In [4]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
plt.rcParams["figure.figsize"] = (10, 20)
import mpld3
mpld3.enable_notebook()
```

In [8]:

```
df = pd.read_csv('bioresponse_csv.csv')
```

In [10]:

```
pd.set_option('display.max_columns', None)
df.head()
```

Out[10]:

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	
0	0.000000	0.497009	0.10	0.0	0.132956	0.678031	0.273166	0.585445	0.743663	0.243144	(
1	0.366667	0.606291	0.05	0.0	0.111209	0.803455	0.106105	0.411754	0.836582	0.106480	(
2	0.033300	0.480124	0.00	0.0	0.209791	0.610350	0.356453	0.517720	0.679051	0.352308	(
3	0.000000	0.538825	0.00	0.5	0.196344	0.724230	0.235606	0.288764	0.805110	0.208989	(
4	0.100000	0.517794	0.00	0.0	0.494734	0.781422	0.154361	0.303809	0.812646	0.125177	(

In [11]:

```
df.isnull().sum()
```

Out[11]:

```
D1        0
D2        0
D3        0
D4        0
D5        0
         ..
D1773     0
D1774     0
D1775     0
D1776     0
target    0
Length: 1777, dtype: int64
```

In [12]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3751 entries, 0 to 3750
Columns: 1777 entries, D1 to target
dtypes: float64(1776), int64(1)
memory usage: 50.9 MB
```

In [13]:

```
df['target'].value_counts()
```

Out[13]:

```
1    2034
0    1717
Name: target, dtype: int64
```

In [14]:

```
X = df.drop(['target'], axis = 1)
y = df['target']
```

In [15]:

```
from sklearn.model_selection import train_test_split
```

In [16]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

In [17]:

```
from sklearn.linear_model import LogisticRegression
```

In [18]:

```
model = LogisticRegression()
```

In [19]:

```
model.fit(X_train, y_train)
```

```
C:\Users\SURYA\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:
432: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Speci
fy a solver to silence this warning.
  FutureWarning)
```

Out[19]:

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='warn', tol=0.0001, verbose=0,
                   warm_start=False)
```

In [20]:

```
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
```

In [21]:

```
predict = model.predict(X_test)
```

In [22]:

```
print(classification_report(predict, y_test))
```

```
              precision    recall  f1-score   support

           0       0.73      0.73      0.73       336
           1       0.78      0.78      0.78       415

    accuracy                           0.76       751
   macro avg       0.75      0.75      0.75       751
weighted avg       0.76      0.76      0.76       751
```

In [23]:

```
from sklearn.ensemble import RandomForestClassifier
```

In [24]:

```
clf = RandomForestClassifier(n_estimators = 10)
```

In [25]:

```
clf.fit(X_train, y_train)
```

Out[25]:

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=None, max_features='auto', max_leaf_nodes=N
one,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=10,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

In [26]:

```
pre = clf.predict(X_test)
```

In [27]:

```
print(classification_report(pre, y_test))
```

```
              precision    recall  f1-score   support

           0       0.79      0.70      0.74       380
           1       0.73      0.81      0.77       371

    accuracy                           0.75       751
   macro avg       0.76      0.76      0.75       751
weighted avg       0.76      0.75      0.75       751
```

In [28]:

```
from sklearn.neighbors import KNeighborsClassifier
```

In [29]:

```
knn = KNeighborsClassifier(n_neighbors = 3)
```

In [30]:

```
knn.fit(X_train, y_train)
```

Out[30]:

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=3, p=2,
                     weights='uniform')
```

In [31]:

```
pre1 = knn.predict(X_test)
```

In [32]:

```
print(classification_report(pre1, y_test))
```

```
              precision    recall  f1-score   support

           0       0.72      0.72      0.72       337
           1       0.77      0.77      0.77       414

    accuracy                           0.75       751
   macro avg       0.75      0.75      0.75       751
weighted avg       0.75      0.75      0.75       751
```

In [33]:

```
from xgboost import XGBClassifier
```

```
C:\Users\SURYA\Anaconda3\lib\site-packages\distributed\utils.py:133: Runtime
Warning: Couldn't detect a suitable IP address for reaching '8.8.8.8', defau
lting to '127.0.0.1': [WinError 10065] A socket operation was attempted to a
n unreachable host
  RuntimeWarning,
```

In [34]:

```
pip install xgboost
```

```
Requirement already satisfied: xgboost in c:\users\surya\anaconda3\lib\site-
packages (1.0.1)
Requirement already satisfied: scipy in c:\users\surya\anaconda3\lib\site-pa
ckages (from xgboost) (1.3.1)
Requirement already satisfied: numpy in c:\users\surya\anaconda3\lib\site-pa
ckages (from xgboost) (1.16.5)
Note: you may need to restart the kernel to use updated packages.
```

In [35]:

```
from xgboost import XGBClassifier
```

In [36]:

```
xg = XGBClassifier()
```

In [37]:

```
xg.fit(X_train, y_train)
```

Out[37]:

```
XGBClassifier(base_score=0.5, booster=None, colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
              importance_type='gain', interaction_constraints=None,
              learning_rate=0.300000012, max_delta_step=0, max_depth=6,
              min_child_weight=1, missing=nan, monotone_constraints=None,
              n_estimators=100, n_jobs=0, num_parallel_tree=1,
              objective='binary:logistic', random_state=0, reg_alpha=0,
              reg_lambda=1, scale_pos_weight=1, subsample=1, tree_method=Non
e,
              validate_parameters=False, verbosity=None)
```

In [38]:

```
pre2 = xg.predict(X_test)
```

In [39]:

```
print(classification_report(pre2, y_test))
```

```
              precision    recall  f1-score   support

           0       0.77      0.76      0.77       340
           1       0.81      0.82      0.81       411

    accuracy                           0.79       751
   macro avg       0.79      0.79      0.79       751
weighted avg       0.79      0.79      0.79       751
```

In [1]:

```
conda install nbconvert
```

```
Collecting package metadata (current_repodata.json): ...working... done
Solving environment: ...working... done

## Package Plan ##

  environment location: C:\Users\SURYA\Anaconda3

  added / updated specs:
    - nbconvert


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    conda-4.8.2                |           py37_0         2.8 MB
    ------------------------------------------------------------
                                           Total:         2.8 MB

The following packages will be UPDATED:

  conda                                        4.7.12-py37_0 --> 4.8.2-py37_0



Downloading and Extracting Packages

conda-4.8.2          | 2.8 MB    |            |   0% 
conda-4.8.2          | 2.8 MB    |            |   1% 
conda-4.8.2          | 2.8 MB    | 2          |   3% 
conda-4.8.2          | 2.8 MB    | 4          |   4% 
conda-4.8.2          | 2.8 MB    | 7          |   8% 
conda-4.8.2          | 2.8 MB    | 8          |   9% 
conda-4.8.2          | 2.8 MB    | #          |  10% 
conda-4.8.2          | 2.8 MB    | #1         |  12% 
conda-4.8.2          | 2.8 MB    | #2         |  13% 
conda-4.8.2          | 2.8 MB    | #3         |  14% 
conda-4.8.2          | 2.8 MB    | #4         |  15% 
conda-4.8.2          | 2.8 MB    | #5         |  16% 
conda-4.8.2          | 2.8 MB    | ##8        |  28% 
conda-4.8.2          | 2.8 MB    | ###2       |  33% 
conda-4.8.2          | 2.8 MB    | ###5       |  36% 
conda-4.8.2          | 2.8 MB    | ###8       |  39% 
conda-4.8.2          | 2.8 MB    | ####1      |  41% 
conda-4.8.2          | 2.8 MB    | #####4     |  55% 
conda-4.8.2          | 2.8 MB    | #####8     |  59% 
conda-4.8.2          | 2.8 MB    | ######2    |  63% 
conda-4.8.2          | 2.8 MB    | ######6    |  66% 
conda-4.8.2          | 2.8 MB    | ######8    |  69% 
conda-4.8.2          | 2.8 MB    | #######1   |  71% 
conda-4.8.2          | 2.8 MB    | #######2   |  73% 
conda-4.8.2          | 2.8 MB    | #######4   |  74% 
conda-4.8.2          | 2.8 MB    | #######6   |  76% 
conda-4.8.2          | 2.8 MB    | #######7   |  78% 
conda-4.8.2          | 2.8 MB    | #######9   |  79% 
conda-4.8.2          | 2.8 MB    | ########   |  81% 
conda-4.8.2          | 2.8 MB    | ########3  |  83% 
```

```
conda-4.8.2          | 2.8 MB    | ########4  |  85% 
conda-4.8.2          | 2.8 MB    | ########5  |  86% 
conda-4.8.2          | 2.8 MB    | ########7  |  87% 
conda-4.8.2          | 2.8 MB    | ########8  |  88% 
conda-4.8.2          | 2.8 MB    | ########9  |  90% 
conda-4.8.2          | 2.8 MB    | #########1 |  91% 
conda-4.8.2          | 2.8 MB    | #########2 |  93% 
conda-4.8.2          | 2.8 MB    | #########3 |  94% 
conda-4.8.2          | 2.8 MB    | #########5 |  95% 
conda-4.8.2          | 2.8 MB    | #########6 |  96% 
conda-4.8.2          | 2.8 MB    | #########7 |  98% 
conda-4.8.2          | 2.8 MB    | #########8 |  99% 
conda-4.8.2          | 2.8 MB    | #########9 | 100% 
conda-4.8.2          | 2.8 MB    | ########## | 100% 
Preparing transaction: ...working... done
Verifying transaction: ...working... done
Executing transaction: ...working... done

Note: you may need to restart the kernel to use updated packages.
```

In [2]:

```
jupyter nbconvert --to pdf MyNotebook.ipynb
```

```
  File "<ipython-input-2-cee3fce6ddd9>", line 1
    jupyter nbconvert --to pdf MyNotebook.ipynb
                    ^
SyntaxError: invalid syntax
```

In [ ]:

```

```