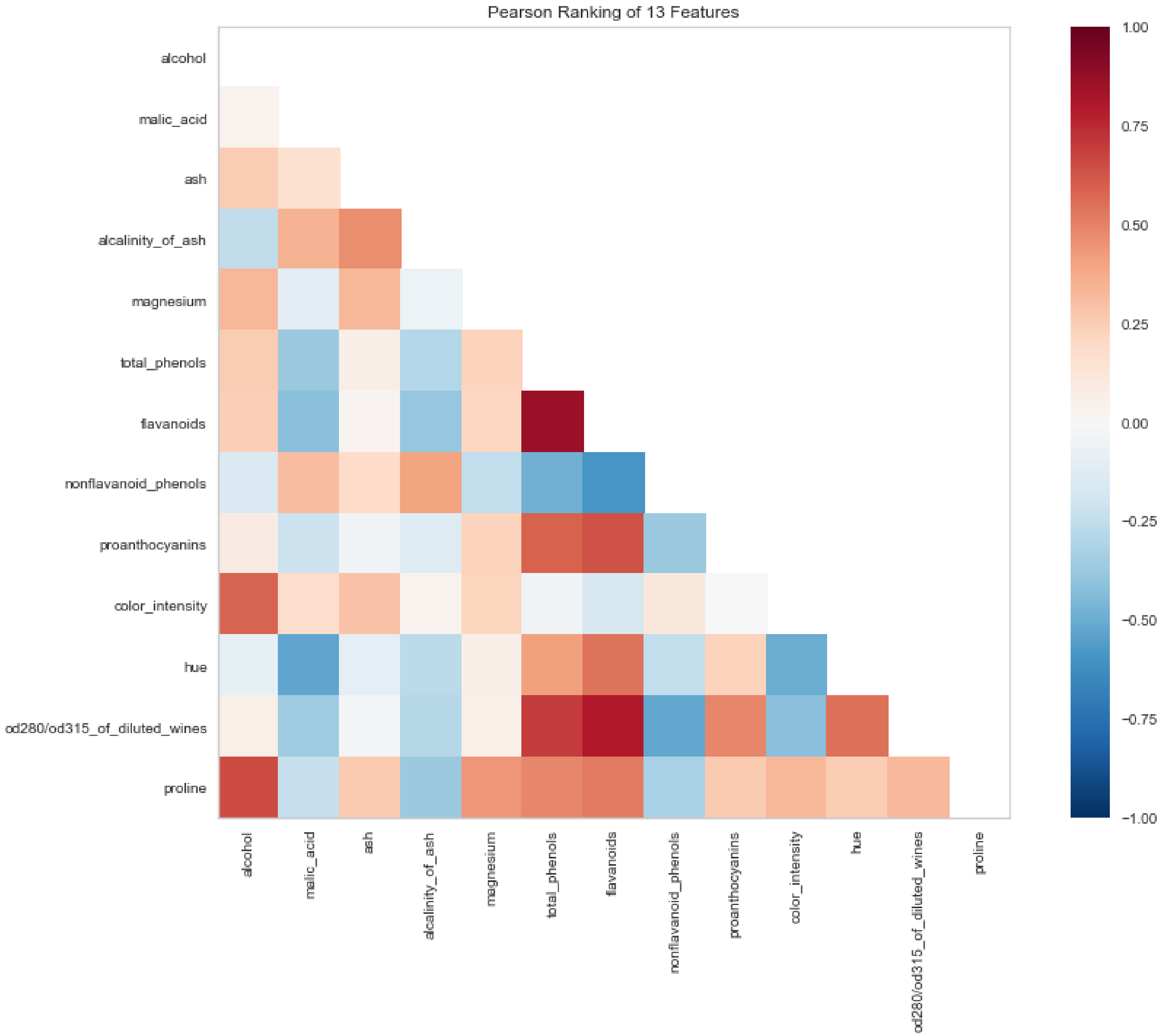
[178 rows x 13 columns]

In [5]:
```
print(y)
```

```
0      0
1      0
2      0
3      0
4      0
5      0
6      0
7      0
8      0
9      0
10     0
11     0
12     0
13     0
14     0
15     0
16     0
17     0
18     0
19     0
20     0
21     0
22     0
23     0
24     0
25     0
26     0
27     0
28     0
29     0
      ..
148    2
149    2
150    2
151    2
152    2
153    2
154    2
155    2
156    2
157    2
158    2
159    2
160    2
161    2
162    2
163    2
164    2
165    2
166    2
167    2
168    2
```

```
169    2
170    2
171    2
172    2
173    2
174    2
175    2
176    2
177    2
Name: target, Length: 178, dtype: int32
```

In [6]:
```
from yellowbrick.features import Rank2D
visualizer = Rank2D(algorithm="pearson",  size=(1080, 720))
visualizer.fit_transform(X_train)
visualizer.poof()
```

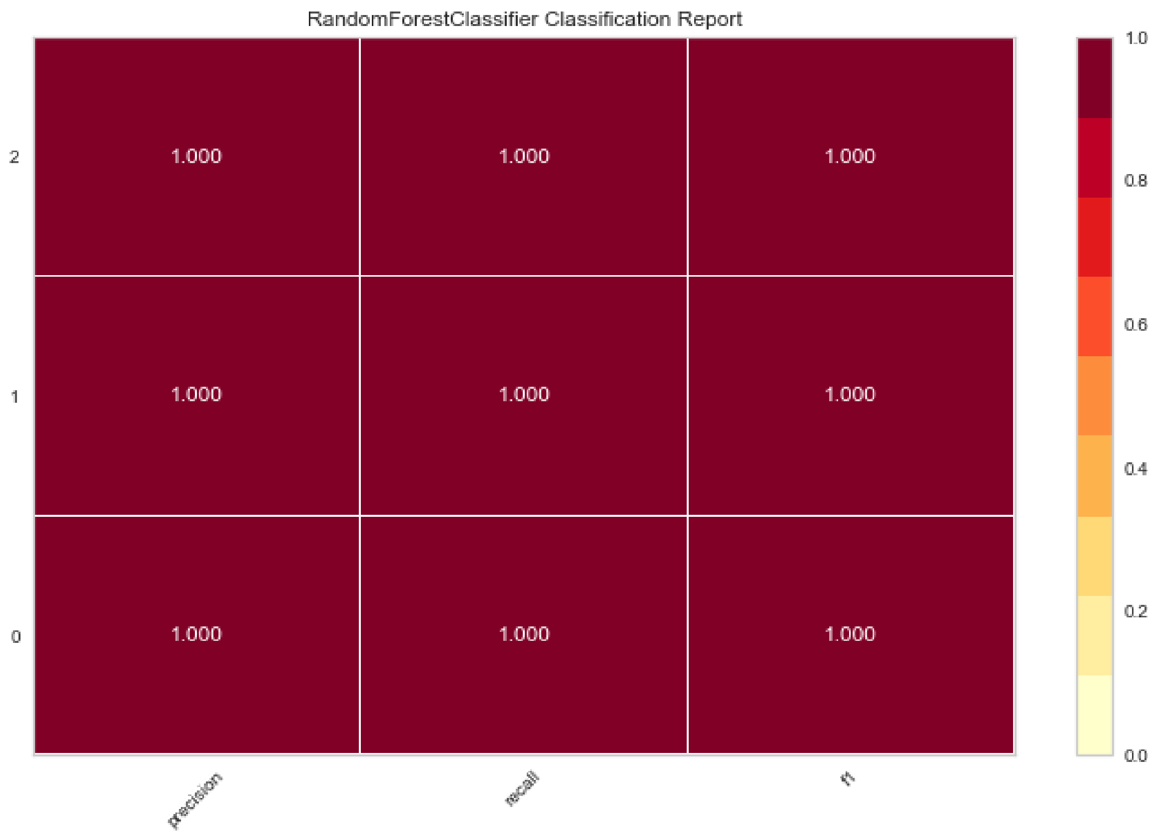


Out[6]: <matplotlib.axes._subplots.AxesSubplot at 0x4ec21b8240>

In [7]:

```
from yellowbrick.classifier import ClassificationReport
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
visualizer = ClassificationReport(model, size=(720, 480))
visualizer.fit(X_train, y_train)
visualizer.score(X_test, y_test)
visualizer.poof()
```

```
C:\Users\nilesh\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: FutureWarning: The default value of n_estimators wil
l change from 10 in version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```



Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x4ec34bea58>

In [8]:
```
import eli5
eli5.show_weights(model, feature_names = X.columns.tolist())
```

Out[8]:

Weight	Feature
0.1843 ± 0.3464	proline
0.1586 ± 0.3622	alcohol
0.1436 ± 0.3373	flavanoids
0.1007 ± 0.2941	color_intensity
0.0843 ± 0.1962	hue
0.0747 ± 0.2275	od280/od315_of_diluted_wines
0.0659 ± 0.1980	alcalinity_of_ash
0.0460 ± 0.1379	total_phenols
0.0337 ± 0.1598	nonflavanoid_phenols
0.0334 ± 0.1417	magnesium
0.0305 ± 0.1326	malic_acid
0.0257 ± 0.0745	proanthocyanins
0.0187 ± 0.0491	ash

In [9]:
```
from eli5 import show_prediction
show_prediction(model, X_train.iloc[1], feature_names = X.columns.tolist(),
                show_feature_values=True)
```

Out[9]:

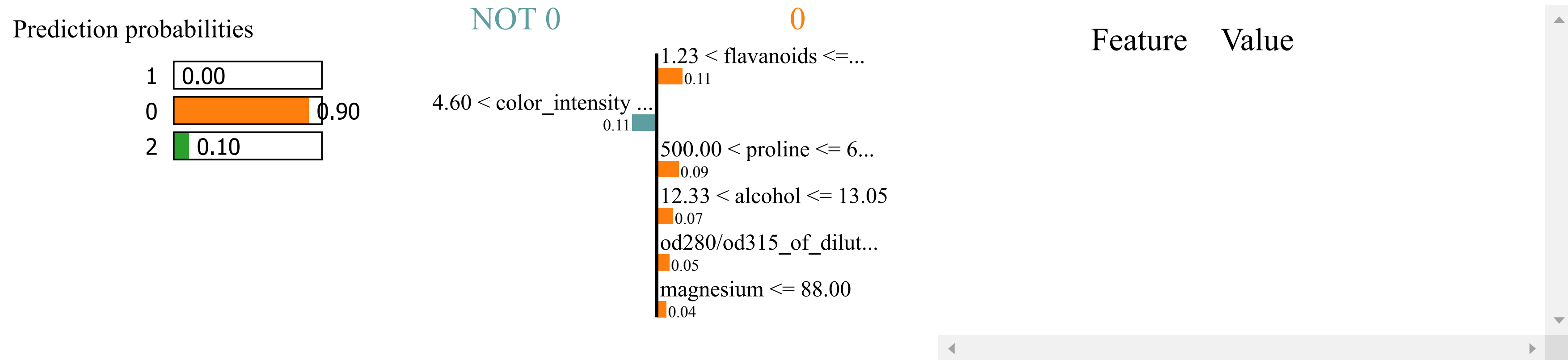
y=0 (probability **0.900**) top features			**y=1** (probability **0.000**) top features			**y=2** (probability **0.100**) top features		
Contribution?	**Feature**	**Value**	**Contribution?**	**Feature**	**Value**	**Contribution?**	**Feature**	**Val**
+0.335	<BIAS>	1.000	+0.398	<BIAS>	1.000	+0.267	<BIAS>	1.0
+0.220	proline	845.000	+0.031	alcalinity_of_ash	20.000	+0.052	alcalinity_of_ash	20.0
+0.149	alcohol	13.500	+0.012	color_intensity	3.520	+0.025	alcohol	13.5
+0.095	flavanoids	2.610	+0.012	od280/od315_of_diluted_wines	3.820	-0.000	magnesium	96.0
+0.058	ash	2.610	+0.009	malic_acid	1.810	-0.021	malic_acid	1.8
+0.030	hue	1.120	+0.007	total_phenols	2.530	-0.023	total_phenols	2.5
+0.029	magnesium	96.000	+0.003	hue	1.120	-0.027	od280/od315_of_diluted_wines	3.8
+0.019	color_intensity	3.520	-0.004	proanthocyanins	1.660	-0.032	color_intensity	3.5
+0.016	total_phenols	2.530	-0.029	flavanoids	2.610	-0.033	hue	1.1
+0.015	od280/od315_of_diluted_wines	3.820	-0.029	magnesium	96.000	-0.041	proline	845.0
+0.012	malic_acid	1.810	-0.058	ash	2.610	-0.066	flavanoids	2.6
+0.004	proanthocyanins	1.660	-0.173	alcohol	13.500			
-0.083	alcalinity_of_ash	20.000	-0.179	proline	845.000			

In [10]:
```
import lime.lime_tabular
```

In [11]:
```
explainer = lime.lime_tabular.LimeTabularExplainer(X_train.values,
                 feature_names=X_train.columns.values.tolist(),
                 class_names=y_train.unique())
```

In [12]:
```
predict_fn = lambda x: model.predict_proba(x).astype(float)
```

In [13]:
```
exp = explainer.explain_instance(X_test.values[0], predict_fn, num_features=6)
exp.show_in_notebook(show_all=False)
```



In [14]:
```
from mlxtend.plotting import plot_decision_regions
from mlxtend.classifier import EnsembleVoteClassifier
import matplotlib.gridspec as gridspec
import itertools
from sklearn import model_selection
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
```

In [15]:
```
import matplotlib.gridspec as gridspec
```

In [16]:
```
from mlxtend.plotting import plot_decision_regions
from mlxtend.classifier import EnsembleVoteClassifier
```

In [17]:
```
from mlxtend.plotting import plot_decision_regions
from mlxtend.classifier import EnsembleVoteClassifier
import matplotlib.gridspec as gridspec
import itertools
from sklearn import model_selection
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
```

In [18]:
```
X_train = X_train[['proline', 'color_intensity']].values
y_train = y_train.values
```

In [24]:
```
clf1 = LogisticRegression(random_state=1)
clf2 = RandomForestClassifier(random_state=1)
clf3 = GaussianNB()
eclf = EnsembleVoteClassifier(clfs=[clf1, clf2, clf3], weights=[1,1,1])
value=1.5
width=1.5
gs = gridspec.GridSpec(2,2)
fig = plt.figure(figsize=(12,10))
labels = ['LOGISTIC REGRESSION', 'RANDOM FOREST', 'NAIVE BAYES', 'ENSEMBLES']
for clf, lab, grd in zip([clf1, clf2, clf3, eclf],
                         labels,
                         itertools.product([0, 1], repeat=2)):

    clf.fit(X_train, y_train)
    ax = plt.subplot(gs[grd[0], grd[1]])
    fig = plot_decision_regions(X=X_train, y=y_train, clf=clf)
    plt.title(lab)
```

```
C:\Users\nilesh\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning: Default solver will be changed
to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
C:\Users\nilesh\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:469: FutureWarning: Default multi_class will be cha
nged to 'auto' in 0.22. Specify the multi_class option to silence this warning.
  "this warning.", FutureWarning)
C:\Users\nilesh\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: FutureWarning: The default value of n_estimators wil
l change from 10 in version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\nilesh\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning: Default solver will be changed
to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
C:\Users\nilesh\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:469: FutureWarning: Default multi_class will be cha
nged to 'auto' in 0.22. Specify the multi_class option to silence this warning.
  "this warning.", FutureWarning)
```