Booz | Allen | Hamilton

RISK BASED PRIORITIZATION

NOVEMBER 3, 2016

WASHINGTON, DC

Agencies currently operate in a complex environment with a need to protect program integrity to fulfill their missions

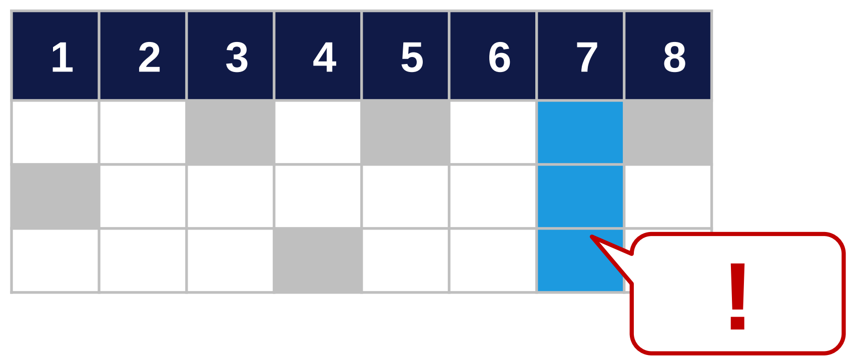
OUR CONTEXT TODAY

- Declining budgets – in real terms – have constrained programmatic spending over the previous 4+ years across many Federal agencies
- Scrutiny of public programs and ongoing public debates on the Federal government's effectiveness has increased over the same period, including:
    - New mandates for oversight and regulation
    - Elevated public expectations for action and accountability
    - Contentious legislative implementations
    - Increased reporting and transparency about program performance
    - Increasing levels of sophistication in malicious actors
- In many cases, the bad actors understand both the pressures and constraints on Federal agencies and are emboldened to continue and expand undesirable behaviors

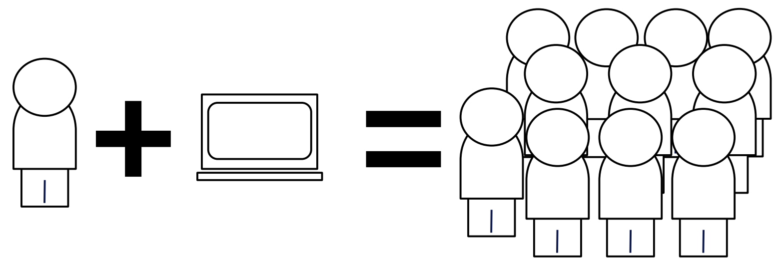
The application of advanced analytics is a powerful component in the fight against fraud and prioritizing risk

THE PROMISE

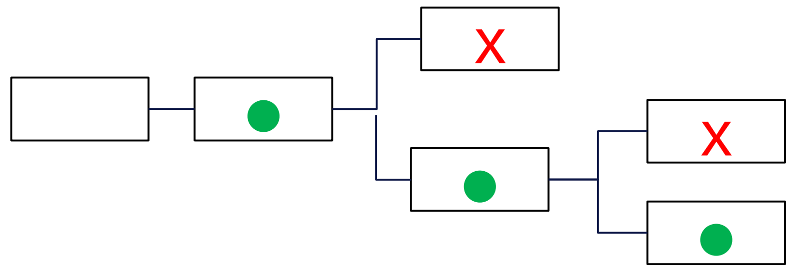
- Provides a deeper understanding of “what” and “why” something may occur



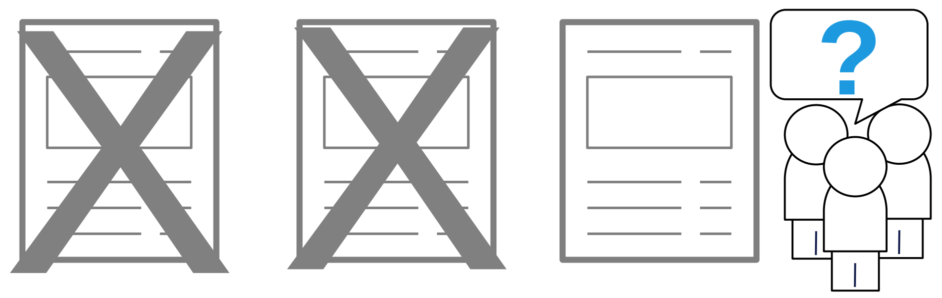
- Scales and, most frequently, improves human judgement



- Simulates decision outcomes at lower-risk to inform decision making and planning



- Enables focusing scarce resources on the highest priorities or greatest potential risks



Risk based prioritization is a critical tool to identify, manage, and mitigate risk across a variety of missions and programs

USE CASES

FRAUD DETECTION

For teams who make sure that payments made by their agency reach the intended recipients…

Benefits Payments
Tax Refunds
Award Spending

CASE SELECTION

For teams who keep the system safe for everyone through investigation and enforcement actions…

Bank Examinations
Food Inspections
Illegal Trade

SAFETY + PREVENTION

For teams who protect against disasters and epidemics through mitigation and preparedness…

Disaster Preparedness
Forest Management
Consumer Protection

Data science techniques can direct scarce resources to the most effective interventions in risk based prioritization

TECHNIQUES

TEST & TREATMENT

Simulation
Experimentation
Prioritization
New Treatments

MODELING

Clustering
Classification
Regression

EXPLORATION

Odds Ratios
Principal Components
Correlations

Box Plots
Histograms
Scatter Plots

Impact to Mission

TECHNIQUES & METHODS

DEMONSTRATION

RISK BASED PRIORITIZATION TECHNIQUES

Case Study: Chief Sommelier of the United States jupyter explorWine Last Checkpoint: Last Sunday at 3:13 PM (unsaved changes)

File Edit View Insert Cell Kernel Help | Python 2

Markdown | CellToolbar

EXPLORATION

In [ ]:

```
import os
os.getcwd()
```

In [1]:

```
import numpy as np
import pandas as pd
whitedata = np.genfromtxt('../Data/winequality-white.csv',
                          delimiter=',',names=True)
brands = pd.read_csv('../Data/wine-brand.csv', header = 0)
```

jupyter exploreWine Last Checkpoint: Last Sunday at 3:13 PM (unsaved changes)

File Edit View Insert Cell Kernel Help Python 2

Markdown CellToolbar

Sneak-A-Peak

In [2]:

```
df = pd.DataFrame(whitedata)
df['brand'] = brands
df.head(5)
```

Out[2]:

	fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	pH	sulphates	alcohol	qua
0	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.0010	3.00	0.45	8.8	6.0
1	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.9940	3.30	0.49	9.5	6.0
2	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.9951	3.26	0.44	10.1	6.0
3	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6.0
4	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6.0

jupyter exploreWine Last Checkpoint: Last Sunday at 3:13 PM (unsaved changes)

File Edit View Insert Cell Kernel Help Python 2

Markdown CellToolbar

Summary Statistics

In [3]:
```
df.describe()
```

Out[3]:

	fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	pH
count	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000
mean	6.854788	0.278241	0.334192	6.391415	0.045772	35.308085	138.360657	0.994027	3.188267
std	0.843868	0.100795	0.121020	5.072058	0.021848	17.007137	42.498065	0.002991	0.151001
min	3.800000	0.080000	0.000000	0.600000	0.009000	2.000000	9.000000	0.987110	2.720000
25%	6.300000	0.210000	0.270000	1.700000	0.036000	23.000000	108.000000	0.991723	3.090000
50%	6.800000	0.260000	0.320000	5.200000	0.043000	34.000000	134.000000	0.993740	3.180000
75%	7.300000	0.320000	0.390000	9.900000	0.050000	46.000000	167.000000	0.996100	3.280000
max	14.200000	1.100000	1.660000	65.800000	0.346000	289.000000	440.000000	1.038980	3.820000

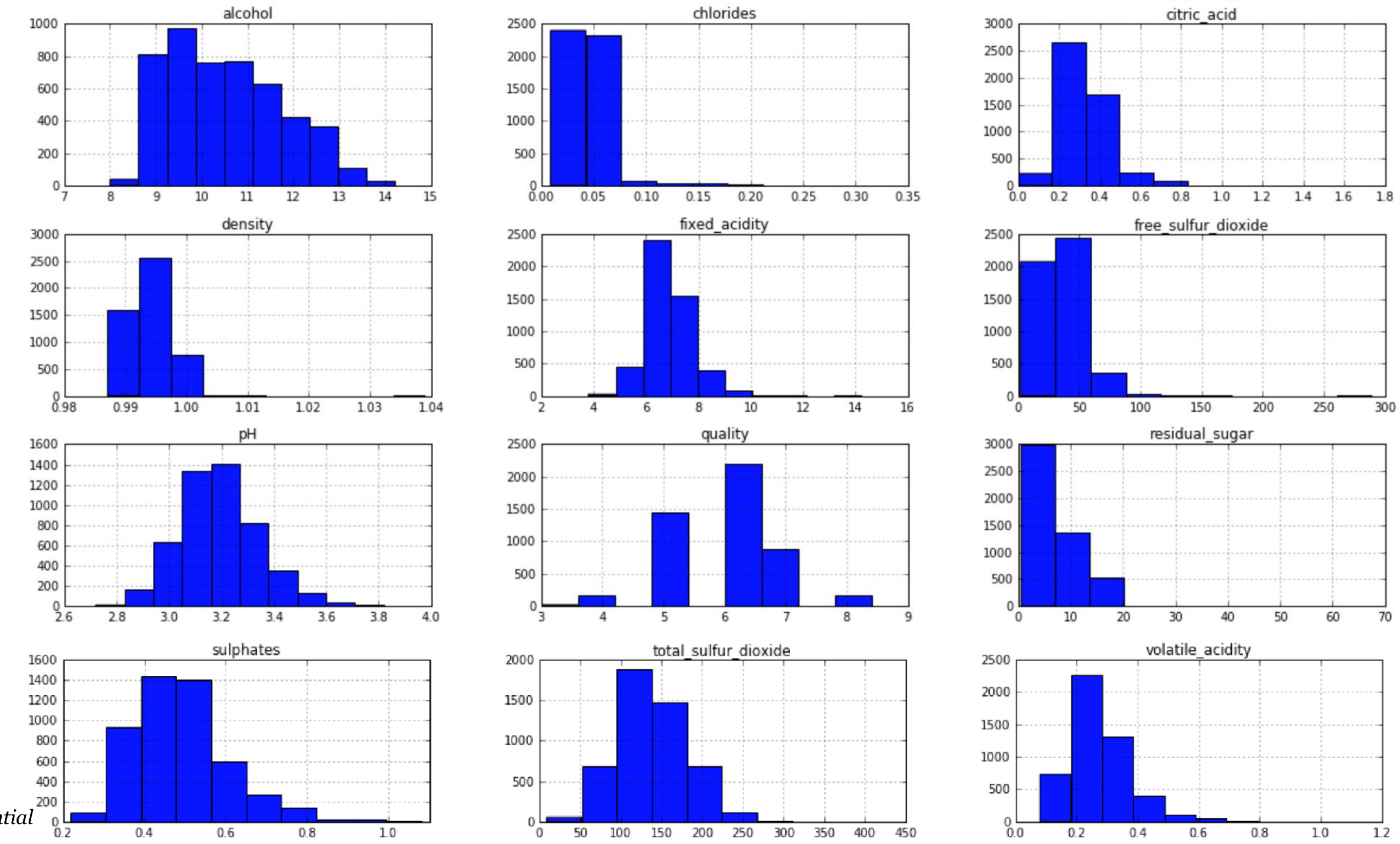
jupyter exploreWine Last Checkpoint: Last Sunday at 3:13 PM (unsaved changes)

File Edit View Insert Cell Kernel Help Python 2

Markdown CellToolbar

Histograms

In [4]:
```
from matplotlib import pylab as pl
%matplotlib inline
df[df.columns[0:12]].hist(figsize=(20,12))
```



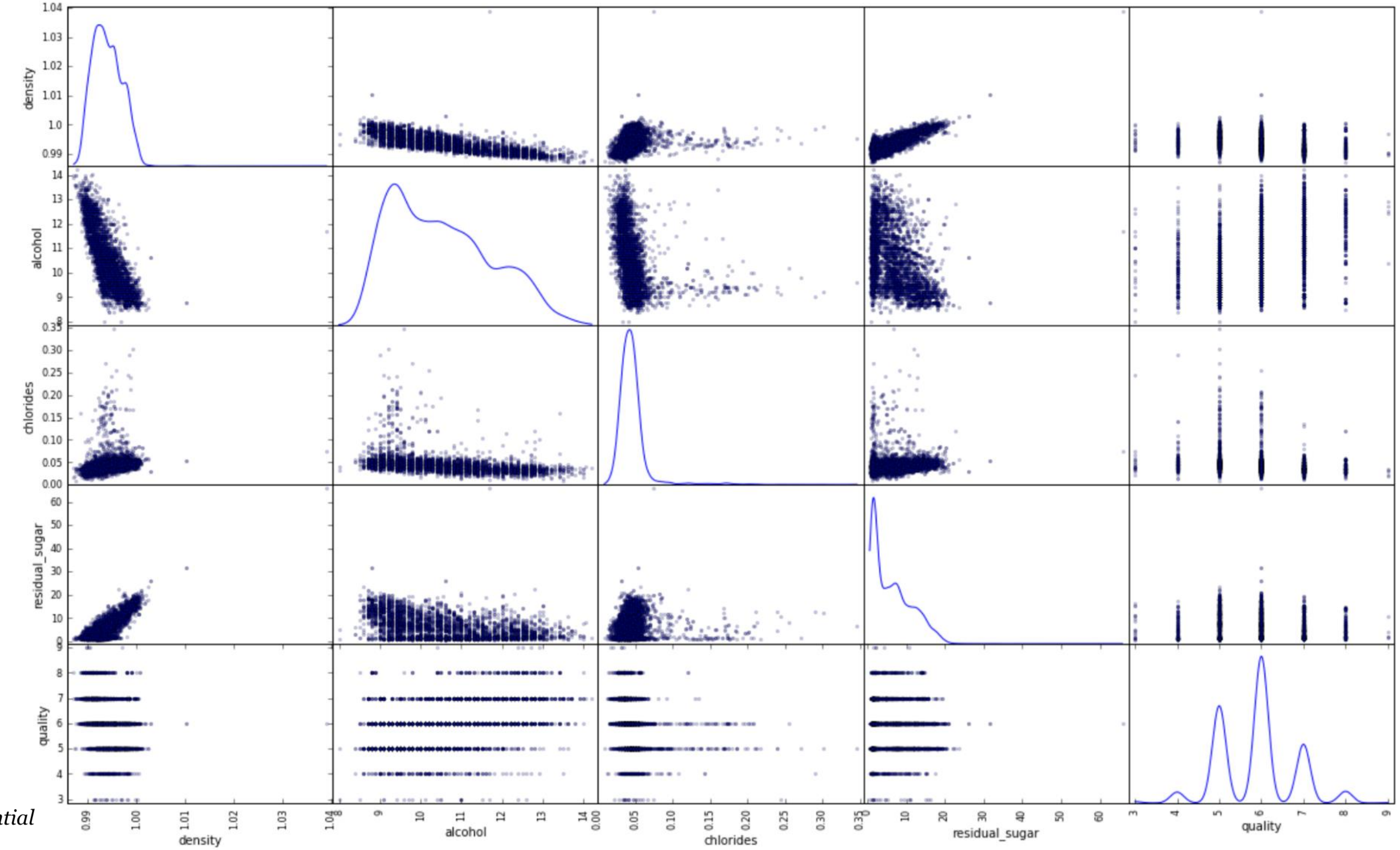
jupyter exploreWine Last Checkpoint: Last Sunday at 3:13 PM (unsaved changes)

File Edit View Insert Cell Kernel Help Python 2

Markdown CellToolbar

Scatter Plots

In [5]:
```
%matplotlib inline
from pandas.tools.plotting import scatter_matrix
scatter_matrix(df[['density','alcohol','chlorides','residual_sugar','quality']],
               alpha=0.2, figsize=(20, 12), diagonal='kde')
```



jupyter exploreWine Last Checkpoint: Last Sunday at 3:13 PM (unsaved changes)

File Edit View Insert Cell Kernel Help Python 2

Markdown CellToolbar

Dimensionality Reduction

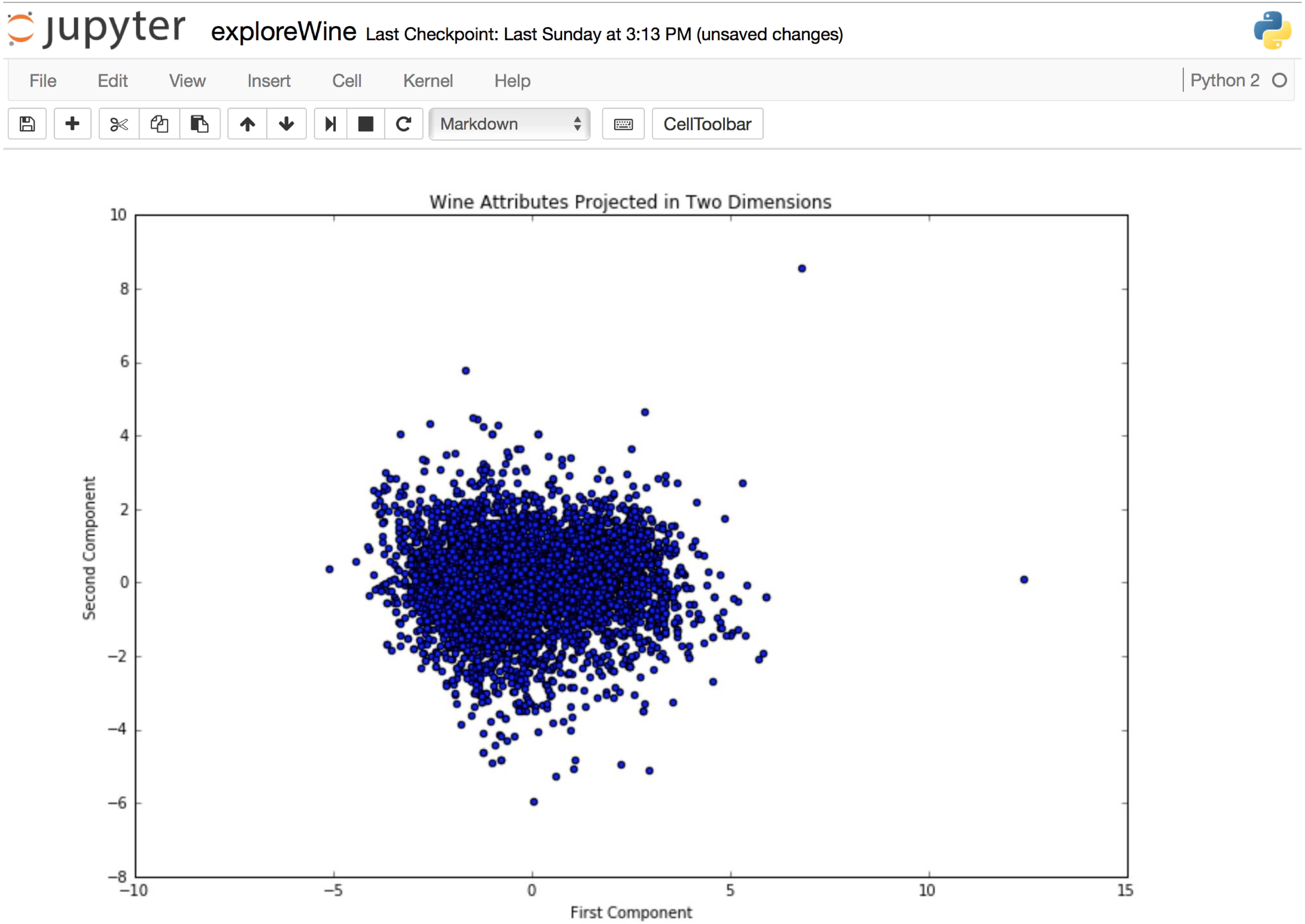
In [6]:

```
##CREATE ARRAYS OF ATTRIBUTES FOR USE IN STATISTICAL TECHNIQUES
from sklearn import preprocessing as pre
#create 2 dimensional array of attributes with quality attribute
names = list(whitedata.dtype.names)
Xdata = np.array([whitedata[name] for name in names]).transpose()
Xsca = pre.scale(Xdata)
#create 2 dimensional array without quality attribute
names.remove('quality')
Xdata_nq = np.array([whitedata[name] for name in names]).transpose()
Xsca_nq = pre.scale(Xdata_nq)
```

In [7]:

```
from sklearn.decomposition import TruncatedSVD
#from mpl_toolkits.mplot3d import Axes3D
svd = TruncatedSVD(n_components=3)
svd.fit(Xsca)
Xrot = svd.transform(Xsca)
print Xrot.shape
fig = pl.figure(figsize=(12,8))
pl.title('Wine Attributes Projected in Two Dimensions')
#ax = fig.add_subplot(111, projection='3d')
#ax.scatter(Xrot[np.random.choice(,0],Xrot[:,1],Xrot[:,2])
#ax.scatter(Xrot[indices[0],0],Xrot[indices[0],1],Xrot[indices[0],2], c='r', marker='*', s=80)
pl.scatter(Xrot[:,0],Xrot[:,1])
#pl.scatter(Xrot[indices[0],0],Xrot[indices[0],1], c='r', marker='*', s=200)
pl.xlabel('First Component')
pl.ylabel('Second Component')
pl.show()
```

```
(4898, 3)
```


jupyter
exploreWine Last Checkpoint: Last Sunday at 3:13 PM (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Markdown
CellToolbar
Wine Attributes Projected in Two Dimensions
Second Component
First Component
10
8
6
4
2
0
−2
−4
−6
−8
−10
−5
0
5
10
15

jupyter exploreWine Last Checkpoint: Last Sunday at 3:13 PM (unsaved changes)

File Edit View Insert Cell Kernel Help Python 2

Markdown CellToolbar

MODELING

Clustering

In [8]:
```
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
kmeans = KMeans(init='k-means++', n_clusters=4, n_init=1)
reduced_data = Xrot[:,0:2]
kmeans.fit(reduced_data)
```

jupyter exploreWine Last Checkpoint: Last Sunday at 3:13 PM (unsaved changes)

File Edit View Insert Cell Kernel Help Python 2

Markdown CellToolbar

Classification

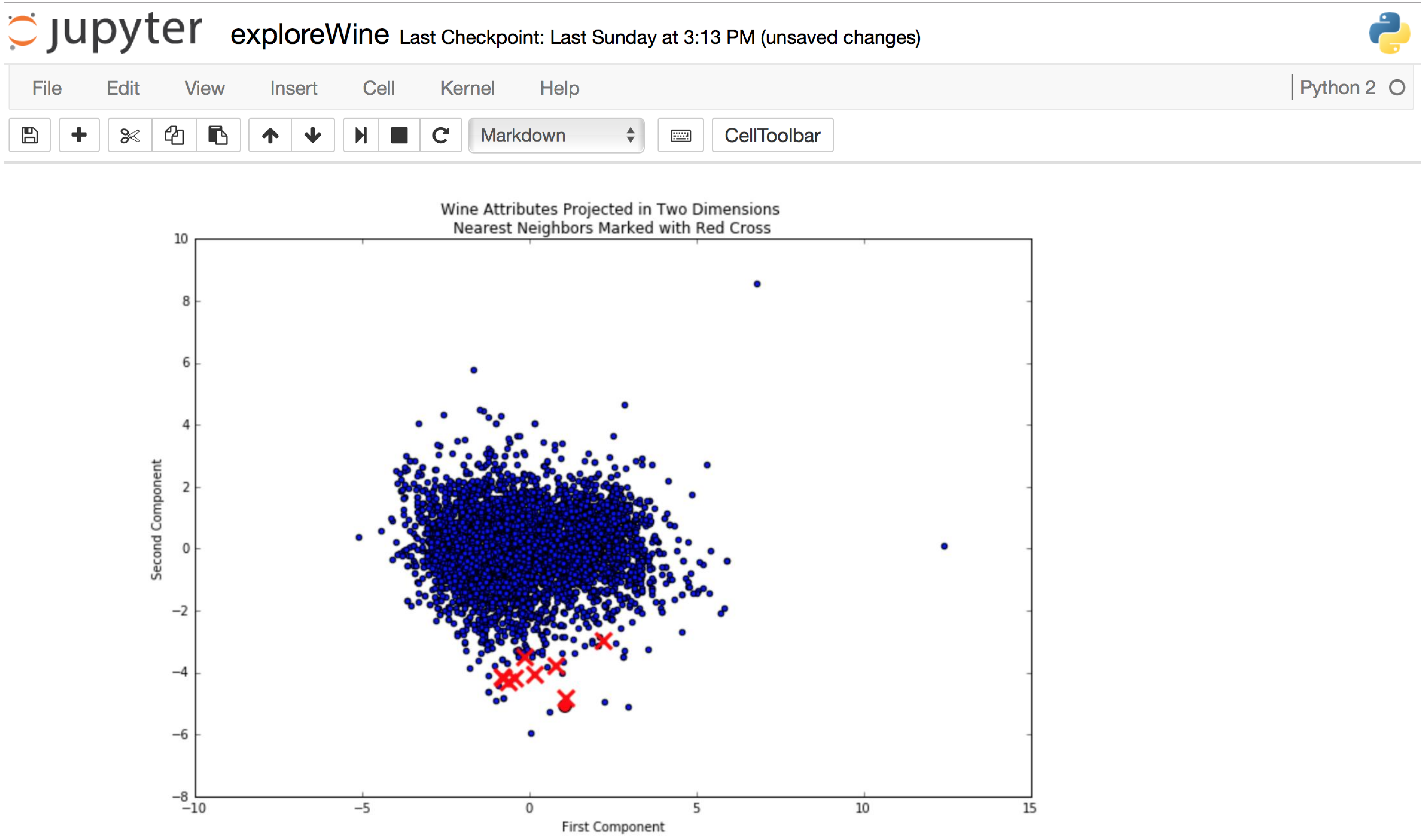
Nearest Neighbors

In [9]:
```
pick = 98
print(df.iloc[pick])
find = Xsca[pick,:]
```

```
fixed_acidity                                  9.8
volatile_acidity                              0.36
citric_acid                                   0.46
residual_sugar                                10.5
chlorides                                    0.038
free_sulfur_dioxide                              4
total_sulfur_dioxide                            83
density                                     0.9956
pH                                            2.89
sulphates                                      0.3
alcohol                                       10.1
quality                                          4
brand                  hidden moon splashed semillon
Name: 98, dtype: object
```

In [10]:
```
from sklearn.neighbors import NearestNeighbors
from sklearn.neighbors import KNeighborsClassifier
neighbors = NearestNeighbors(n_neighbors=10, algorithm='brute').fit(Xsca)
distances, indices = neighbors.kneighbors(find)
for i in range(len(indices[0])):
  print "%2f  %s" % (distances[0][i], df.brand[indices[0][i]])
```

```
0.000000  hidden moon splashed semillon
1.466411  gloaming frog enchanted chardonnay
2.343017  spring beguiling leaping sauvignon blanc
2.379129  frog hidden enchanted pinot gris
2.569685  honeybee spring enchanted chardonnay
2.643458  equinox moon splashed semillon
2.735275  equinox penguin splashed moscato
2.753440  seaside beguiling diving reisling
2.788180  spring seaside diving reisling
2.831303  mossy equinox splashed chardonnay
```


jupyter
exploreWine Last Checkpoint: Last Sunday at 3:13 PM (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Markdown
CellToolbar
Wine Attributes Projected in Two Dimensions
Nearest Neighbors Marked with Red Cross
Second Component
First Component
10
8
6
4
2
0
−2
−4
−6
−8
−10
−5
0
5
10
15

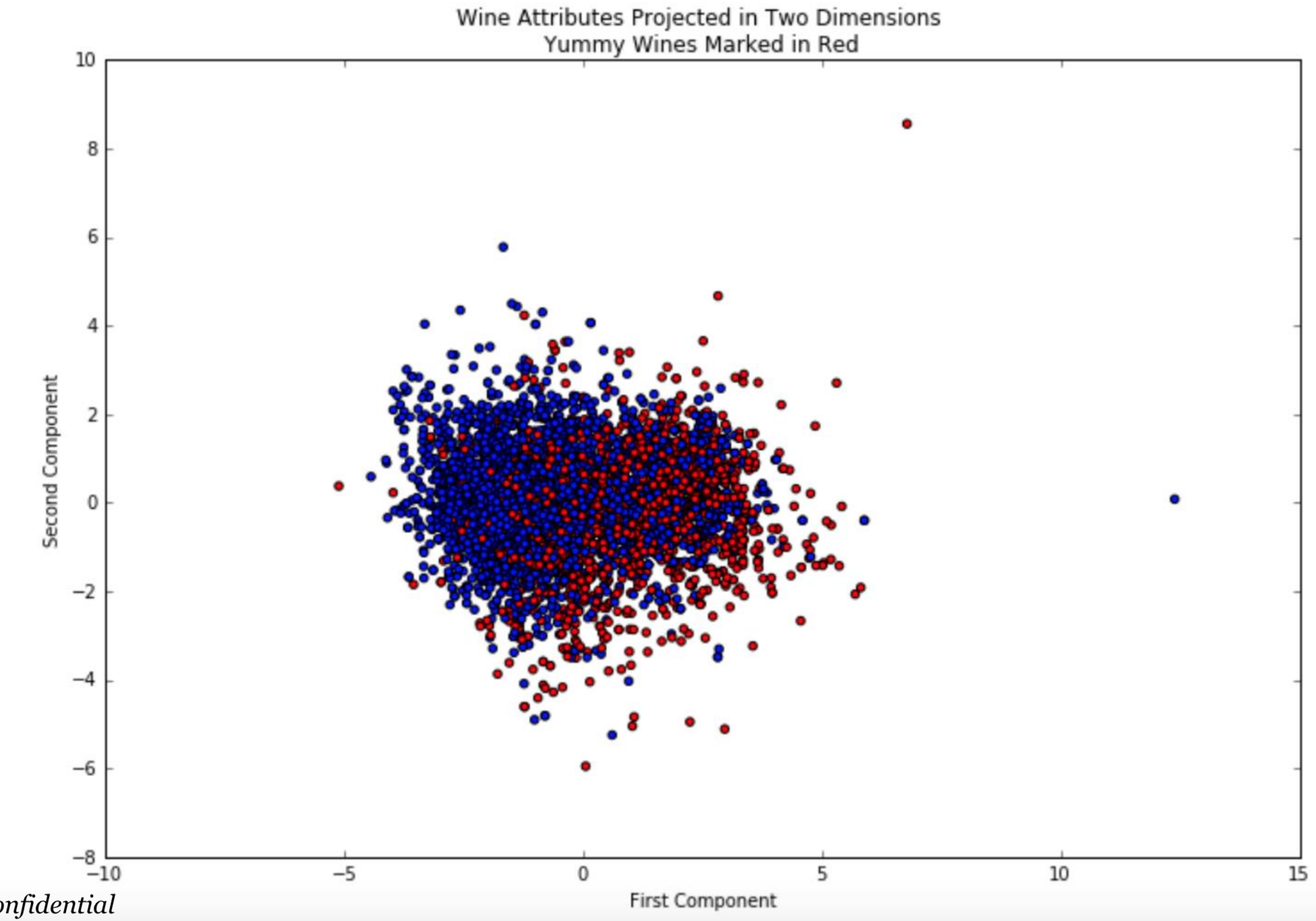
jupyter exploreWine Last Checkpoint: Last Sunday at 3:13 PM (unsaved changes)

File Edit View Insert Cell Kernel Help Python 2

Markdown CellToolbar

Support Vector Machine

In [12]:
```
##ADD LABELS, SUMMARIZE
classlabel = np.genfromtxt('../Data/winequality-2classlabels-white.csv',delimiter=',',names=True)
ldf = df
ldf['yum'] = pd.DataFrame(classlabel)
ldf.yum = ldf.yum.apply(lambda x:int(x))
print ldf.yum.describe()
```




```
count    4898.000000
mean        0.334831
std         0.471979
min         0.000000
25%         0.000000
50%         0.000000
75%         1.000000
max         1.000000
```

jupyter exploreWine Last Checkpoint: Last Sunday at 3:13 PM (unsaved changes)

File Edit View Insert Cell Kernel Help Python 2

Markdown CellToolbar

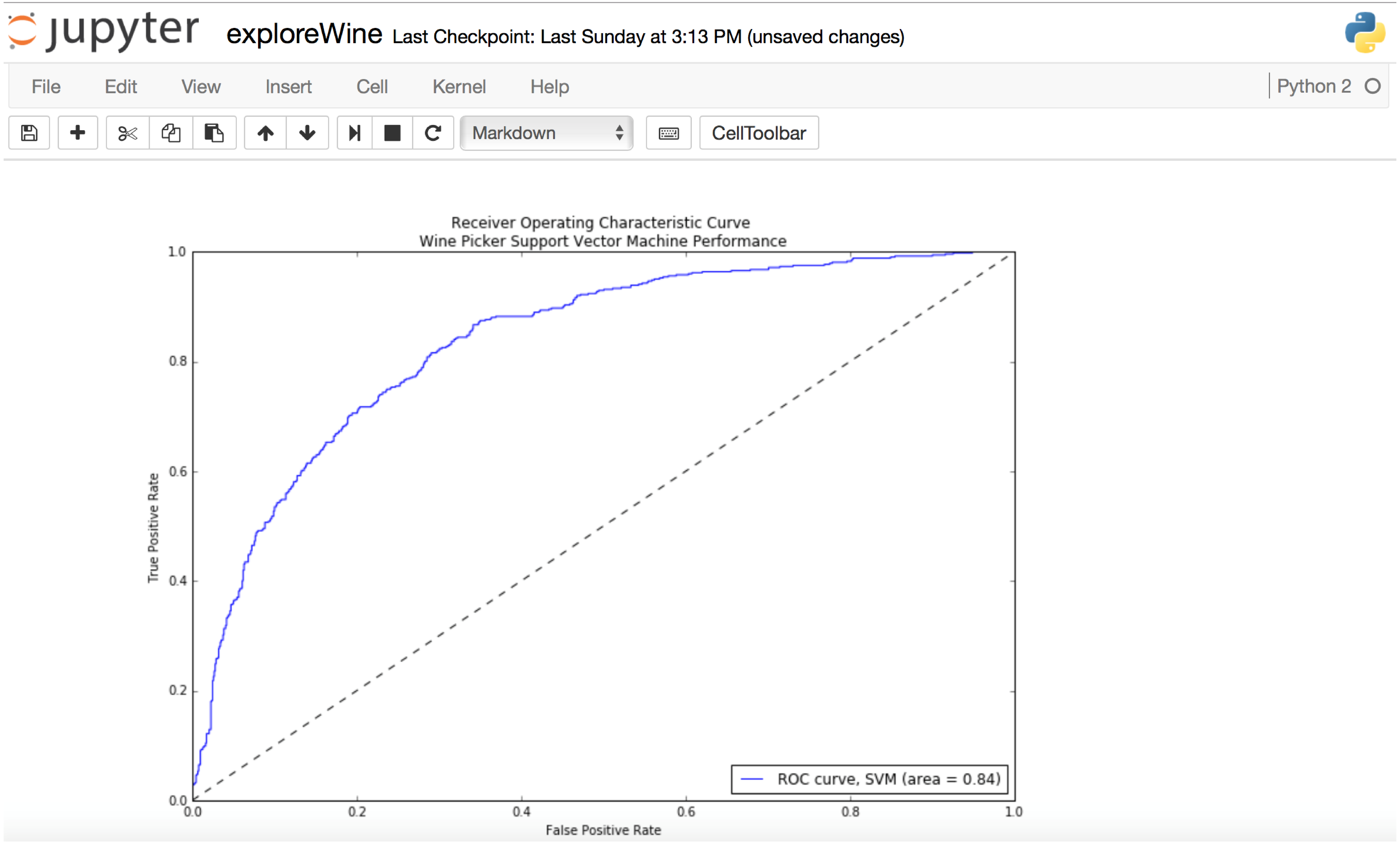
In [14]:
```
##SPLIT DATA INTO TRAINING AND TESTING SETS
from sklearn import cross_validation
X_train, X_test, y_train, y_test = cross_validation.train_test_split(
    Xsca_nq, classlabel, test_size=0.3, random_state=0)
print '--TRAINING SET--'
print 'Total wines in set: ', format(len(y_train), ",d")
print 'Of which, Yummy Wines: ', format(sum([int(x[0]) for x in y_train.tolist()]), ",d")
print '\n--TEST SET--'
print 'Total Wines set: ', format(len(y_test), ",d")
print 'Of which, Yummy Wines: ', format(sum([int(x[0]) for x in y_test.tolist()]), ",d")
```

```
--TRAINING SET--
Total wines in set:  3,428
Of which, Yummy Wines:  1,113

--TEST SET--
Total Wines set:  1,470
Of which, Yummy Wines:  527
```

In [15]:
```
##FIT THE SVM MODEL ON THE TRAINING SET, OUTPUT THE MEAN ACCURACY SCORE ON TEST SET
from sklearn import svm
clf = svm.SVC( probability=True, random_state=0)
clf = clf.fit(X_train,y_train)
y_pred_svm = clf.predict(X_test)
probas_svm = clf.predict_proba(X_test)
print '--Support Vector Machine-- \nMean Accuracy Score: ', clf.score(X_test, y_test)
```

```
--Support Vector Machine--
Mean Accuracy Score:  0.765306122449
```


jupyter
exploreWine Last Checkpoint: Last Sunday at 3:13 PM (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Markdown
CellToolbar
Receiver Operating Characteristic Curve
Wine Picker Support Vector Machine Performance
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve, SVM (area = 0.84)

jupyter exploreWine Last Checkpoint: Last Sunday at 3:13 PM (unsaved changes)

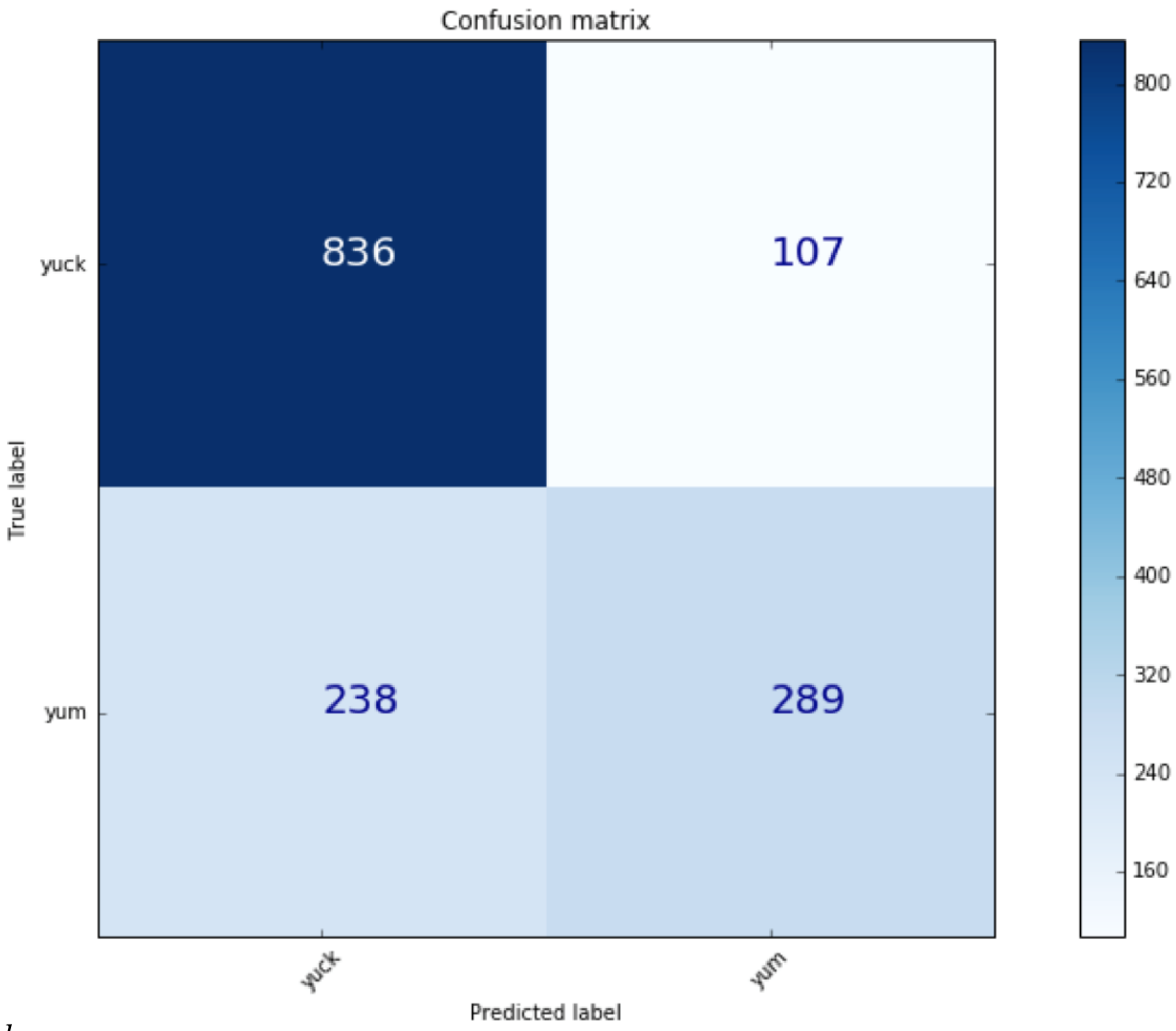
File Edit View Insert Cell Kernel Help Python 2

Markdown CellToolbar

TREATMENT

Confusion Matrix

```
Total Wines in TEST set:  1,470
Of which, Yummy Wines:  527
```



LESSONS FROM OUR WORK

Our experience is that it is the approach and teams – not the tools – that delivers success in these efforts

LESSONS FROM THE TRENCHES

1. There are no magical tools, perfect data, nor ultra-secret or proprietary techniques that will solve Risk Based Prioritization or fraud out of the box
2. Analytical models must be developed with deep appreciation and regard for the business context and the resources that will use the output
3. Superlative results arise from a structured process of iteration, experimentation, and rigorous analysis of data and outcomes
4. The risk, adversary, or actor is constantly evolving and evolves faster than teams expect – using historical patterns must be balanced with other methods to discern, and adapt to, new events or changes in behaviors
5. Unbridled curiosity and doggedness is absolutely necessary – a shared sense of mission and resolve fuel great teams

Booz | Allen | Hamilton