Peeking into blackbox models

1. About myself and Go-jek
2. Interpreting Tabular (Gradient Boosting) Binary Classification
3. Interpreting Text/Document (Naive Bayes) Multi-Classification

Shameless self-promo

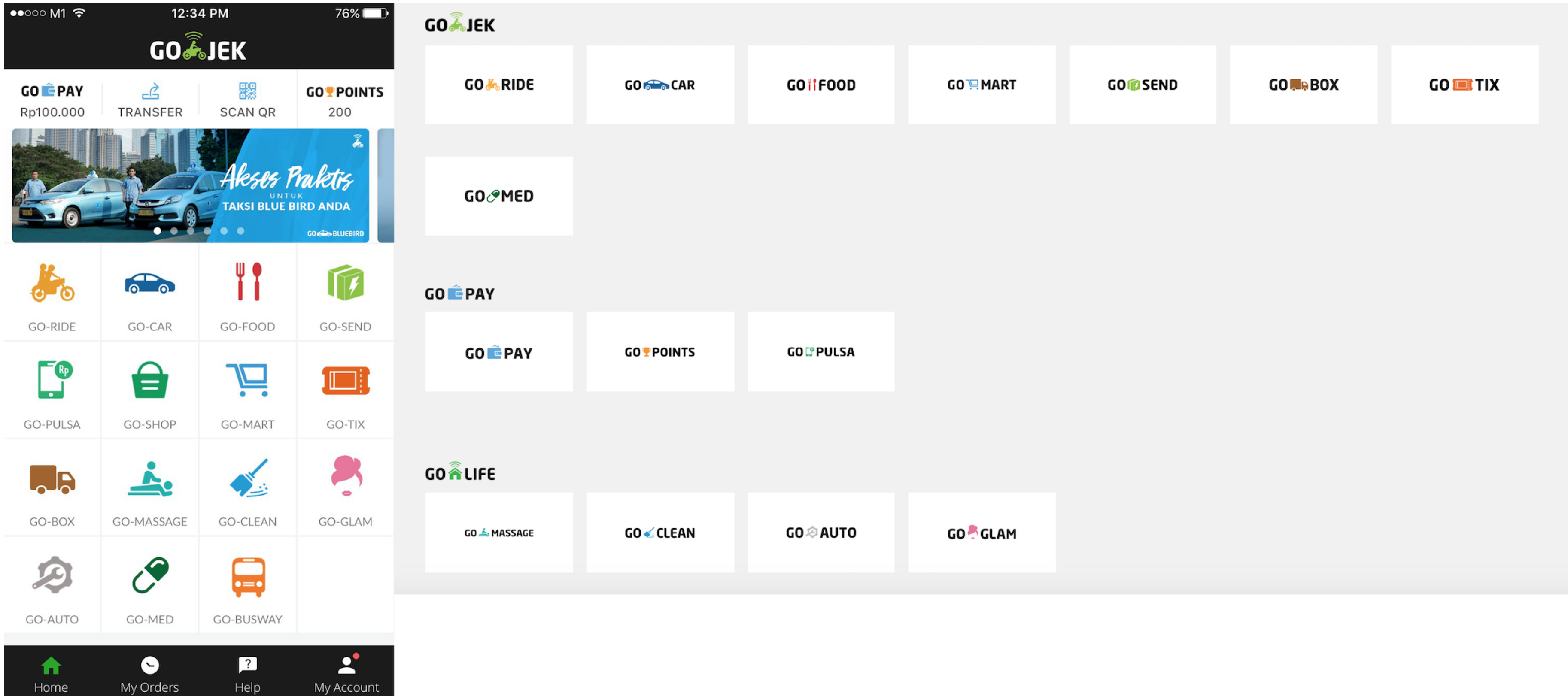
Zane Lim

- Senior data-scientist at Go-jek
- Lead and mentor a sub-team of data-scientists
- Mentor in Udacity's AI nanodegree
- Avid participant in Kaggle / data science hackathons

zyuanlim@gmail.com (zyuanlim@gmail.com) / zane.lim@go-jek.com (zane.lim@go-jek.com)

- https://www.linkedin.com/in/limzane/ (https://www.linkedin.com/in/limzane/)
- https://www.kaggle.com/zanelim (https://www.kaggle.com/zanelim)

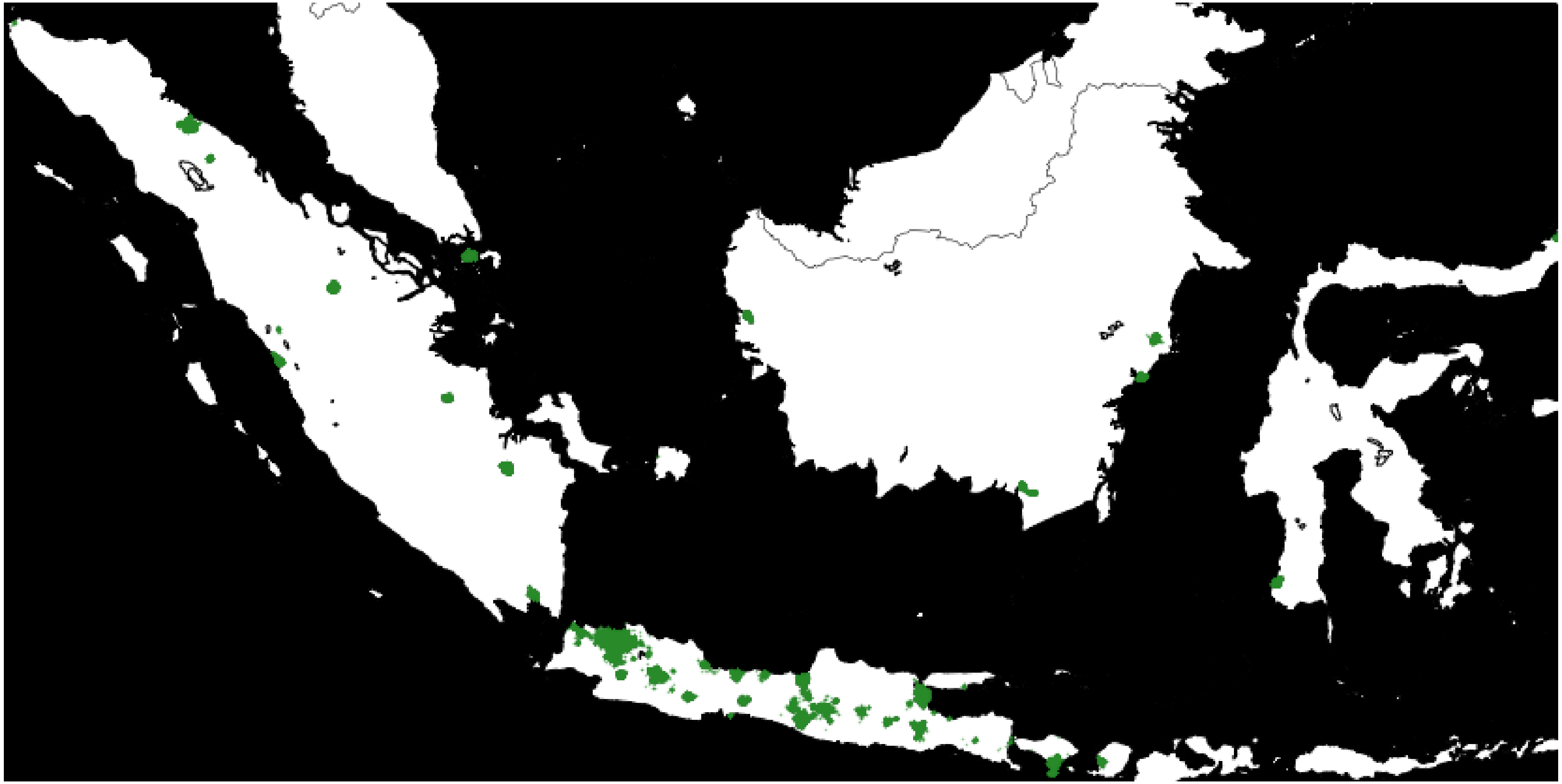
Go-Jek



- On-demand mobile platform that provides a variety of complete services
  - transportation (300,000 *ojek* and car driver partners)
  - logistics / food delivery (100,000 food vendors)
  - payment
  - other on-demand services (masseurs, cleaners etc.)

- As of July 2017, downloaded more than 44 million times
- Operates in 50 cities in Indonesia- Jakarta, Bandung, Surabaya, Bali, Makassar, Medan, Palembang, Semarang, Yogyakarta, Balikpapan, Malang, Solo, Manado, Samarinda,

Batam, Sidoarjo, Gresik, Pekanbaru, Jambi, Sukabumi, Bandar Lampung, Padang, Pontianak, Banjarmasin, Mataram etc.



Drank

- Webservice that serves real-time driver rankings
- Based on predictions by machine learning model eg. *probability of completion*
- Rank drivers who are available in the vicinity of customer whenever a booking is requested
- Assign the job to the top ranked driver

Interpretation, Transparency and Fairness Evaluation

1. Driver's feedback- monitor drivers' complaints about not getting jobs
2. Promote transparency- explain and provide suggestions to drivers
3. Fairness / inequality- Gini coefficient of drivers income and jobs received
4. Measure average ranking / predicted probability of each driver
5. Model interpretability- Visualizing model, Partial Dependence Plot (PDP), Local Interpretable Model-Agnostic Explanations (LIME)

Interpreting Gradient Boosting Model

1. Direct Visualization
2. Features Importance
3. Partial Dependence Plot (PDP)
4. Local Interpretable Model-Agnostic Explanations (LIME)

In [234]:

```
import pandas as pd
import numpy as np
import lime
from lime import lime_tabular
import xgboost as xgb
import matplotlib.pyplot as plt
%matplotlib inline
import sklearn
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.datasets import load_wine
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
import plotly.graph_objs as go
init_notebook_mode(connected=True)
```

In [37]:

```
print 'pandas version: {}'.format(pd.__version__)
print 'numpy version: {}'.format(np.__version__)
print 'xgboost version: {}'.format(xgb.__version__)
print 'sklearn version: {}'.format(sklearn.__version__)
```

```
pandas version: 0.20.3
numpy version: 1.13.1
xgboost version: 0.6
sklearn version: 0.19.0
```

In [127]:

```
data = load_wine()
```

In [129]:
```
data['data'] = data.data[(data.target != 0)]
data['target'] = data.target[(data.target != 0)]-1
data['target_names'] = ['class_1', 'class_2']
```

In [8]:
```
print data.DESCR
```

```
Wine Data Database
====================

Notes
-----
Data Set Characteristics:
    :Number of Instances: 178 (50 in each of three classes)
    :Number of Attributes: 13 numeric, predictive attributes and the class
    :Attribute Information:
                - 1) Alcohol
                - 2) Malic acid
                - 3) Ash
                - 4) Alcalinity of ash
                - 5) Magnesium
                - 6) Total phenols
                - 7) Flavanoids
                - 8) Nonflavanoid phenols
                - 9) Proanthocyanins
                - 10)Color intensity
                - 11)Hue
                - 12)OD280/OD315 of diluted wines
                - 13)Proline
                - class:
                - class_0
                - class_1
                - class_2
```

:Summary Statistics:

	Min	Max	Mean	SD
Alcohol:	11.0	14.8	13.0	0.8
Malic Acid:	0.74	5.80	2.34	1.12
Ash:	1.36	3.23	2.36	0.27
Alcalinity of Ash:	10.6	30.0	19.5	3.3
Magnesium:	70.0	162.0	99.7	14.3
Total Phenols:	0.98	3.88	2.29	0.63
Flavanoids:	0.34	5.08	2.03	1.00
Nonflavanoid Phenols:	0.13	0.66	0.36	0.12
Proanthocyanins:	0.41	3.58	1.59	0.57
Colour Intensity:	1.3	13.0	5.1	2.3
Hue:	0.48	1.71	0.96	0.23
OD280/OD315 of diluted wines:	1.27	4.00	2.61	0.71
Proline:	278	1680	746	315

:Missing Attribute Values: None
:Class Distribution: class_0 (59), class_1 (71), class_2 (48)
:Creator: R.A. Fisher
:Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
:Date: July, 1988

This is a copy of UCI ML Wine recognition datasets.

https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data

The data is the results of a chemical analysis of wines grown in the same region in Italy by three different cultivators. There are thirteen different measurements taken for different constituents found in the three types of wine.

Original Owners:

Forina, M. et al, PARVUS -
An Extendible Package for Data Exploration, Classification and Correlation.
Institute of Pharmaceutical and Food Analysis and Technologies,
Via Brigata Salerno, 16147 Genoa, Italy.

Citation:

Lichman, M. (2013). UCI Machine Learning Repository
[http://archive.ics.uci.edu/ml]. Irvine, CA: University of California,
School of Information and Computer Science.

References
----
(1)
S. Aeberhard, D. Coomans and O. de Vel,
Comparison of Classifiers in High Dimensional Settings,
Tech. Rep. no. 92-02, (1992), Dept. of Computer Science and Dept. of
Mathematics and Statistics, James Cook University of North Queensland.
(Also submitted to Technometrics).

```
The data was used with many others for comparing various
classifiers. The classes are separable, though only RDA
has achieved 100% correct classification.
(RDA : 100%, QDA 99.4%, LDA 98.9%, 1NN 96.1% (z-transformed data))
(All results using the leave-one-out technique)

(2)
S. Aeberhard, D. Coomans and O. de Vel,
"THE CLASSIFICATION PERFORMANCE OF RDA"
Tech. Rep. no. 92-01, (1992), Dept. of Computer Science and Dept. of
Mathematics and Statistics, James Cook University of North Queensland.
(Also submitted to Journal of Chemometrics).
```

In [132]:

```
data.target_names
```

Out[132]:

```
['class_1', 'class_2']
```

In [133]:
```
data.feature_names
```

Out[133]:
```
['alcohol',
 'malic_acid',
 'ash',
 'alcalinity_of_ash',
 'magnesium',
 'total_phenols',
 'flavanoids',
 'nonflavanoid_phenols',
 'proanthocyanins',
 'color_intensity',
 'hue',
 'od280/od315_of_diluted_wines',
 'proline']
```

In [134]:
```
data.data.shape
```

Out[134]:
```
(119, 13)
```

In [135]:
```
pd.Series(data.target).value_counts()
```

Out[135]:
```
0    71
1    48
dtype: int64
```

In [212]:
```
X_train, X_test, y_train, y_test = train_test_split(data.data, data.target, test_size=0.3,
random_state=0)
```

In [138]:
```
print X_train.shape, X_test.shape
```

```
(83, 13) (36, 13)
```

In [140]:
```
dtrain = xgb.DMatrix(X_train, y_train, feature_names=data.feature_names)
```

In [141]:
```
params = {'learning_rate': 0.1, 'max_depth': 4, 'subsample': 1.0, 'colsample_bytree': 0.8,
          'objective': 'binary:logistic', 'eval_metric': 'logloss'}

cv = xgb.cv(params, dtrain, 200, nfold=5, stratified=True, early_stopping_rounds=20,
seed=0)
```

In [142]:
```
cv.iloc[-1]
```

Out[142]:
```
test-logloss-mean     0.072455
test-logloss-std      0.020414
train-logloss-mean    0.031961
train-logloss-std     0.000370
Name: 61, dtype: float64
```

In [143]:

```
params['n_estimators'] = 62
params['n_jobs'] = 5
params['random_state'] = 0
xgb_clf = xgb.XGBClassifier(**params)
xgb_clf.fit(X_train, y_train)
```

Out[143]:

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
       colsample_bytree=0.8, eval_metric='logloss', gamma=0,
       learning_rate=0.1, max_delta_step=0, max_depth=4,
       min_child_weight=1, missing=None, n_estimators=62, n_jobs=5,
       nthread=None, objective='binary:logistic', random_state=0,
       reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
       silent=True, subsample=1.0)
```

In [144]:

```
y_probs = xgb_clf.predict_proba(X_test)
y_preds = xgb_clf.predict(X_test)
```

In [145]:

```
print classification_report(y_test, y_preds, target_names=data.target_names)
```

```
             precision    recall  f1-score   support

    class_1       1.00      0.91      0.95        23
    class_2       0.87      1.00      0.93        13

avg / total       0.95      0.94      0.95        36
```

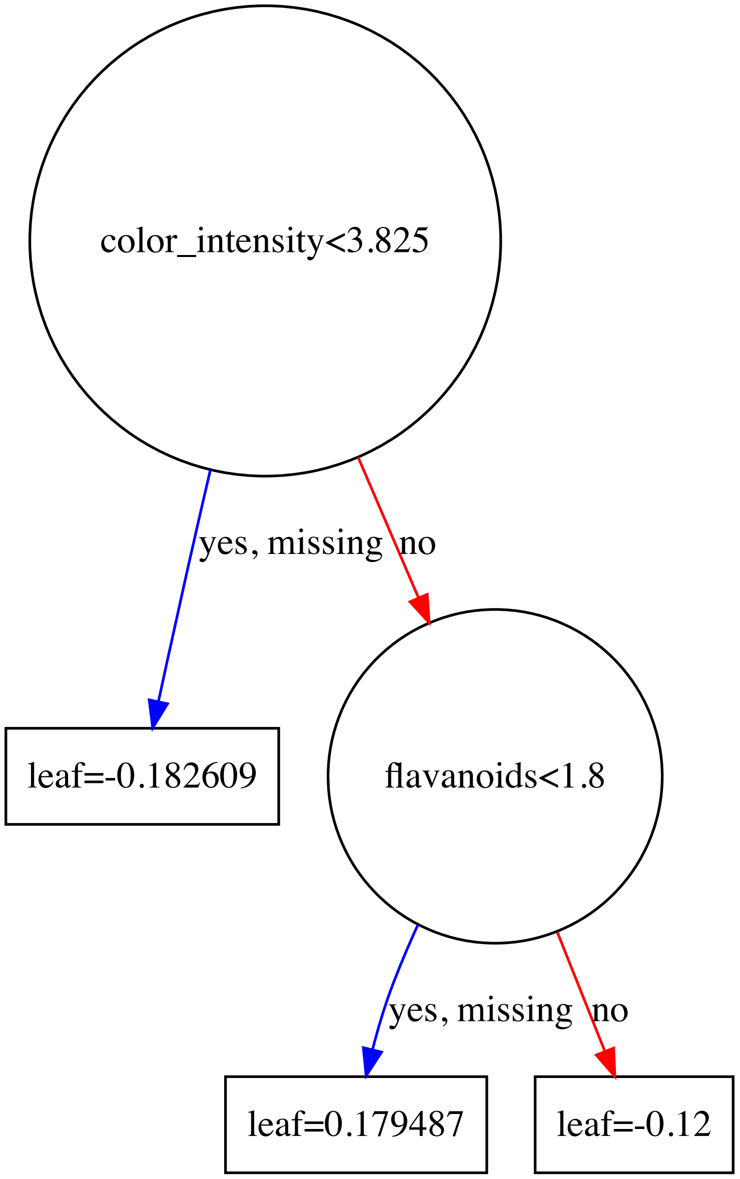
But can I trust the model?

We can plot/display it directly

In [351]:

```
xgb.to_graphviz(xgb_clf_2, num_trees=0, size='100,150')
```

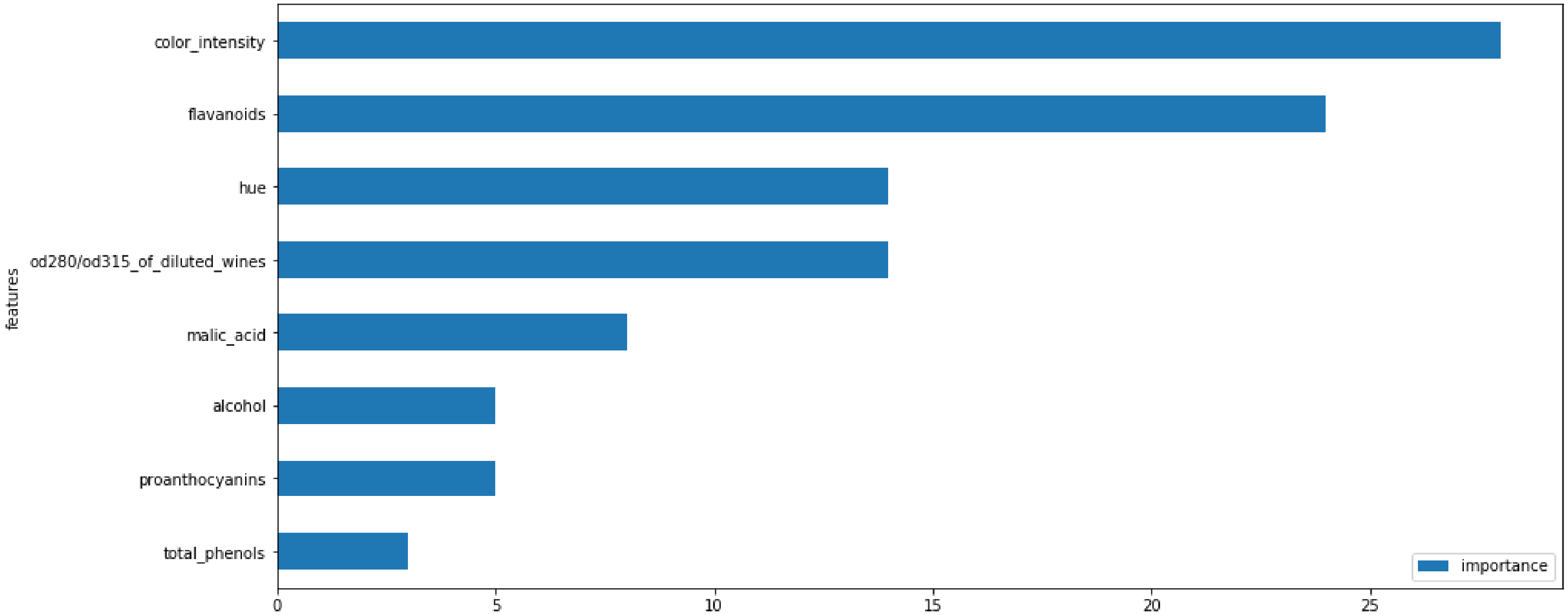
Out[351]:



Feature Importance

In [232]:

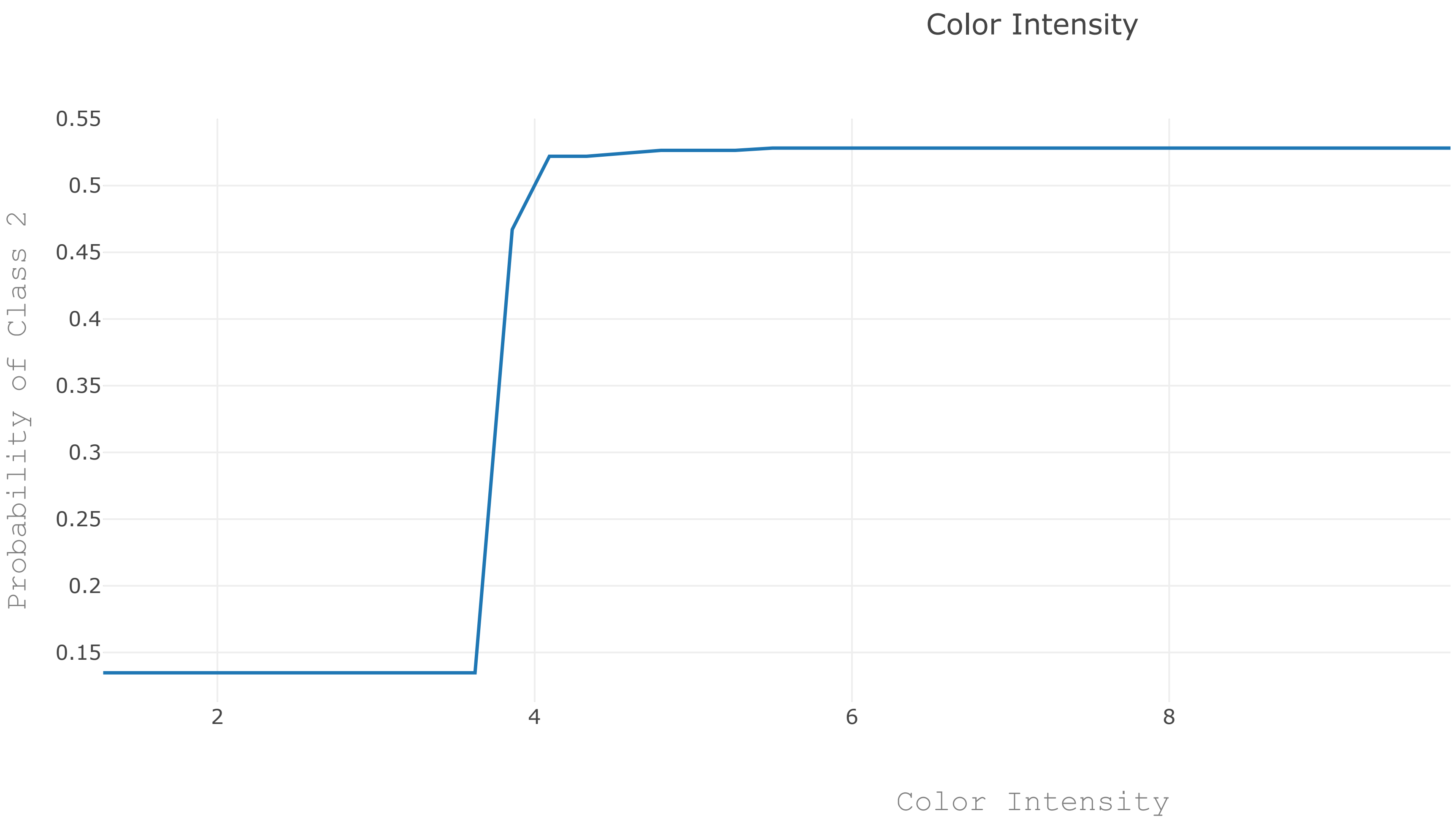
```
features_importance = pd.DataFrame.from_dict(xgb_clf_2.get_fscore(), orient='index').reset_
index().rename(columns={'index': 'features', 0: 'importance'})
features_importance.sort_values('importance', ascending=False, inplace=True)

features_importance.plot.barh(x='features', y='importance', figsize=(15,7))
plt.gca().invert_yaxis()
```



Partial Dependence Plot

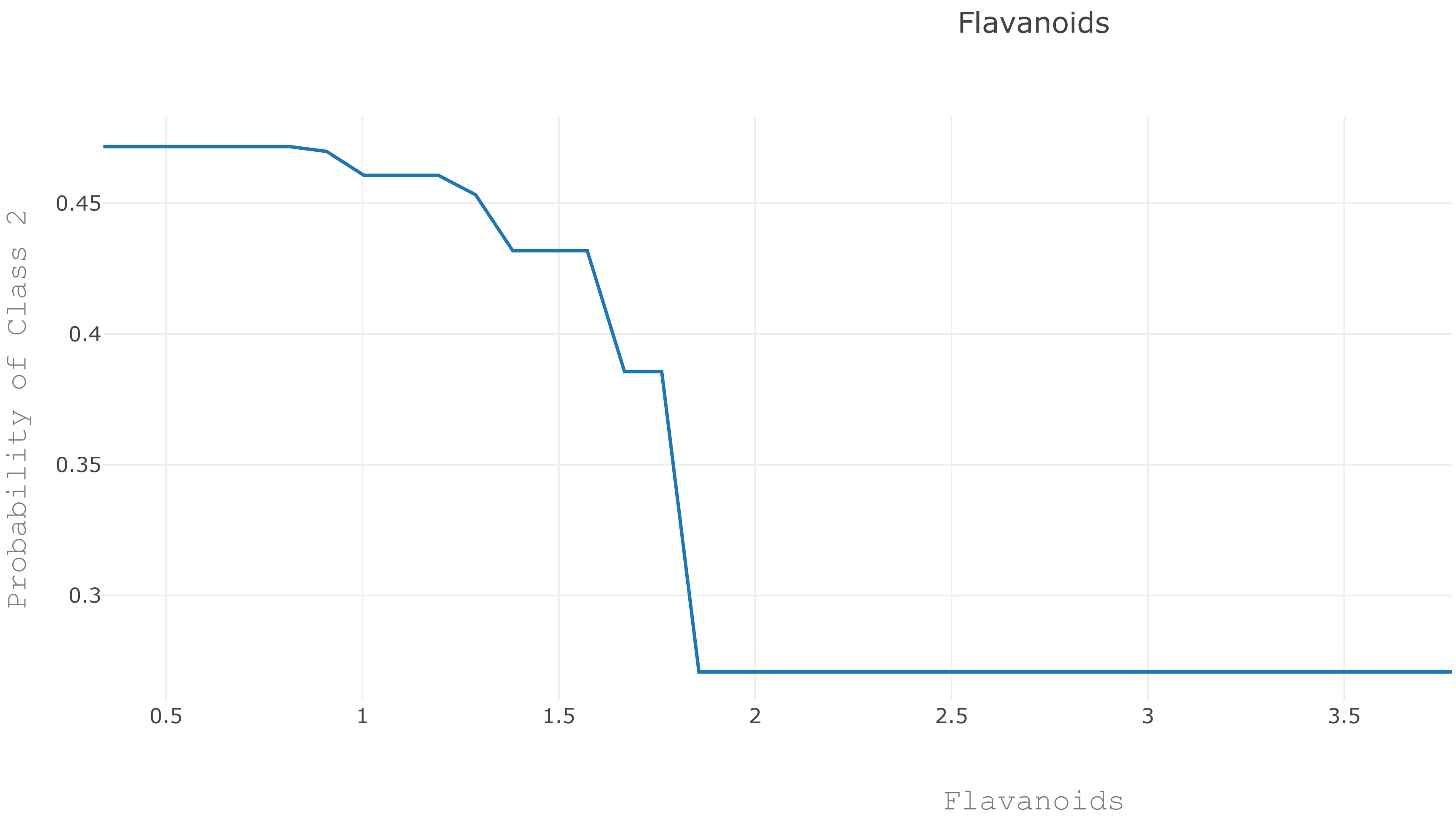
- Display how response functions change based on the values of one or two independent variables of interest
- Averaging out the effects of all other independent variables

In [235]:
```
df_plot = pd.read_csv("./data/pdp/color_int.csv")
iplot(plotly_figure(df_plot, 'Color Intensity', 'Color Intensity',
                    'Probability of Class 2', 'color_intensity', "yhat"),
     filename="figure")
```



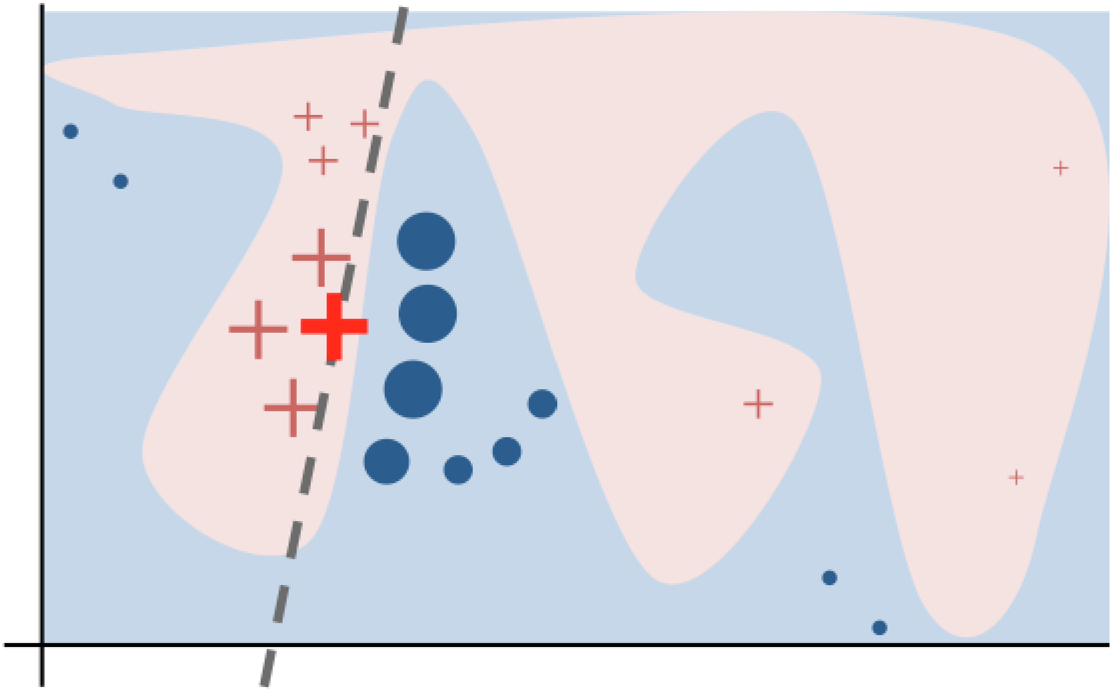
In [236]:

```
df_plot = pd.read_csv("./data/pdp/flavanoids.csv")
iplot(plotly_figure(df_plot, 'Flavanoids', 'Flavanoids',
                    'Probability of Class 2', 'flavanoids', "yhat"),
     filename="figure")
```



Local Interpretable Model-Agnostic Explanation (LIME)

- local linear approximation of the model's behaviour
- perturb the instance we want to explain and learn a sparse linear model around it



In [157]:

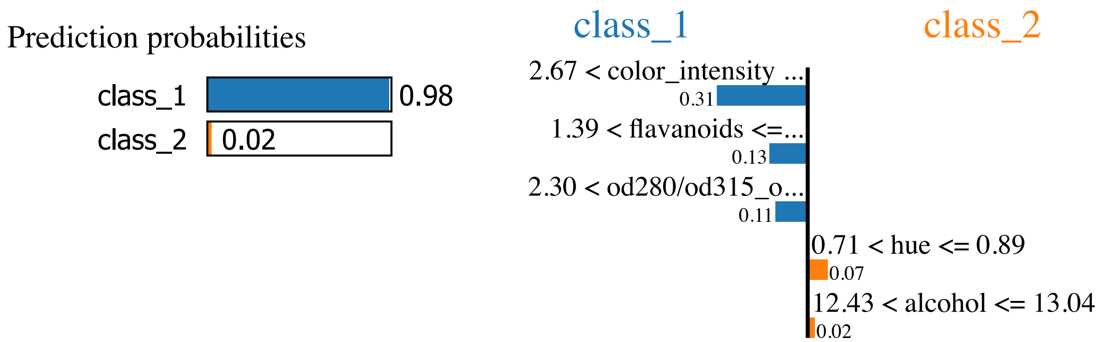
```
explainer = lime_tabular.LimeTabularExplainer(
    X_train,
    feature_names=data.feature_names,
    class_names=data.target_names,
    discretize_continuous=True
)
```

In [158]:
```
i = np.random.randint(0, X_test.shape[0])
print i
test_instance = X_test[i].copy()
```

```
0
```

In [215]:
```
exp = explainer.explain_instance(test_instance, xgb_clf.predict_proba, num_features=5)
```

In [216]:
```
exp.show_in_notebook(show_table=True, show_all=False)
```



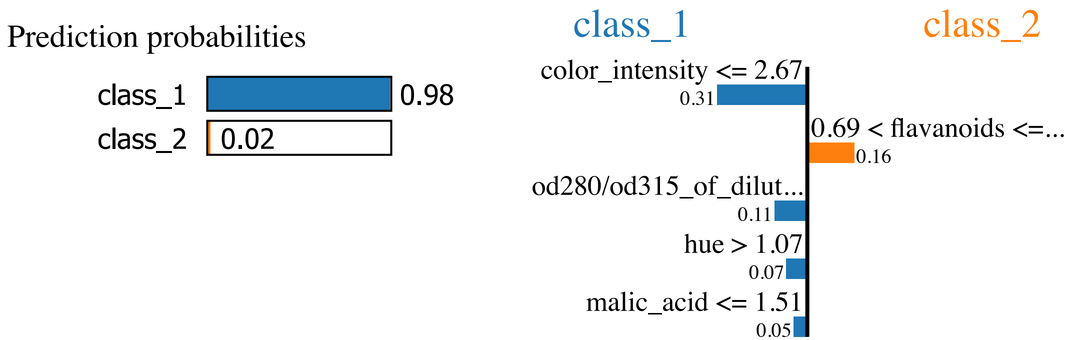
Feature	Value
color_intensity	3.30
flavanoids	1.76
od280/od315_of_diluted_wines	2.42
hue	0.88
alcohol	12.72

In [161]:
```
i = np.random.randint(0, X_test.shape[0])
print i
test_instance = X_test[i].copy()
```

7

In [175]:
```
exp = explainer.explain_instance(test_instance, xgb_clf.predict_proba, num_features=5)
```

In [177]:
```
exp.show_in_notebook(show_table=True, show_all=False)
```



Feature	Value
color_intensity	1.00
flavanoids	1.28
od280/od315_of_diluted_wines	3.07
hue	1.28
malic_acid	1.19

In [191]:

```
i = np.random.randint(0, X_test.shape[0])
print i
test_instance = X_test[i].copy()
```

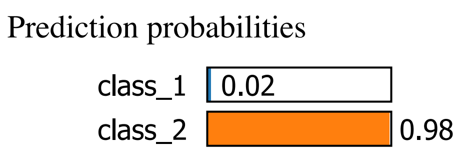
```
18
```

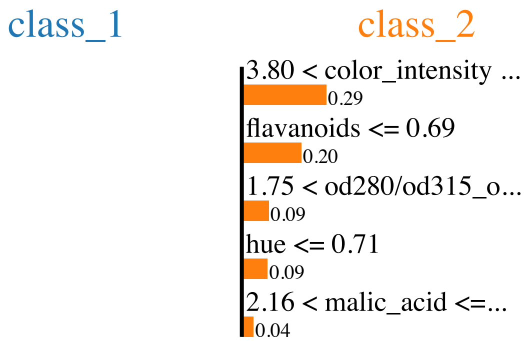
In [218]:

```
exp = explainer.explain_instance(test_instance, xgb_clf.predict_proba, num_features=5)
```

In [219]:

```
exp.show_in_notebook(show_table=True, show_all=False)
```





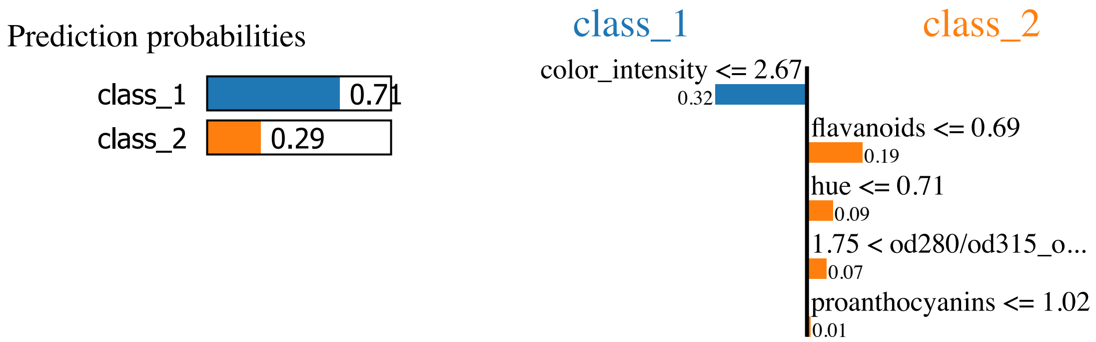
Feature	Value
color_intensity	5.50
flavanoids	0.49
od280/od315_of_diluted_wines	1.83
hue	0.66
malic_acid	3.03

In [220]:

```
test_instance[9] = 2.5
```

In [221]:
```
exp = explainer.explain_instance(test_instance, xgb_clf.predict_proba, num_features=5)
```

In [222]:
```
exp.show_in_notebook(show_table=True, show_all=False)
```

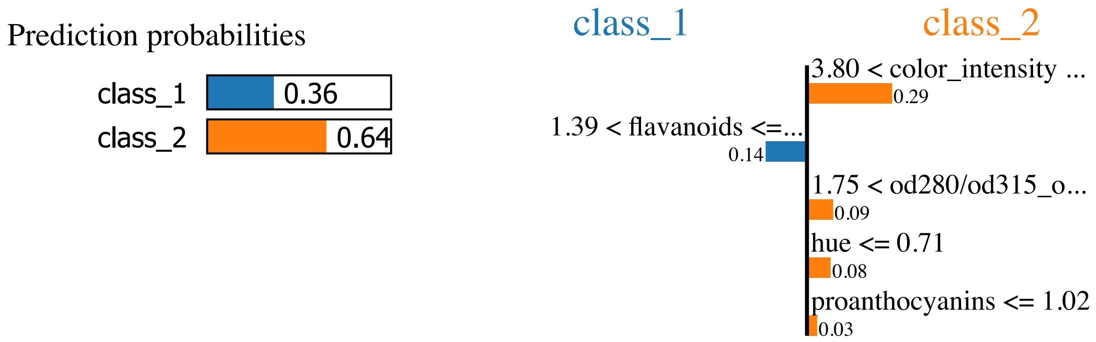


Feature	Value
color_intensity	2.50
flavanoids	0.49
hue	0.66
od280/od315_of_diluted_wines	1.83
proanthocyanins	0.73

In [223]:
```
test_instance = X_test[i].copy()
test_instance[6] = 1.9
```

In [224]:
```
exp = explainer.explain_instance(test_instance, xgb_clf.predict_proba, num_features=5)
```

In [225]:
```
exp.show_in_notebook(show_table=True, show_all=False)
```



Feature	Value
color_intensity	5.50
flavanoids	1.90
od280/od315_of_diluted_wines	1.83
hue	0.66
proanthocyanins	0.73

Interpreting Document Classification Model

Dataset used: 20 newsgroups

[http://qwone.com/~jason/20Newsgroups/ (http://qwone.com/~jason/20Newsgroups/)](http://qwone.com/~jason/20Newsgroups/)

The 20 Newsgroups data set is a collection of approximately 20,000 newsgroup documents, across 20 different newsgroups.

In [237]:

```
from sklearn.datasets import fetch_20newsgroups
newsgroups_train = fetch_20newsgroups(subset='train', remove=('headers', 'footers', 'quote
s'))
newsgroups_test = fetch_20newsgroups(subset='test',remove=('headers', 'footers', 'quotes'))
# making class names shorter
class_names = [x.split('.')[-1] if 'misc' not in x else '.'.join(x.split('.')[-2:]) for x i
n newsgroups_train.target_names]
class_names[3] = 'pc.hardware'
class_names[4] = 'mac.hardware'
```

```
Downloading 20news dataset. This may take a few minutes.
Downloading dataset from https://ndownloader.figshare.com/files/5975967 (14 MB)
```

In [238]:

```
print(','.join(class_names))
```

```
atheism,graphics,ms-windows.misc,pc.hardware,mac.hardware,x,misc.forsale,autos,motorcyc
les,baseball,hockey,crypt,electronics,med,space,christian,guns,mideast,politics.misc,re
ligion.misc
```

In [311]:

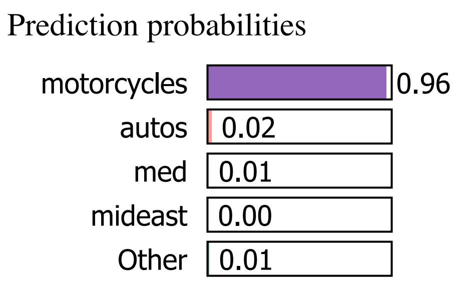
```
from sklearn.naive_bayes import MultinomialNB
from lime import lime_text
from sklearn.pipeline import make_pipeline
from lime.lime_text import LimeTextExplainer
```

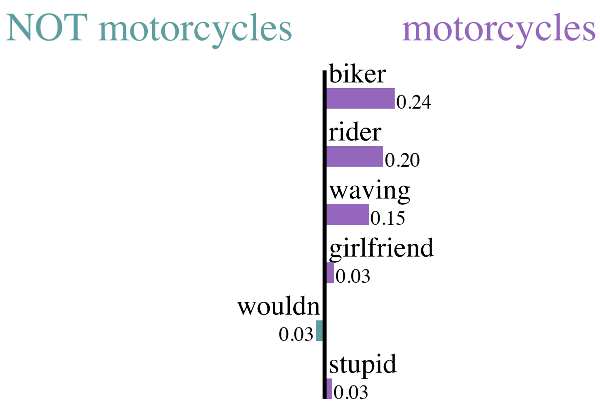
In [312]:
```
train_vectors = vectorizer.fit_transform(newsgroups_train.data)
test_vectors = vectorizer.transform(newsgroups_test.data)
nb = MultinomialNB(alpha=.01)
nb.fit(train_vectors, newsgroups_train.target)
c = make_pipeline(vectorizer, nb)
explainer = LimeTextExplainer(class_names=class_names)
```

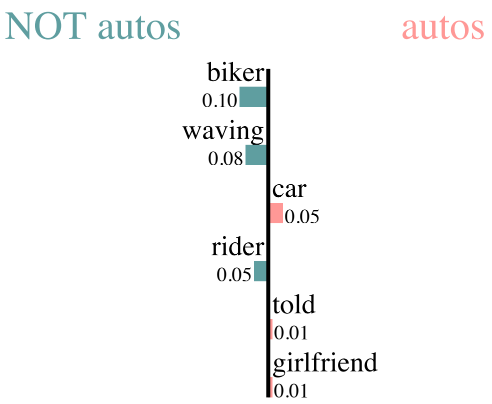
In [322]:

```
exp = explainer.explain_instance(newsgroups_test.data[idx], c.predict_proba,
num_features=6, top_labels=2)
print(exp.available_labels())
exp.show_in_notebook(text=newsgroups_test.data[idx])
```

```
[8, 7]
```







Text with highlighted words

I did it once with a biker-girlfriend in the car, and she told me that I was stupid, the rider wouldn't know why I was waving.

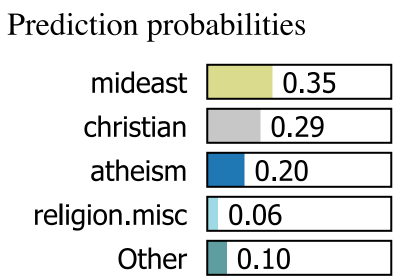
...She's long gone...

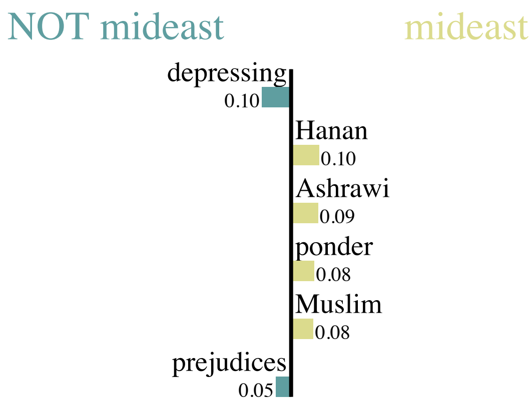
One.

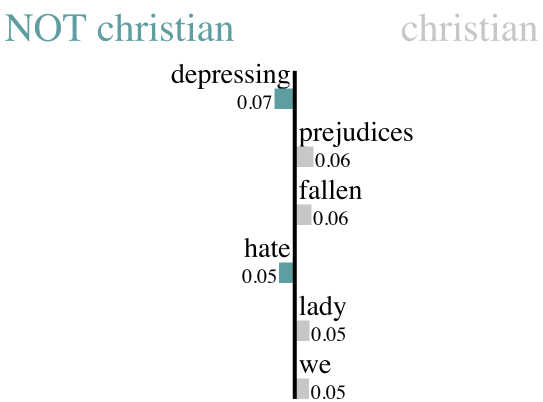
In [334]:

```
exp = explainer.explain_instance(newsgroups_test.data[idx], c.predict_proba,
num_features=6, top_labels=2)
print(exp.available_labels())
exp.show_in_notebook(text=newsgroups_test.data[idx])
```

```
[17, 15]
```







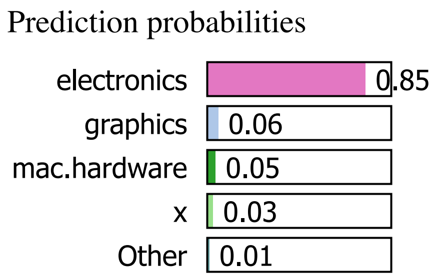
Text with highlighted words

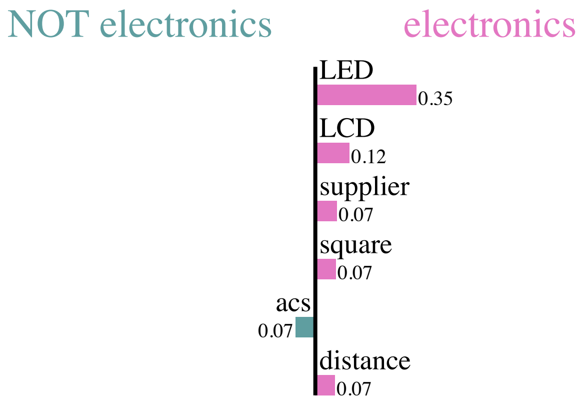
I would like to publicly apologize to our Anisa Aldoubosh
for playing :
****
Well Anisa I am not sure that I feel the necessary remorse.
You and another Muslim lady (Hanan Ashrawi) seems to me to
be some attempt to charm the west into forgetting what you
are really saying.
It is not that we hate muslims but we hate certain things you
are saying every now and then. And it is depressing to ponder the
prospects for peace while those wievs are held by your people.
Not that we are better then you , we have our own prejudices
and vices in the West thank you. But your views are really
depressing . Thus I have fallen in the temptation to tease
and make a little fun instead of ....
and have problems to mobilize the necessary remorse!

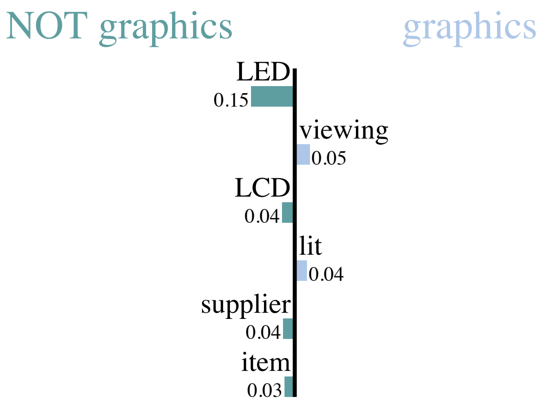
In [346]:

```
exp = explainer.explain_instance(newsgroups_test.data[idx], c.predict_proba,
num_features=6, top_labels=2)
print(exp.available_labels())
exp.show_in_notebook(text=newsgroups_test.data[idx])
```

```
[12, 1]
```







Text with highlighted words

I am interested in finding a supplier for an array of leds on material which is transparent when nothing is lit.

I'm not quite sure what LCD screens are like away from the laptop but I would guess they are not too clear.

An ideal item would be an LED array for which each LED is about 1/2" square. (Yes very course) This is for distance viewing, but on a window.

Any pointers of suggestions would be much appreciated.

-Mark Battisti
mbattist@magnus.acs.ohio-state.edu

Thank you

- https://github.com/marcotcr/lime (https://github.com/marcotcr/lime)
- https://www.oreilly.com/learning/introduction-to-local-interpretable-model-agnostic-explanations-lime (https://www.oreilly.com/learning/introduction-to-local-interpretable-model-agnostic-explanations-lime)
- https://www.oreilly.com/ideas/ideas-on-interpreting-machine-learning (https://www.oreilly.com/ideas/ideas-on-interpreting-machine-learning)

Hiring at ds-jobs@go-jek.com (ds-jobs@go-jek.com)!

Conducting Deep Learning Masterclass in NUS on 26th Oct together with 3 deep learning practitioners, for more information pls visit: NUS ISS Deep Learning Masterclass (https://www.iss.nus.edu.sg/community/events/event-details/2017/10/26/default-calendar/deep-learning-masterclass-on-computer-vision?utm=ad-FBpost)