Assignment 1

March 4, 2018

```
In [1]: from IPython.display import HTML
        HTML('''<style>html, body{overflow-y: visible !important} .CodeMirror{min-width:105% !i
```

```
Out[1]: <IPython.core.display.HTML object>
```

1 Foundations of Data Mining: Assignment 1

Please complete all assignments in this notebook. You should submit this notebook, as well as a PDF version (See File > Download as).

```
In [2]: # Please fill in your names here
        NAME_STUDENT_1 = "Bram van der Pol"
        NAME_STUDENT_2 = "Joris van der Heijden"
```

```
In [3]: %matplotlib inline
        from preamble import *
        plt.rcParams['savefig.dpi'] = 200 # This controls the size of your figures
        # Comment out and restart notebook if you only want the last output of each cell.
        InteractiveShell.ast_node_interactivity = "all"
```

1.1 MoneyBall (5 points, 1+2+1+1)

In the early 2000s, 2 baseball scouts completely changed the game of baseball by analysing the available data about baseball players and hiring the best ones. The MoneyBall dataset contains this data (click the link for more details). The goal is to accurately predict the number of 'runs' each player can score.

```
In [4]: moneyball = oml.datasets.get_dataset(41021); # Download MoneyBall data
        # Get the predictors X and the target y
        X, y, attribute_names = moneyball.get_data(target=moneyball.default_target_attribute, r
        # Describe the data with pandas, just to get an overview
        ballframe = pd.DataFrame(X, columns=attribute_names);
        ballframe.describe();

        import matplotlib;
        #n, bins, patches = ax.hist(X, num_bins, normed=1)
```

```
for index in range(len(X.T)):
    #if (index < 8):
    print(index);
    column=X[:,index];
    Filtered_column1=column[~np.isnan(column)];

    Filtered_column2 = [np.nan if np.isnan(x) else x for x in column];

    fig1 = plt.figure(); #Generate new figure
    matplotlib.pyplot.subplot(1,2,1);
    matplotlib.pyplot.hist(Filtered_column1);
    matplotlib.pyplot.title('Histogram',fontweight='bold',fontsize=15);


    matplotlib.pyplot.subplot(1,2,2);
    matplotlib.pyplot.scatter(Filtered_column2,y);
    matplotlib.pyplot.title('Scatter',fontweight='bold',fontsize=15);
    matplotlib.pyplot.xlabel('X');
    matplotlib.pyplot.ylabel('y');

    matplotlib.pyplot.suptitle(attribute_names[index]+" Index "+str(index) ,fontweight=
    matplotlib.pyplot.subplots_adjust(left=0.2,wspace=0.8,top=0.8);
```

Out[4]:

	Team	League	Year	RA	...	RankPlayoffs	G	\
count	1232.00	1232.0	1232.00	1232.00	...	244.00	1232.00	
mean	15.67	0.5	1988.96	715.08	...	1.72	3.92	
std	9.72	0.5	14.82	93.08	...	1.10	0.62	
min	0.00	0.0	1962.00	472.00	...	0.00	0.00	
25%	7.00	0.0	1976.75	649.75	...	1.00	4.00	
50%	16.00	0.5	1989.00	709.00	...	2.00	4.00	
75%	23.00	1.0	2002.00	774.25	...	3.00	4.00	
max	38.00	1.0	2012.00	1103.00	...	4.00	7.00	

	OOBP	OSLG
count	420.00	420.00
mean	0.33	0.42
std	0.02	0.03
min	0.29	0.35
25%	0.32	0.40
50%	0.33	0.42
75%	0.34	0.44
max	0.38	0.50

[8 rows x 14 columns]

```
0
```

Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a38459c4a8>

```
Out[4]: (array([152., 188., 152.,  98., 141., 175., 132.,  95.,  78.,  21.]),
         array([ 0. ,  3.8,  7.6, 11.4, 15.2, 19. , 22.8, 26.6, 30.4, 34.2, 38. ]),
         <a list of 10 Patch objects>)

Out[4]: Text(0.5,1,'Histogram')

Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a3846a9550>

Out[4]: <matplotlib.collections.PathCollection at 0x1a3846eb518>

Out[4]: Text(0.5,1,'Scatter')

Out[4]: Text(0.5,0,'X')

Out[4]: Text(0,0.5,'y')

Out[4]: Text(0.5,0.98,'Team Index 0')

1


Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a3845d3828>

Out[4]: (array([616.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0., 616.]),
         array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
         <a list of 10 Patch objects>)

Out[4]: Text(0.5,1,'Histogram')

Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a384980f60>

Out[4]: <matplotlib.collections.PathCollection at 0x1a3849bd9b0>

Out[4]: Text(0.5,1,'Scatter')

Out[4]: Text(0.5,0,'X')

Out[4]: Text(0,0.5,'y')

Out[4]: Text(0.5,0.98,'League Index 1')

2


Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a38471d518>

Out[4]: (array([100., 112.,  96., 104., 130., 130.,  82., 148., 150., 180.]),
         array([1962., 1967., 1972., 1977., 1982., 1987., 1992., 1997., 2002.,
                2007., 2012.]),
         <a list of 10 Patch objects>)

Out[4]: Text(0.5,1,'Histogram')
```

```
Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a384a022b0>

Out[4]: <matplotlib.collections.PathCollection at 0x1a384a42b38>

Out[4]: Text(0.5,1,'Scatter')

Out[4]: Text(0.5,0,'X')

Out[4]: Text(0,0.5,'y')

Out[4]: Text(0.5,0.98,'Year Index 2')

3


Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a3849d0668>

Out[4]: (array([ 20., 110., 239., 327., 270., 163.,  77.,  24.,   1.,   1.]),
         array([ 472. ,  535.1,  598.2,  661.3,  724.4,  787.5,  850.6,  913.7,
                 976.8, 1039.9, 1103. ]),
         <a list of 10 Patch objects>)

Out[4]: Text(0.5,1,'Histogram')

Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a384a885c0>

Out[4]: <matplotlib.collections.PathCollection at 0x1a384ada208>

Out[4]: Text(0.5,1,'Scatter')

Out[4]: Text(0.5,0,'X')

Out[4]: Text(0,0.5,'y')

Out[4]: Text(0.5,0.98,'RA Index 3')

4


Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a384ada710>

Out[4]: (array([  2.,  17.,  50., 167., 243., 291., 286., 134.,  39.,   3.]),
         array([ 40. ,  47.6,  55.2,  62.8,  70.4,  78. ,  85.6,  93.2, 100.8,
                108.4, 116. ]),
         <a list of 10 Patch objects>)

Out[4]: Text(0.5,1,'Histogram')

Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a384b03828>

Out[4]: <matplotlib.collections.PathCollection at 0x1a384b616d8>
```

```
Out[4]: Text(0.5,1,'Scatter')

Out[4]: Text(0.5,0,'X')

Out[4]: Text(0,0.5,'y')

Out[4]: Text(0.5,0.98,'W Index 4')

5


Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a384ae8278>

Out[4]: (array([  7.,  22.,  64., 193., 322., 266., 225.,  81.,  45.,   7.]),
         array([0.277, 0.287, 0.296, 0.306, 0.315, 0.325, 0.335, 0.344, 0.354,
                0.363, 0.373]),
         <a list of 10 Patch objects>)

Out[4]: Text(0.5,1,'Histogram')

Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a384ba3588>

Out[4]: <matplotlib.collections.PathCollection at 0x1a384bf1828>

Out[4]: Text(0.5,1,'Scatter')

Out[4]: Text(0.5,0,'X')

Out[4]: Text(0,0.5,'y')

Out[4]: Text(0.5,0.98,'OBP Index 5')

6


Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a384b69400>

Out[4]: (array([ 13.,  31., 100., 186., 276., 258., 202.,  93.,  57.,  16.]),
         array([0.301, 0.32 , 0.339, 0.358, 0.377, 0.396, 0.415, 0.434, 0.453,
                0.472, 0.491]),
         <a list of 10 Patch objects>)

Out[4]: Text(0.5,1,'Histogram')

Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a384c2e240>

Out[4]: <matplotlib.collections.PathCollection at 0x1a384c7b588>

Out[4]: Text(0.5,1,'Scatter')

Out[4]: Text(0.5,0,'X')
```

```
Out[4]: Text(0,0.5,'y')

Out[4]: Text(0.5,0.98,'SLG Index 6')

7


Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a384bf8198>

Out[4]: (array([  4.,  14.,  41., 102., 242., 329., 229., 186.,  58.,  27.]),
         array([0.214, 0.222, 0.23 , 0.238, 0.246, 0.254, 0.262, 0.27 , 0.278,
                0.286, 0.294]),
         <a list of 10 Patch objects>)

Out[4]: Text(0.5,1,'Histogram')

Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a384cbc3c8>

Out[4]: <matplotlib.collections.PathCollection at 0x1a384c4b6a0>

Out[4]: Text(0.5,1,'Scatter')

Out[4]: Text(0.5,0,'X')

Out[4]: Text(0,0.5,'y')

Out[4]: Text(0.5,0.98,'BA Index 7')

8


Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a384c861d0>

Out[4]: (array([988.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0., 244.]),
         array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
         <a list of 10 Patch objects>)

Out[4]: Text(0.5,1,'Histogram')

Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a384d48550>

Out[4]: <matplotlib.collections.PathCollection at 0x1a384d958d0>

Out[4]: Text(0.5,1,'Scatter')

Out[4]: Text(0.5,0,'X')

Out[4]: Text(0,0.5,'y')

Out[4]: Text(0.5,0.98,'Playoffs Index 8')
```

```
9
```

```
Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a384d0e588>
```

```
Out[4]: (array([52., 53., 44.,  0., 44., 21.,  0., 20.,  9.,  1.]),
         array([0. , 0.7, 1.4, 2.1, 2.8, 3.5, 4.2, 4.9, 5.6, 6.3, 7. ]),
         <a list of 10 Patch objects>)
```

```
Out[4]: Text(0.5,1,'Histogram')
```

```
Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a384dd1978>
```

```
Out[4]: <matplotlib.collections.PathCollection at 0x1a384e175f8>
```

```
Out[4]: Text(0.5,1,'Scatter')
```

```
Out[4]: Text(0.5,0,'X')
```

```
Out[4]: Text(0,0.5,'y')
```

```
Out[4]: Text(0.5,0.98,'RankSeason Index 9')
```

```
10
```

```
Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a384e20160>
```

```
Out[4]: (array([47.,  0., 47.,  0.,  0., 80.,  0., 68.,  0.,  2.]),
         array([0. , 0.4, 0.8, 1.2, 1.6, 2. , 2.4, 2.8, 3.2, 3.6, 4. ]),
         <a list of 10 Patch objects>)
```

```
Out[4]: Text(0.5,1,'Histogram')
```

```
Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a384e5b550>
```

```
Out[4]: <matplotlib.collections.PathCollection at 0x1a384eab780>
```

```
Out[4]: Text(0.5,1,'Scatter')
```

```
Out[4]: Text(0.5,0,'X')
```

```
Out[4]: Text(0,0.5,'y')
```

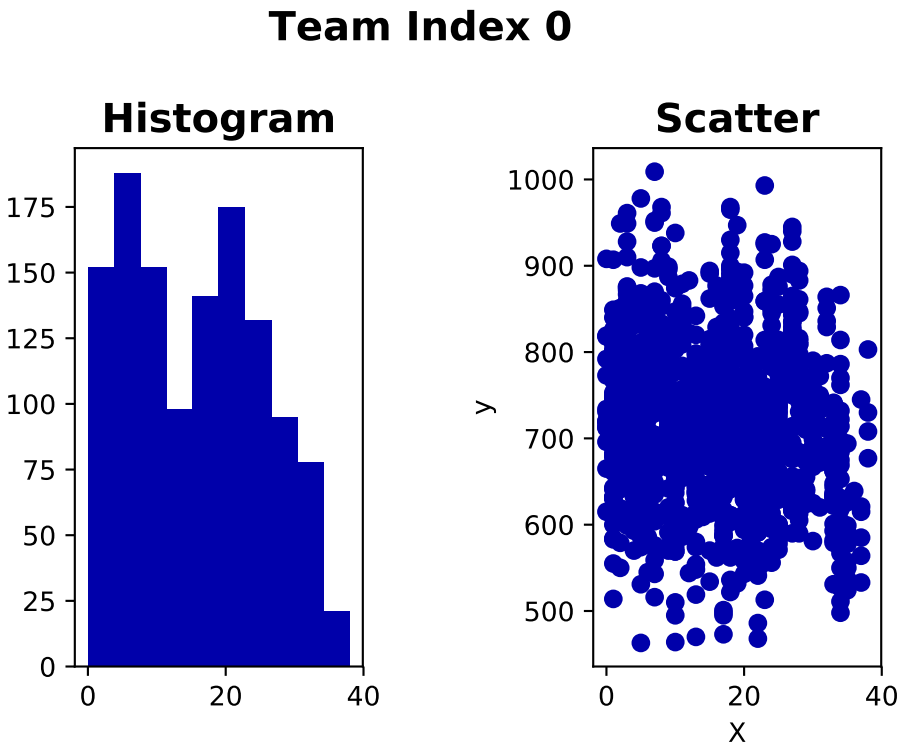
```
Out[4]: Text(0.5,0.98,'RankPlayoffs Index 10')
```

```
11
```

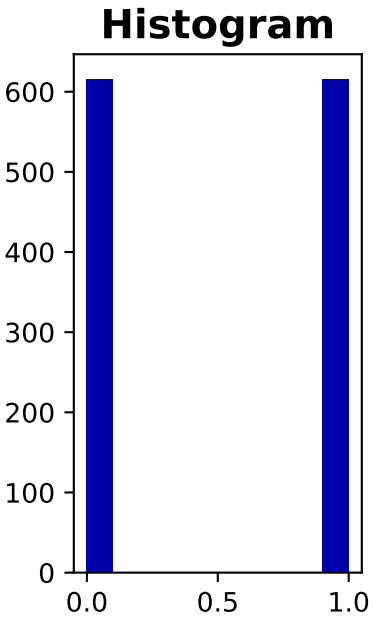
```
Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a384e243c8>
```

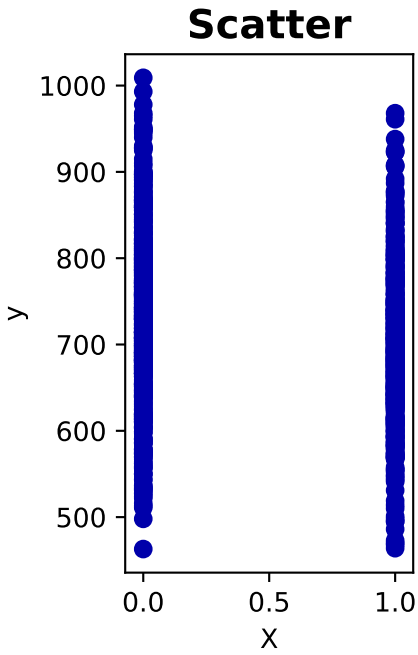
```
Out[4]: (array([  1.,  10.,  23.,   0., 139., 954.,   0.,  93.,  10.,   2.]),
         array([0. , 0.7, 1.4, 2.1, 2.8, 3.5, 4.2, 4.9, 5.6, 6.3, 7. ]),
         <a list of 10 Patch objects>)

Out[4]: Text(0.5,1,'Histogram')

Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a384eea198>

Out[4]: <matplotlib.collections.PathCollection at 0x1a384f39518>

Out[4]: Text(0.5,1,'Scatter')

Out[4]: Text(0.5,0,'X')

Out[4]: Text(0,0.5,'y')

Out[4]: Text(0.5,0.98,'G Index 11')

12


Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a384eb01d0>

Out[4]: (array([ 4., 33., 64., 89., 94., 74., 41., 15.,  5.,  1.]),
         array([0.294, 0.303, 0.312, 0.321, 0.33 , 0.339, 0.348, 0.357, 0.366,
                0.375, 0.384]),
         <a list of 10 Patch objects>)

Out[4]: Text(0.5,1,'Histogram')

Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a384f76358>

Out[4]: <matplotlib.collections.PathCollection at 0x1a384fc37f0>

Out[4]: Text(0.5,1,'Scatter')

Out[4]: Text(0.5,0,'X')

Out[4]: Text(0,0.5,'y')

Out[4]: Text(0.5,0.98,'OOBP Index 12')

13


Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a384f94a20>

Out[4]: (array([ 6., 14., 36., 92., 82., 82., 61., 31., 14.,  2.]),
         array([0.346, 0.361, 0.377, 0.392, 0.407, 0.422, 0.438, 0.453, 0.468,
                0.484, 0.499]),
         <a list of 10 Patch objects>)
```

```
Out[4]: Text(0.5,1,'Histogram')

Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1a3850080b8>

Out[4]: <matplotlib.collections.PathCollection at 0x1a38504d4a8>

Out[4]: Text(0.5,1,'Scatter')

Out[4]: Text(0.5,0,'X')

Out[4]: Text(0,0.5,'y')

Out[4]: Text(0.5,0.98,'OSLG Index 13')
```

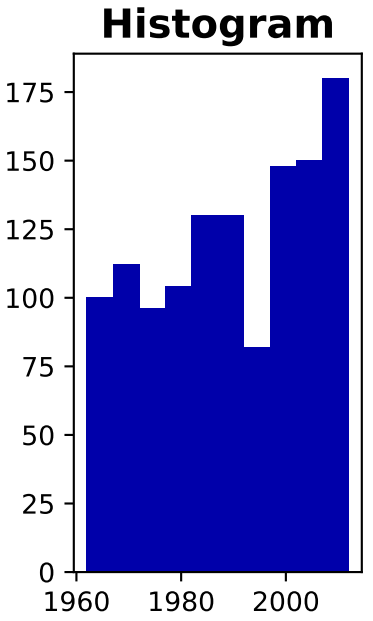


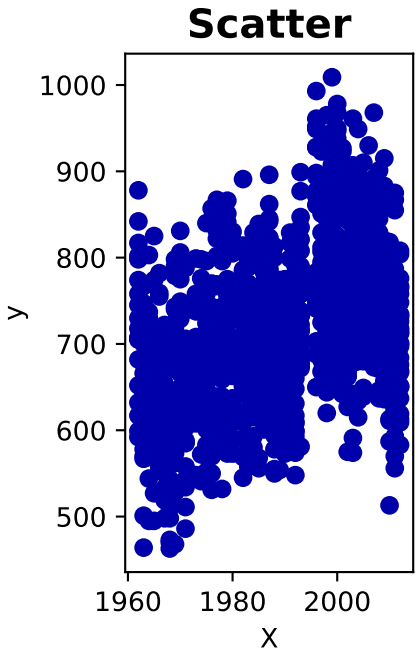
League Index 1



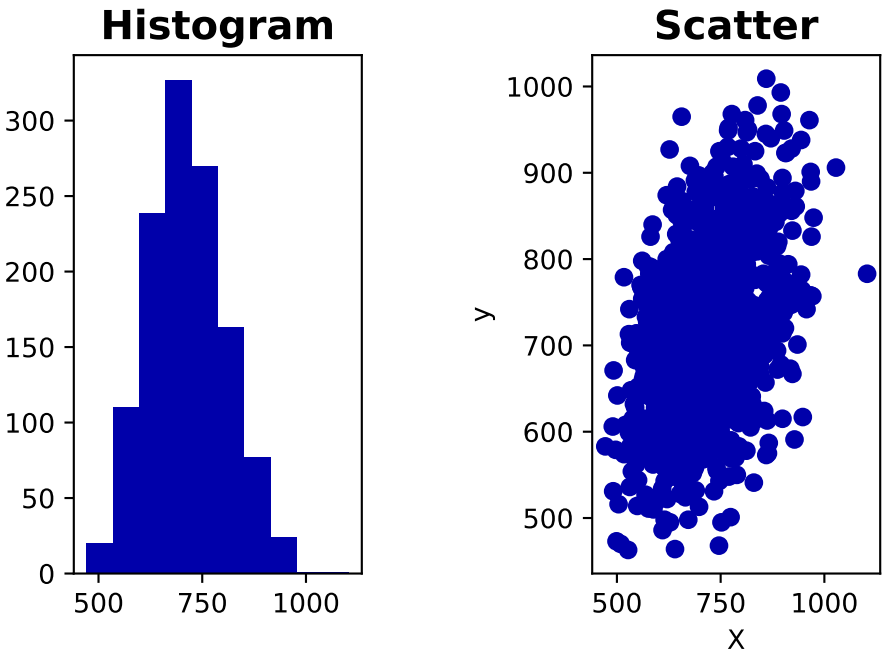


Year Index 2

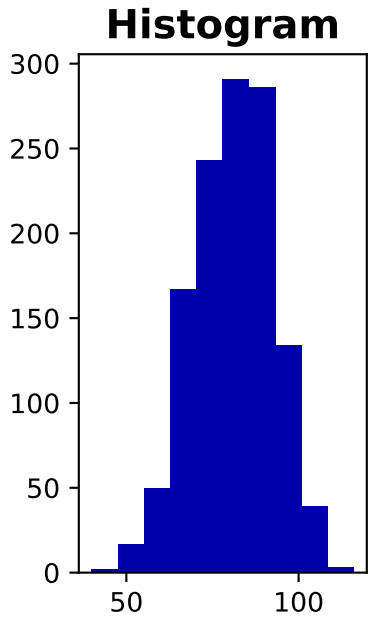


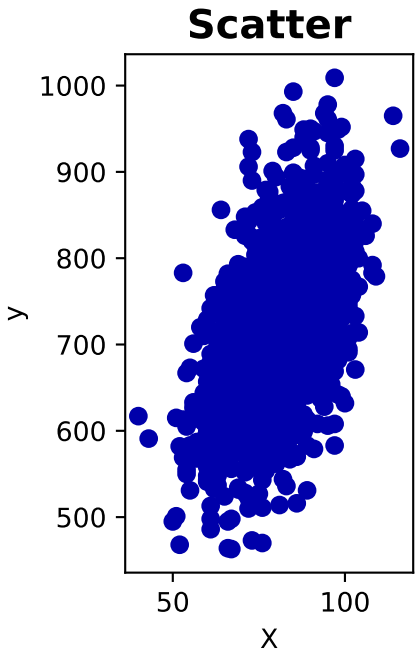


RA Index 3

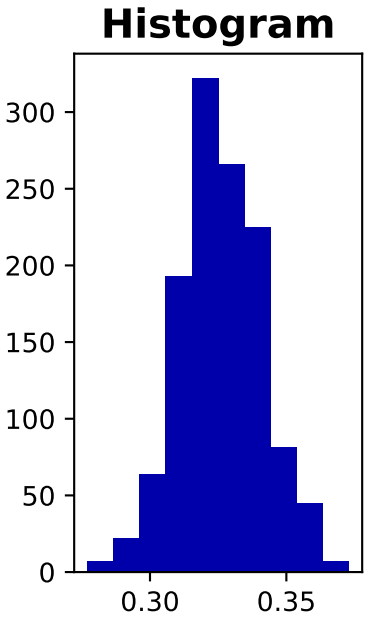

Histogram
300
250
200
150
100
50
0
500
750
1000
Scatter
1000
900
800
700
600
500
y
500
750
1000
X

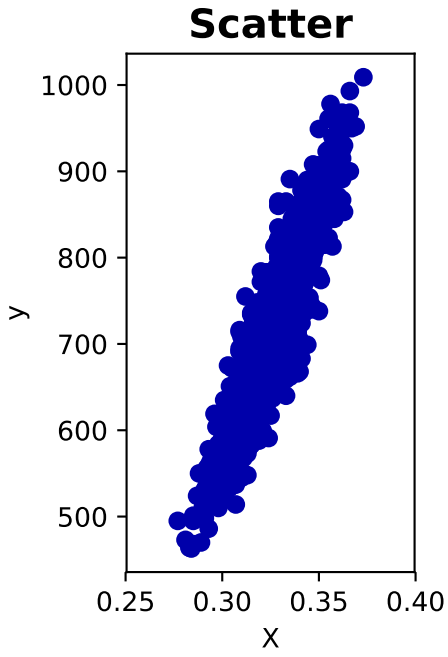
W Index 4



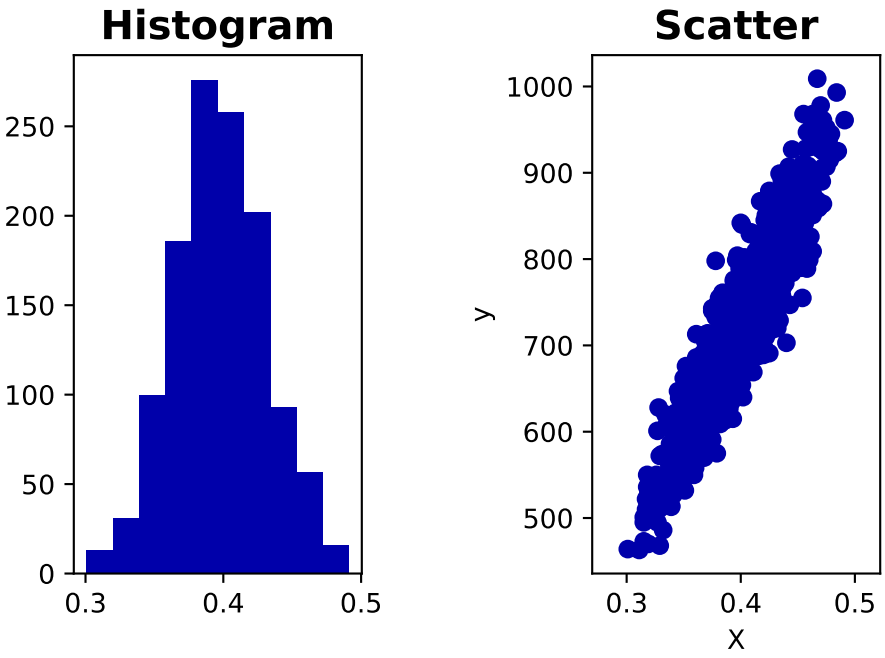


OBP Index 5

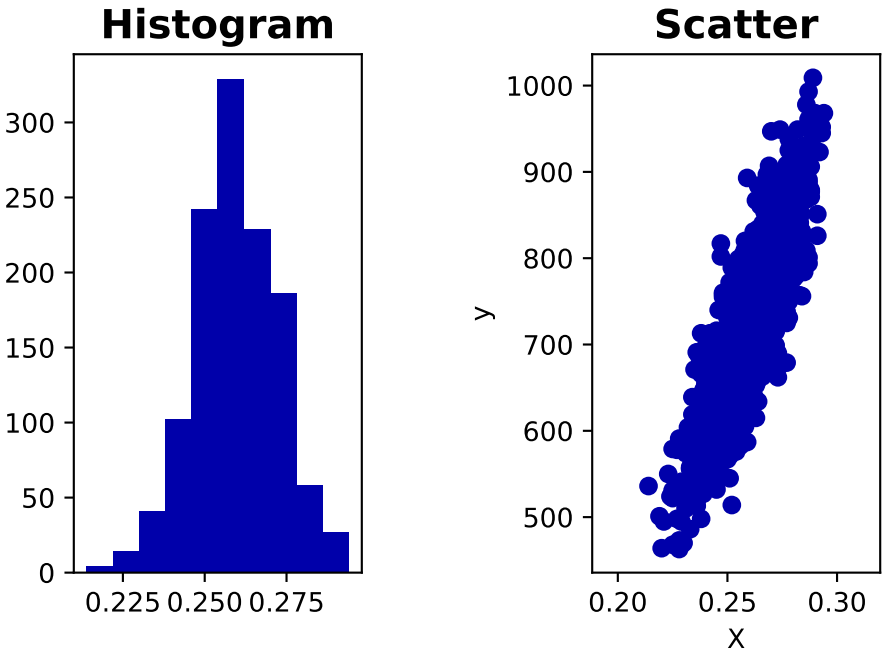




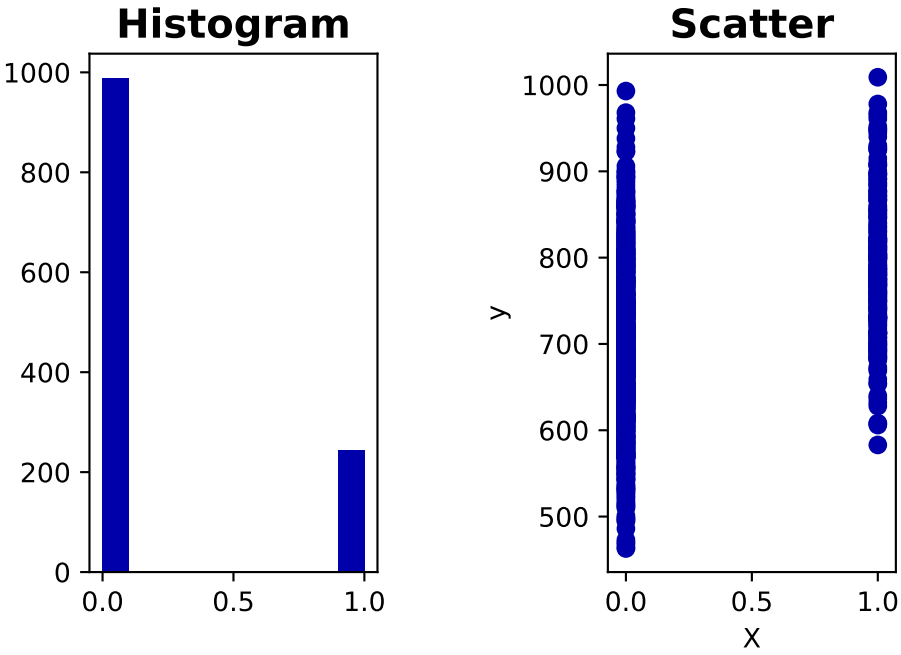
SLG Index 6


Histogram
250
200
150
100
50
0
0.3
0.4
0.5
Scatter
1000
900
800
700
600
500
y
0.3
0.4
0.5
X

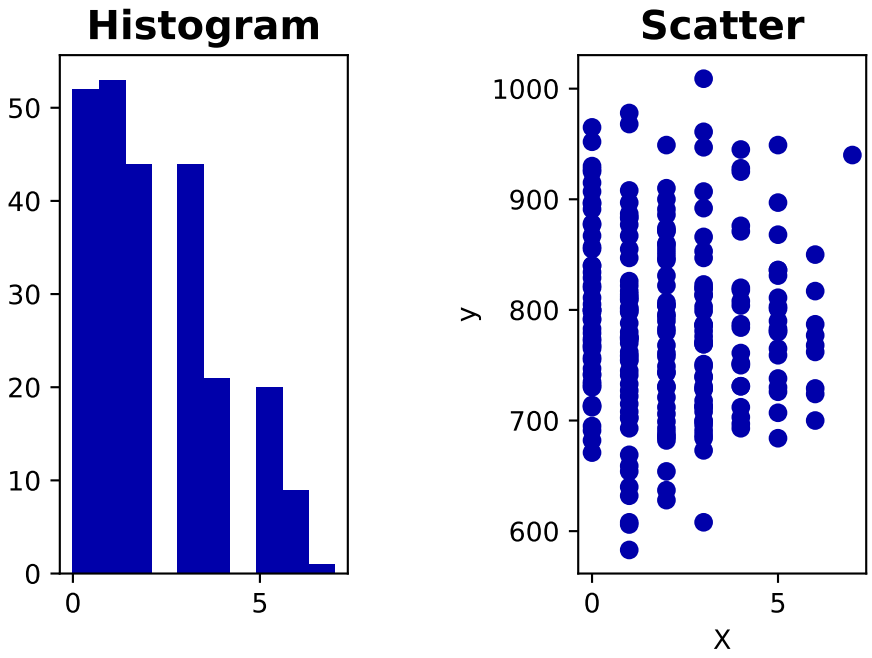
BA Index 7



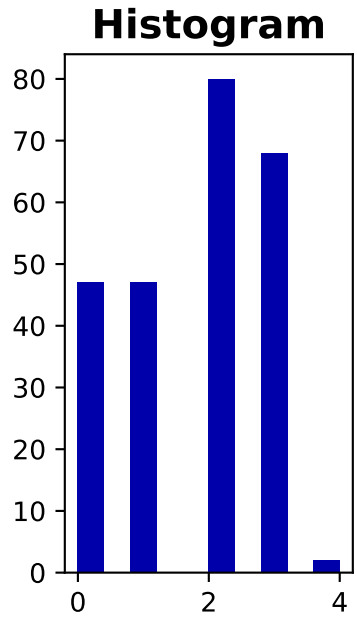
Playoffs Index 8

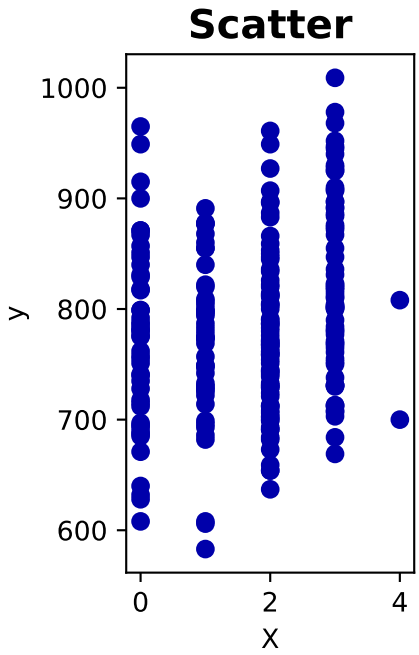


RankSeason Index 9

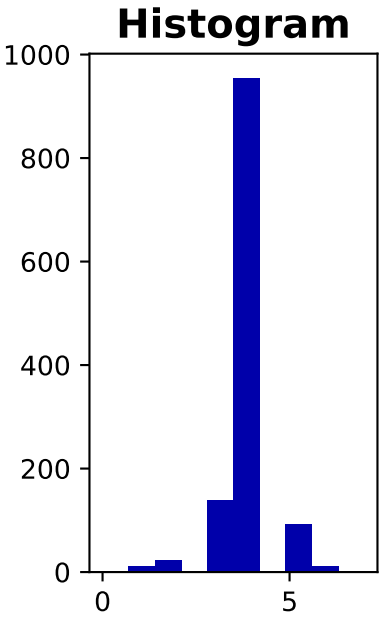


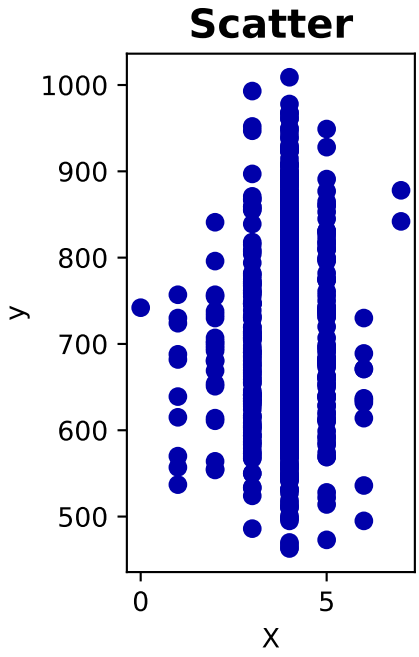
RankPlayoffs Index 10



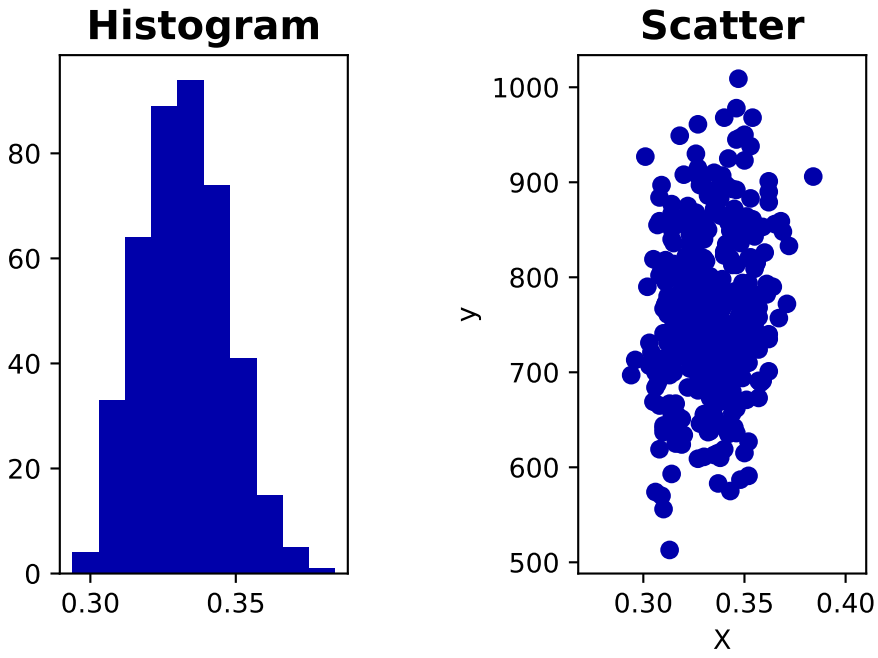


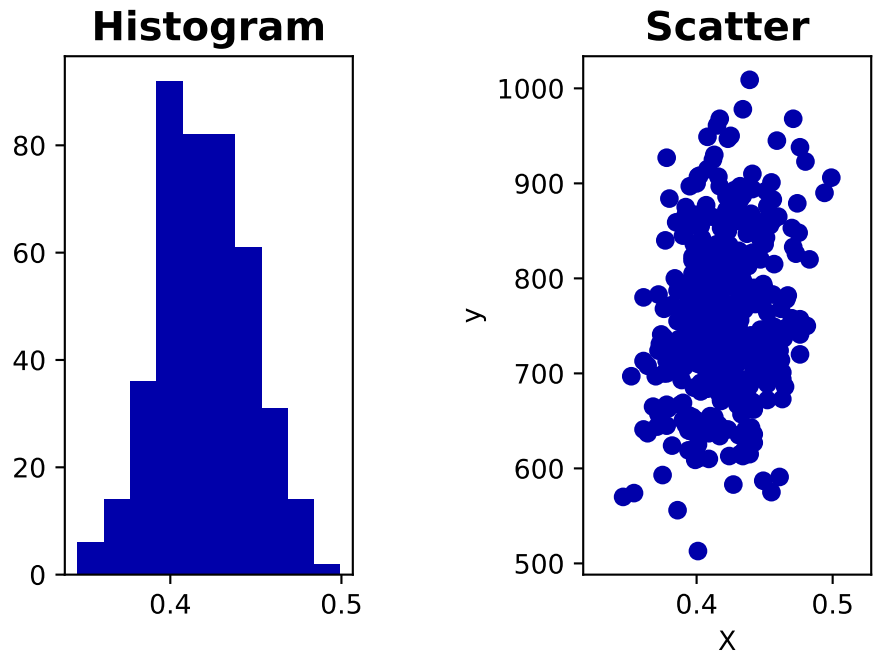
G Index 11





OOBP Index 12






```
In [5]: '''
        1 . Visually explore the data. Plot the distribution of each feature (e.g. histograms)
        - Feel free to create additional plots that help you understand the data
        - Only visualize the data, you don't need to change it (yet)

        **Answers:**
        The first column of figures shows the histograms of each column of the data in X.
        The second column shows the relationship between y and x.
        *Is there anything that stands out?*
        The dataset containts non real numbers, so the data set must first be cleaned to visua
        Beside the NaN the data the figures that do not show a clear distrubution are: 1, 8, 1

        Is there something that you think might require special treatment?
        '''

Out[5]: "\n1 . Visually explore the data. Plot the distribution of each feature (e.g. histogra
```

2 . Compare all linear regression algorithms that we covered in class (Linear Regression, Ridge, Lasso and ElasticNet), as well as kNN. Evaluate using cross-validation and the R^2 score, with the default parameters. Does scaling the data with StandardScaler help? Provide a concise but meaningful interpretation of the results. - Preprocess the data as needed (e.g. are there nominal features that are not ordinal?). If you don't know how to proceed, remove the feature and continue.

3 . Do a default, shuffled train-test split and optimize the linear models for the degree of regularization ($alpha$) and choice of penalty (L1/L2). For Ridge and Lasso, plot a curve showing the effect of the training and test set performance (R^2) while increasing the degree of regularization for different penalties. For ElasticNet, plot a heatmap $alpha \times l1\_ratio \rightarrow R^2$ using test set performance. Report the optimal performance. Again, provide a concise but meaningful interpretation. What does the regularization do? Can you get better results? - Think about how you get the L1/L2 loss. This is not a hyperparameter in regression. - We've seen how to generate such heatmaps in Lecture 3.

4 . Visualize the coefficients of the optimized models. Do they agree on which features are important? Compare the results with the feature importances returned by a RandomForest. Does it agree with the linear models? What would look for when scouting for a baseball player?

1.2 Nepalese character recognition (5 points, 1+2+2)

The Devnagari-Script dataset contains 92,000 images (32x32 pixels) of 46 characters from Devanagari script. Your goal is to learn to recognize the right letter given the image.

In [6]:
```
# Initial the setting so the code can be run from this point
from IPython.display import HTML
HTML('''<style>html, body{overflow-y: visible !important} .CodeMirror{min-width:105% !
%matplotlib inline
from preamble import *
plt.rcParams['savefig.dpi'] = 200 # This controls the size of your figures
# Comment out and restart notebook if you only want the last output of each cell.
InteractiveShell.ast_node_interactivity = "all"
```

Out[6]:
```
<IPython.core.display.HTML object>
```

In [7]:
```
devnagari = oml.datasets.get_dataset(40923) # Download Devnagari data
# Get the predictors X and the labels y
X, y = devnagari.get_data(target=devnagari.default_target_attribute);
if not 'classes' in locals():
    classes = devnagari.retrieve_class_labels(target_name='character') # This one take
```

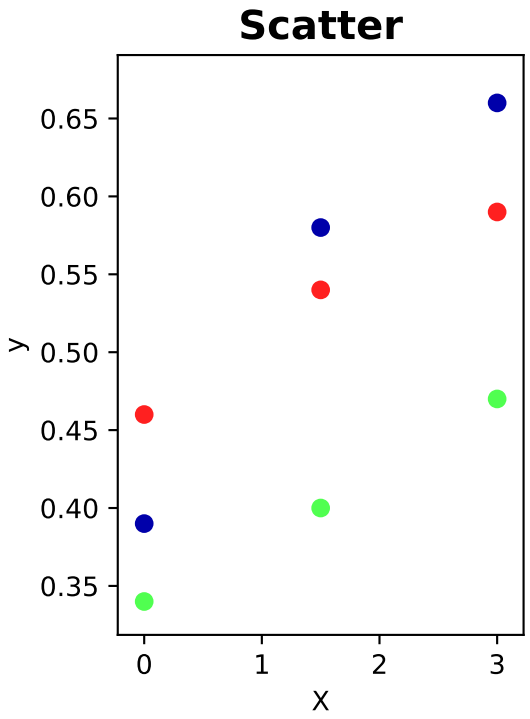
In [8]:
```
from random import randint
# Take some random examples, reshape to a 32x32 image and plot
fig, axes = plt.subplots(1, 5,  figsize=(10, 5))
for i in range(5):
    n = randint(0,90000)
    axes[i].imshow(X[n].reshape(32, 32), cmap=plt.cm.gray_r)
    axes[i].set_xlabel("Class: %s" % (classes[y[n]]))
plt.show();
```

1. Evaluate k-Nearest Neighbors, Logistic Regression and RandomForests with their default settings.
    - Take a stratified 10% subsample of the data.
    - Use the default train-test split and predictive accuracy. Is predictive accuracy a good scoring measure for this problem?
    - Try to build the same models on increasingly large samples of the dataset (e.g. 10%, 20%,...). Plot the training time and the predictive performance for each. Stop when the training time becomes prohibitively large (this will be different for different models).

In [9]:
```
# Import the functions (standard variables are used)
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report, confusion_matrix
import time
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import KFold
import matplotlib.pyplot as plt

def Question1():

    length_data=len(y); #92000

    number_of_test=100; # To have more control of the computation time
    number_of_training=1000;

    Fraction=np.array([0.01,0.02,0.03])
    A = np.zeros((len(Fraction), 3)) # Accuracy Matrix
    T = np.zeros((len(Fraction),3))  # Computation time Matrix

    # Split the data in test data and train data (stratified 10% subsample)
    from sklearn.model_selection import train_test_split

    #n1 = randint(0,len(y_test1))
    #n2 = randint(0,len(y_test1))

    for i in range(len(Fraction)):
        X_del, X_train, y_del, y_train = train_test_split(X, y, test_size=Fraction[i])
        X_del, X_test, y_del, y_test = train_test_split(X, y, test_size=number_of_test,

        print(X_test.shape)
        print(X_train.shape)

        # Solve the learning problems if the solutions do not exist yet
```

```
        #if not 'classifier' in locals():
        tic = time.clock()
        knn = KNeighborsClassifier(n_neighbors=5)
        knn.fit(X_train, y_train)
        toc = time.clock()
        y_predKNearest = knn.predict(X_test)
        Accuracy_KNearest=accuracy_score(y_test,y_predKNearest)
        A[i][0]=Accuracy_KNearest
        T[i][0]=toc-tic

        mglearn

        #if not 'logistic' in locals():
        tic = time.clock()
        logistic = LogisticRegression()
        logistic.fit(X_train,y_train)
        toc = time.clock()
        y_predLogistic = logistic.predict(X_test)
        Accuracy_Logistic=accuracy_score(y_test,y_predLogistic)
        A[i][1]=Accuracy_Logistic
        T[i][1]=toc-tic

        #if not 'Forest' in locals():
        tic = time.clock()
        Forest = RandomForestClassifier()
        Forest.fit(X_train,y_train)
        toc = time.clock()
        y_predForest = Forest.predict(X_test)
        Accuracy_Forest=accuracy_score(y_test,y_predForest)
        A[i][2]=Accuracy_Forest
        T[i][2]=toc-tic

        xas=np.linspace(0,len(A[:,0]),len(A[:,0]));

    plt.subplot(1,2,2);
    plt.scatter(xas,A[:,0]);
    plt.scatter(xas,A[:,1]);
    plt.scatter(xas,A[:,2]);
    plt.title('Scatter',fontweight='bold',fontsize=15);
    plt.xlabel('X');
    plt.ylabel('y');
    plt.show()
    print(A)
    print(T)
    return

Question1()
```

```
(100, 1024)
(920, 1024)
(100, 1024)
(1840, 1024)
(100, 1024)
(2760, 1024)
```




```
[[0.39 0.46 0.34]
 [0.58 0.54 0.4 ]
 [0.66 0.59 0.47]]
[[ 0.041  3.641  0.118]
 [ 0.128 11.147  0.222]
 [ 0.216 23.627  0.325]]
```

2 . Optimize the value for the number of neighbors k (keep $k < 50$) and the number of trees (keep *n_estimators* < 100) on the stratified 10% subsample. Use 10-fold crossvalidation and plot k and *n_estimators* against the predictive accuracy. Which value of k, *n_estimators* should you pick?

```
In [ ]: from sklearn.model_selection import cross_val_score

        def Question2():
```

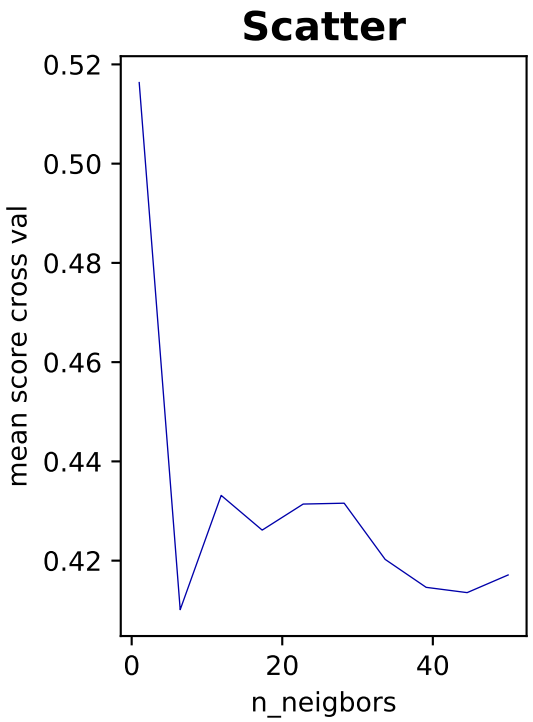
```
#    length_data=len(y); #92000

    number_of_test=100; # To have more control of the computation time

    Fraction=np.array([0.01])

    n_neighborsA=np.linspace(1,50,10)
    n_estimatorsA=np.linspace(1,100,10)

    A1 = np.zeros((len(n_neighborsA), 3)) # Accuracy Matrix
    T1 = np.zeros((len(n_neighborsA),3))  # Computation time Matrix

    A2 = np.zeros((len(n_estimatorsA), 3)) # Accuracy Matrix
    T2 = np.zeros((len(n_estimatorsA),3))  # Computation time Matrix

    # Split the data in test data and train data (stratified 10% subsample)
    from sklearn.model_selection import train_test_split

    X_del, X_train, y_del, y_train = train_test_split(X, y, test_size = Fraction[0])
    X_del, X_test, y_del, y_test = train_test_split(X, y, test_size = number_of_test/(

    X_train.shape
    y_train.shape

    for i in range(len(n_neighborsA)):

        # Reduce the test data as well to 10 samples
        print("knn iteration: %d " % i)
        # Solve the learning problems if the solutions do not exist yet
        #if not 'classifier' in locals():
        tic = time.clock()
        knn = KNeighborsClassifier(n_neighbors=i+1)
        toc = time.clock()

        scores = cross_val_score(knn, X_train, y_train, cv=10)
        Accuracy_knn=np.mean(scores)

#         Accuracy_KNearest=accuracy_score(y_test,y_predKNearest)
        A1[i][0]=Accuracy_knn
        T1[i][0]=toc-tic

#       if not 'logistic' in locals(): # There is no n_neighbors or # of trees option
        '''
        tic = time.clock()
        logistic = LogisticRegression()
        logistic.fit(X_train,y_train)
        toc = time.clock()
        y_predLogistic = logistic.predict(X_test)
```

```
            Accuracy_Logistic=accuracy_score(y_test,y_predLogistic)
            A[i][1]=Accuracy_Logistic
            T[i][1]=toc-tic
            '''

            #if not 'Forest' in locals():

        plt.subplot(1,2,1);
        plt.plot(n_neighborsA,A1[:,0]);
        plt.title('Scatter',fontweight='bold',fontsize=15);
        plt.xlabel('n_neigbors');
        plt.ylabel('mean score cross val');
        plt.show()

        for i in range(len(n_estimatorsA)):
            print("Random Forest iteration: %d " % i)
            tic = time.clock()
            Forest = RandomForestClassifier(n_estimators=i+1) # Number of trees
            toc = time.clock()

            scores = cross_val_score(Forest, X_train, y_train, cv=10)
            Accuracy_Forest=np.mean(scores)

            A2[i][0]=Accuracy_Forest
            T2[i][0]=toc-tic

        plt.subplot(1,2,2);
        plt.plot(n_estimatorsA,A2[:,0]);
        plt.title('Scatter',fontweight='bold',fontsize=15);
        plt.xlabel('n_estimators');
        plt.ylabel('Mean score cross val');
        plt.show()

        return

    Question2()
```

```
knn iteration: 0


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)


knn iteration: 1


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
```

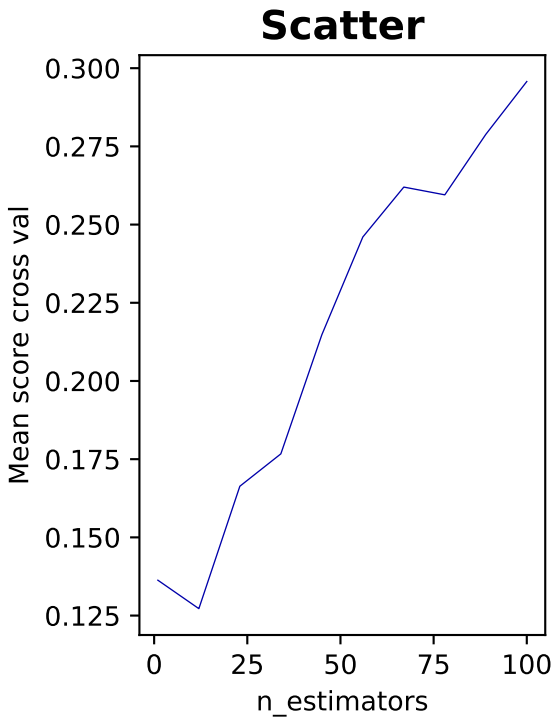
```
knn iteration: 2


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)


knn iteration: 3


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)


knn iteration: 4


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)


knn iteration: 5


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)


knn iteration: 6


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)


knn iteration: 7


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)


knn iteration: 8


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)


knn iteration: 9
```

```
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
```




```
Random Forest iteration: 0
Random Forest iteration: 1


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)


Random Forest iteration: 2


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)


Random Forest iteration: 3
```

```
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)


Random Forest iteration: 4


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)


Random Forest iteration: 5


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)


Random Forest iteration: 6


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)


Random Forest iteration: 7


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)


Random Forest iteration: 8


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)


Random Forest iteration: 9


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
```



3 . For the RandomForest, optimize both *n_estimators* and *max_features* at the same time on the entire dataset. - Use a nested cross-validation and a random search over the possible values, and measure the accuracy. Explore how fine-grained this grid/random search can be, given your computational resources. What is the optimal performance you find? - Hint: choose a nested cross-validation that is feasible. Don't use too many folds in the outer loop. - Repeat the grid search and visualize the results as a plot (heatmap) $n\_estimators \times max\_features \rightarrow ACC$ with ACC visualized as the color of the data point. Try to make the grid as fine as possible. Interpret the results. Can you explain your observations? What did you learn about tuning RandomForests?

```
In [ ]: from sklearn.model_selection import cross_val_score
        from sklearn.model_selection import GridSearchCV
        from sklearn.pipeline import Pipeline
        from sklearn.pipeline import make_pipeline

        def Question3():

        #    length_data=len(y); #92000

            number_of_test=100; # To have more control of the computation time
            Fraction=np.array([0.01])

            n_estimatorsAF=np.linspace(1,50,20)
            n_estimatorsAFR=np.round(n_estimatorsAF)
            n_estimatorsA=n_estimatorsAFR.astype(int)
```

```
print(n_estimatorsA)
max_featuresAF=np.linspace(1,46,20)
max_featuresAFR=np.round(max_featuresAF)
max_featuresA=max_featuresAFR.astype(int)
print(max_featuresA)

A1 = np.zeros((len(n_estimatorsA),len(max_featuresA),1)) # Accuracy Matrix
T1 = np.zeros((len(n_estimatorsA),len(max_featuresA),1))  # Computation time Matri

# Split the data in test data and train data (stratified 10% subsample)
from sklearn.model_selection import train_test_split

X_del, X_train, y_del, y_train = train_test_split(X, y, test_size = Fraction[0])
X_del, X_test, y_del, y_test = train_test_split(X, y, test_size = number_of_test/(



for i in range(len(n_estimatorsA)):
    print("Random Forest iteration: %d " % i)
    for n in range(len(max_featuresA)):
        tic = time.clock()
        Forest = RandomForestClassifier(n_estimators=n_estimatorsA[i], max_features
        Forest_outer = RandomForestClassifier(n_estimators=n_estimatorsA[i], max_fe
        toc = time.clock()

        Forest.fit(X_train, y_train)
        scores = cross_val_score(Forest, X_train, y_train, cv=10)
        Accuracy_Forest=scores.mean()

        A1[i][n][0]=Accuracy_Forest
        T1[i][n][0]=toc-tic

A1.shape=(len(n_estimatorsA),len(max_featuresA))
im1 = plt.imshow(A1, cmap=plt.cm.viridis, interpolation='nearest')
plt.xlabel("# estimators")
plt.ylabel("# features")
plt.xticks(range(len(n_estimatorsA)))
plt.yticks(range(len(max_featuresA)))
'''
plt.subplot(1,2,2);
plt.title('Scatter',fontweight='bold',fontsize=15);
plt.xlabel('n_estimators');
plt.ylabel('Mean score cross val');
plt.show()
'''
return
```

```
Question3()
```

```
[ 1  4  6  9 11 14 16 19 22 24 27 29 32 35 37 40 42 45 47 50]
[ 1  3  6  8 10 13 15 18 20 22 25 27 29 32 34 37 39 41 44 46]
Random Forest iteration: 0


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
```

Random Forest iteration: 1

```
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
```

Random Forest iteration: 2

```
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
```

```
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)


Random Forest iteration: 3


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
```

```
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
```

```
Random Forest iteration: 4
```

```
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
```

```
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)


Random Forest iteration: 5


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
```

```
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)


Random Forest iteration: 6


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
```

```
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)


Random Forest iteration: 7


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
```

```
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)


Random Forest iteration: 8


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
```

```
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)


Random Forest iteration: 9


C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
```

```
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
C:\Users\s140737\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\model_selection\_s
  % (min_groups, self.n_splits)), Warning)
```

1.3 3. Understanding Ensembles (5 points (3+2))

Do a deeper analysis of how RandomForests and Gradient Boosting reduce their prediction error. We'll use the MAGIC telescope dataset (http://www.openml.org/d/1120). When high-energy particles hit the atmosphere, they produce chain reactions of other particles called 'showers', and you need to detect whether these are caused by gamma rays or cosmic rays.

```
In [ ]: # Get the data
        magic_data = oml.datasets.get_dataset(1120) # Download MAGIC Telescope data
        X, y = magic_data.get_data(target=magic_data.default_target_attribute);
```

```
In [ ]: # Quick visualization
        X, y, attribute_names = magic_data.get_data(target=magic_data.default_target_attribute
        magic = pd.DataFrame(X, columns=attribute_names)
        magic.plot(figsize=(20,10))
        # Also plot the target: 1 = gamma, 0 = background
        pd.DataFrame(y).plot(figsize=(20,1));
```

1 . Do a bias-variance analysis of both algorithms. For each, vary the number of trees on a log scale from 1 to 1024, and plot the bias error (squared), variance, and total error (in one plot per algorithm). Interpret the results. Which error is highest for small ensembles, and which reduced most by each algorithm as you use a larger ensemble? When are both algorithms under- or over-fitting? Provide a detailed explanation of why random forests and gradient boosting behave this way. - See lecture 3 for an example on how to do the bias-variance decomposition - To save time, you can use a 10% stratified subsample in your initial experiments, but show the plots for the full dataset in your report.

```
In [ ]: from sklearn.model_selection import ShuffleSplit
        from sklearn import ensemble


        def Question4():
            # Bootstraps
            n_treesAF=np.logspace(np.log10(1),np.log10(1024),20)
            n_treesAFR=np.round(n_treesAF)
            n_treesA=n_treesAFR.astype(int)
            print(n_treesA)
```

```
n_repeat = 5
shuffle_split = ShuffleSplit(train_size=0.01, n_splits=n_repeat)

bias_sq=np.zeros((len(n_treesA),1))
var=np.zeros((len(n_treesA),1))
error=np.zeros((len(n_treesA),1))
bias_sqG=np.zeros((len(n_treesA),1))
varG=np.zeros((len(n_treesA),1))
errorG=np.zeros((len(n_treesA),1))

for n in range(len(n_treesA)):
    # Store sample predictions
    y_all_pred = [[] for _ in range(len(y))]

    # Train classifier on each bootstrap and score predictions
    for i, (train_index, test_index) in enumerate(shuffle_split.split(X)):
        # Train and predict
        clf =  RandomForestClassifier(n_estimators=n_treesA[n]) #ensemble.Gradient
        clf.fit(X[train_index], y[train_index])
        y_pred = clf.predict(X[test_index])

        # Store predictions
        for i,index in enumerate(test_index):
            y_all_pred[index].append(y_pred[i])

            # Compute bias, variance, error
        bias_sumi=0
        var_sumi=0
        error_sumi=0
        for i in range(len(y_all_pred)):
            x=y_all_pred[i]
            if not len(x)==0:
                bias_sumi=bias_sumi+(1 - x.count(y[i])/len(x))**2 * len(x)/n_repeat
                var_sumi=var_sumi+((1 - ((x.count(0)/len(x))**2 + (x.count(1)/len(x
                error_sumi=error_sumi+(1 - x.count(y[i])/len(x)) * len(x)/n_repeat

    bias_sq[n]=bias_sumi
    var[n]=var_sumi
    error[n]=var_sumi
    print("Forest tree Bias squared: %.2f, Variance: %.2f, Total error: %.2f" % (bi
    # Train classifier on each bootstrap and score predictions
    for i, (train_index, test_index) in enumerate(shuffle_split.split(X)):
        # Train and predict
        clf =  ensemble.GradientBoostingClassifier(n_estimators=n_treesA[n])
        clf.fit(X[train_index], y[train_index])
        y_pred = clf.predict(X[test_index])

        # Store predictions
```

```
                for i,index in enumerate(test_index):
                    y_all_pred[index].append(y_pred[i])

                    # Compute bias, variance, error
                bias_sumi=0
                var_sumi=0
                error_sumi=0
                for i in range(len(y_all_pred)):
                    x=y_all_pred[i]
                    if not len(x)==0:
                        bias_sumi=bias_sumi+(1 - x.count(y[i])/len(x))**2 * len(x)/n_repeat
                        var_sumi=var_sumi+((1 - ((x.count(0)/len(x))**2 + (x.count(1)/len(x
                        error_sumi=error_sumi+(1 - x.count(y[i])/len(x)) * len(x)/n_repeat

            bias_sqG[n]=bias_sumi
            varG[n]=var_sumi
            errorG[n]=var_sumi
            print("Gradient Boosting Bias squared: %.2f, Variance: %.2f, Total error: %.2f
        plt.subplot(3,1,1);
        plt.semilogx(n_treesA, bias_sq);
        plt.semilogx(n_treesA, bias_sqG);
        plt.subplot(3,1,2);
        plt.semilogx(n_treesA, var);
        plt.semilogx(n_treesA, varG);
        plt.subplot(3,1,3);
        plt.semilogx(n_treesA, error);
        plt.semilogx(n_treesA, errorG);
        plt.show()
        return

    Question4()
```

2 . A *validation curve* can help you understand when a model starts under- or overfitting. It plots both training and test set error as you change certain characteristics of your model, e.g. one or more hyperparameters. Build validation curves for gradient boosting, evaluated using AUROC, by varying the number of iterations between 1 and 500. In addition, use at least two values for the learning rate (e.g. 0.1 and 1), and tree depth (e.g. 1 and 4). This will yield at least 4 curves. Interpret the results and provide a clear explanation for the results. When is the model over- or underfitting? Discuss the effect of the different combinations learning rate and tree depth and provide a clear explanation. - While scikit-learn has a `validation_curve` function, we'll use a modified version (below) that provides a lot more detail and can be used to study more than one hyperparameter. You can use a default train-test split.

```
In [4]: # Plots validation curves for every classifier in clfs.
        # Also indicates the optimal result by a vertical line
```

```
# Uses 1-AUROC, so lower is better
def validation_curve(clfs, X_test, y_test, X_train, y_train):
    for n,clf in enumerate(clfs):
        test_score = np.empty(len(clf.estimators_))
        train_score = np.empty(len(clf.estimators_))

        for i, pred in enumerate(clf.staged_decision_function(X_test)):
            test_score[i] = 1-roc_auc_score(y_test, pred)

        for i, pred in enumerate(clf.staged_decision_function(X_train)):
            train_score[i] = 1-roc_auc_score(y_train, pred)

        best_iter = np.argmin(test_score)
        learn = clf.get_params()['learning_rate']
        depth = clf.get_params()['max_depth']
        test_line = plt.plot(test_score,
                             label='learn=%.1f depth=%i (%.2f)'%(learn,depth,
                                                                 test_score[best_iter])

        colour = test_line[-1].get_color()
        plt.plot(train_score, '--', color=colour)

        plt.xlabel("Number of boosting iterations")
        plt.ylabel("1 - area under ROC")
        plt.axvline(x=best_iter, color=colour)

    plt.legend(loc='best')
```