Data Analysis for Epileptic Seizure Prediction

Arman Kerimbek

Supervisor: Dr. Bruce Smith

Dalhousie University, Halifax, Nova Scotia

Abstract

Terminal remission – also known as epileptic seizure freedom at the end of follow-up by a pediatrician – will be analyzed using data analysis tools such as generalized linear models with Binomial distribution, classification trees, and random forests due to the binary nature of the response variable. The status of terminal remission is a binary data with no missing values and two levels: in remission and not in remission. The above data analysis techniques will be used to determine which factors – patient's sex, age at first seizure, health condition, types of seizures, socio-economic status of their family, etc. – are important in predicting terminal remission status. While trying to identify the key factors that affect the outcome of epilepsy, the said techniques will be discussed in depth by looking at their similarities, differences, advantages, and disadvantages.

Introduction

Children who have epilepsy can outgrow the condition or can control the condition with medication; thus, the main purpose of the analysis is to identify the key factors that affect the outcome of epilepsy. The dataset that will be used to answer that scientific question of interest is a collection of records on 554 children diagnosed with epilepsy provided by two retired pediatricians from the IWK Health Centre in Halifax, Nova Scotia. When in it comes keeping clinical records, it is a common practice to record all observations made during a clinical encounter. Although there are 451 variables in the data, only few of them carry some explanatory value; thus, a careful variable exploration and selection are required when working with medical data. In this case, the Exploratory Data Analysis will be the first step in making "an educated guess" and selecting the variables that might potentially be correlated with the terminal remission status. Later, the preselected variables will be used to fit a logistic regression model, to build a classification tree, and to grow a random forest.

Methods

Exploratory Data Analysis

Exploratory Data Analysis (EDA) should always be the first step taken in any data analysis project. It involves using graphical and descriptive tools that help to better understand the data before beginning any analysis. It tells what kind of model to use and what kind of assumptions to make. Examining the variables helps to catch mistakes, find patterns, discover potential outliers, and determine whether any of the assumptions made have been violated.

Although the analysis will be mainly carried out in Python, Excel is an efficient tool that can be used for visual inspection of the data due to its substantial size. Once the process of variable selection is done, the data visualization tools, such as bar plots, histograms, and boxplots, will be used to visualize the data. Most of the explanatory variables are categorical, but there are some continuous measures as well. The variable 'aedend' which stores the status of terminal remission should be inspected for any missing values.

Generalized Logistic Regression

The logistic regression – more specifically a generalized linear model with Binomial distribution – will be used to make a suitable model for predicting the response variable and to test the significance of the preselected input variables. First, all of the variables will be used to fit a so called full or null model, and then all of the redundant variables will be removed – one at a time based on the largest p-value – in order to arrive to the simplest yet the most efficient model based on AIC criterion. It is crucial to note that AIC by itself does not tell anything about the quality/goodness of a single model. It becomes useful in comparing several models between each other to identify the most effective one. The lower the AIC, the better the fit. AIC is calculated using the following formula:

$$AIC = 2k - 2(\text{log-likelihood}), \text{ where } k \text{ is the number of parameters used in the model}$$

The AIC criterion is very similar to R-squared, which penalizes the use of too many explanatory variables to improve the fit. Similarly, AIC goes up in case the model is too complex.

The model assumptions, such as observation independence, absence of multicollinearity, and balanced sample, should be checked after the model is fit. Splitting the data into train and test sets will allow us to calculate the classification accuracy rate, which will later be used to compare the three data analysis techniques. The goal is essentially to use the train set to train a model, and then use the test set to test the previously trained model.

Classification Tree

Classification tree, also known as CART, is a data analysis technique used to classify observations into classes of a categorical response variable based on the values of the input variables. When one fits linear models and GLMs, they need to make sure that appropriate distributional assumptions are made. They need to check that the residuals are independent and identically distributed from a normal distribution with mean zero and constant standard deviation for linear models; and need to specify the distribution of the response variable for GLMs. The main advantage of the CARTs is that they do not require making any distributional assumptions such as the ones mentioned above. In addition, the CARTs are easier to interpret and visualize, and they have a way to use observations with missing values (using surrogate splits) instead of simply removing them. The first step will be using the train set to build a classification tree, and then test the performance of the classification tree on the test set.

Further, the said classification tree can be pruned in order to reduce overfitting. Pruning a decision tree reduces its size by removing branches that "contribute the least"; thus, making it less complex and increasing its (test) classification accuracy. Although, the pruned classification tree might not fit or predict the train set as well as before, it will be easer to interpret a simpler tree.

Random Forest

In theory, random forest does a better job than classification trees because it consists of 500 reasonably uncorrelated trees and it uses the most frequent classification (something like majority vote) out of these 500 trees to make predictions. Due to the fact that every tree is trained on a different sample of data, the random forest is good at the reduction of overfitting.

As mentioned previously, decision trees are quite prone to overfitting; therefore, we use random forests in order to reduce the risk of overfitting. Since every tree in a random forest is trained on a different/random subset of the data, the trees are relatively uncorrelated and altogether produce a higher classification accuracy than a single decision tree. At the same time, single decision trees are easier to interpret and plot; and they can be plotted rather quickly. Unlike single classification trees, random forests cannot handle missing values and simply removes them.

Results

Symptomatic epilepsy is a type of epilepsy that caused by an injury to the brain as a result of significant head injury, CNS infection, stroke, brain tumor, surgery, etc. The bar charts in Figure 1 demonstrate that a significant proportion of patients who were diagnosed with symptomatic epilepsy are in remission; thus, this suggests that there might be a significant relationship which should be investigated further. The following frequency table clearly illustrates that relationship as well:

	Non-symptomatic	Symptomatic
In remission	156	57
Not in remission	315	26

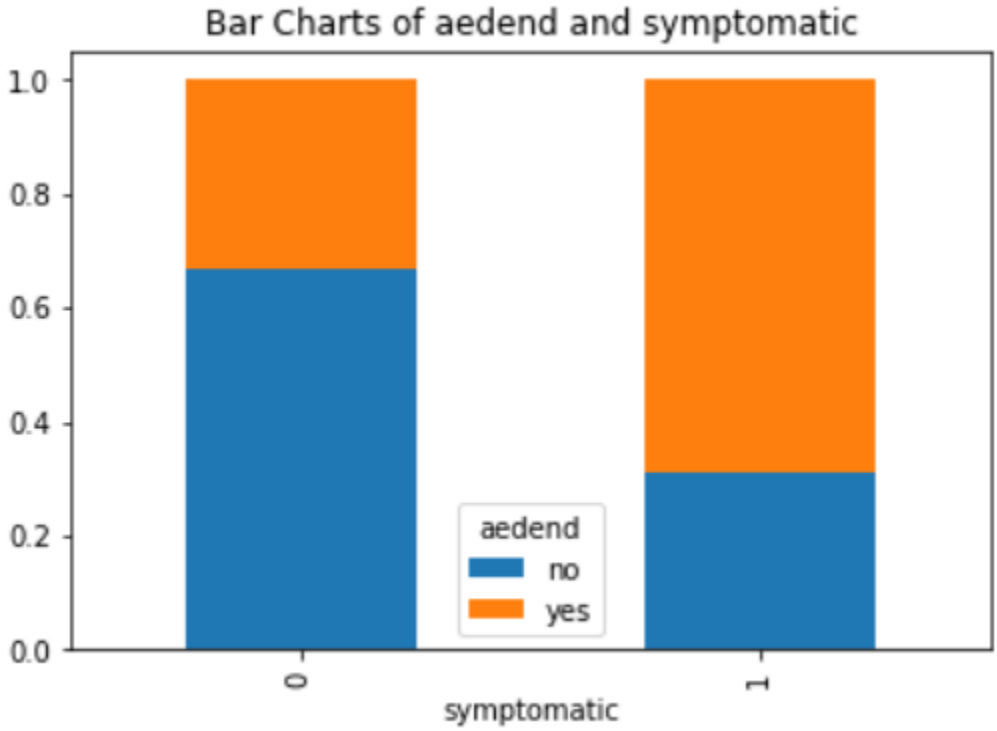


Figure 1. The bar charts for terminal remission status and symptomatic seizure status

Although most of the input variables are categorical, the 'agefirst' variable is a continuous variable which indicates the age of a patient at first seizure which can be a variable of interest. The side-by-side boxplots in Figure 2 suggest that, on average, the patients in remission experience their first seizures at a slightly younger age and the spread of the ages is much larger for them; this also suggests a potential relationship.

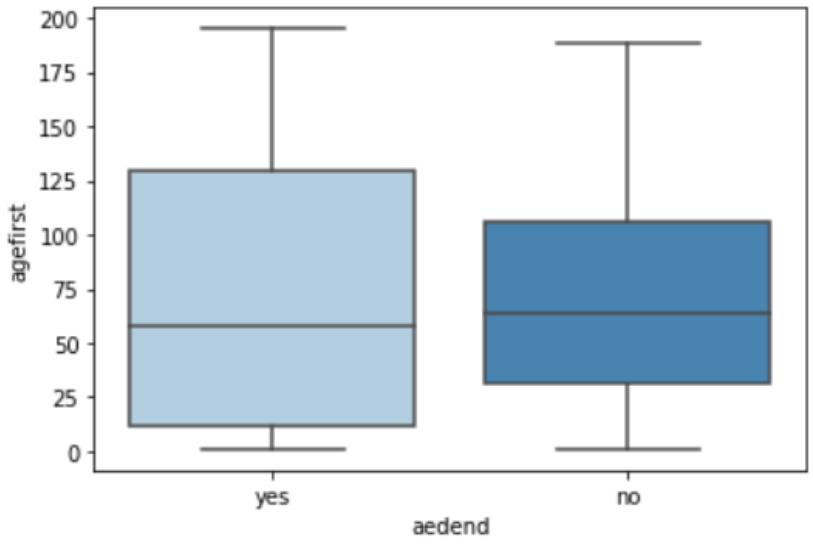


Figure 2. The side-by-side boxplots of age at first seizure (in months) for two levels of terminal remission status

The previously mentioned variables and six more were chosen to be included in the null logistic regression model and the model summary is displayed in Figure 3. As it was predicted, the variables 'symptomatic' and 'agefirst' are significant at the significance level of alpha = 0.01; in addition, the variables 'generalized' – indicator variable for a type of epilepsy that occurs throughout the whole brain – and 'fu' – total follow-up from first seizure to last contact in months – show some potential to be significant in a reduced model. The summary of the null model is illustrated in Figure 3.

```
Optimization terminated successfully.
         Current function value: 114.569145
         Iterations 5
                         Results: Logit
==================================================================
Model:              Logit            Pseudo R-squared: inf
Dependent Variable: y                AIC:              126960.6128
Date:               2020-04-20 10:06 BIC:              126999.4673
No. Observations:   554              Log-Likelihood:   -63471.
Df Model:           8                LL-Null:          0.0000
Df Residuals:       545              LLR p-value:      1.0000
Converged:          1.0000           Scale:            1.0000
No. Iterations:     5.0000
------------------------------------------------------------------
                 Coef.   Std.Err.     z     P>|z|    [0.025  0.975]
------------------------------------------------------------------
intercept       -0.8361    0.5116  -1.6344  0.1022  -1.8388  0.1665
agefirst         0.0066    0.0019   3.4623  0.0005   0.0029  0.0103
symptomatic      1.7685    0.5168   3.4223  0.0006   0.7557  2.7813
sex             -0.0287    0.1866  -0.1538  0.8777  -0.3944  0.3370
focal           -0.0806    0.4769  -0.1689  0.8659  -1.0153  0.8542
fu              -0.0014    0.0009  -1.5571  0.1194  -0.0031  0.0004
poor            -0.0339    0.2283  -0.1487  0.8818  -0.4813  0.4135
adequate         0.0458    0.2273   0.2014  0.8404  -0.3998  0.4914
generalized      0.2808    0.5271   0.5328  0.5942  -0.7522  1.3139
==================================================================
```

Figure 3. The summary of the null GLM model

The said reduced model is obtained by manually removing the insignificant variables one at a time and its summary is displayed in Figure 4. Unfortunately, the variables 'fu' and 'generalized' did not prove to be significant and the reduced model only contain two variables: 'symptomatic' and 'agefirst', but it is the best model based on the AIC criterion. The AIC for two models is shown in the table below:

Model	AIC
Full	126960.6128
Reduced	122936.9132
Reduced (with an interaction term)	123637.9815

Results: Logit

Model:	Logit	Pseudo R-squared:	inf
Dependent Variable:	y	AIC:	122936.9132
Date:	2020-04-20 10:05	BIC:	122949.8647
No. Observations:	554	Log-Likelihood:	-61465.
Df Model:	2	LL-Null:	0.0000
Df Residuals:	551	LLR p-value:	1.0000
Converged:	1.0000	Scale:	1.0000
No. Iterations:	5.0000		

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
intercept	-1.2420	0.1862	-6.6686	0.0000	-1.6070	-0.8770
agefirst	0.0065	0.0018	3.5375	0.0004	0.0029	0.0101
symptomatic	1.8814	0.2834	6.6384	0.0000	1.3259	2.4369

Figure 4. The summary of the reduced GLM model

The side-by-side boxplots in Figure 5 demonstrate a potential relationship between the symptomatic epilepsy status and the age at first seize; therefore, the possibility of interaction between 'symptomatic' and 'agefirst' should not be ignored. An additional model with the interaction term was constructed; however, the interaction term has a huge p-value, shown in Figure 6, and almost no contribution to the model (increase in AIC), thus will not be included in the reduced model.

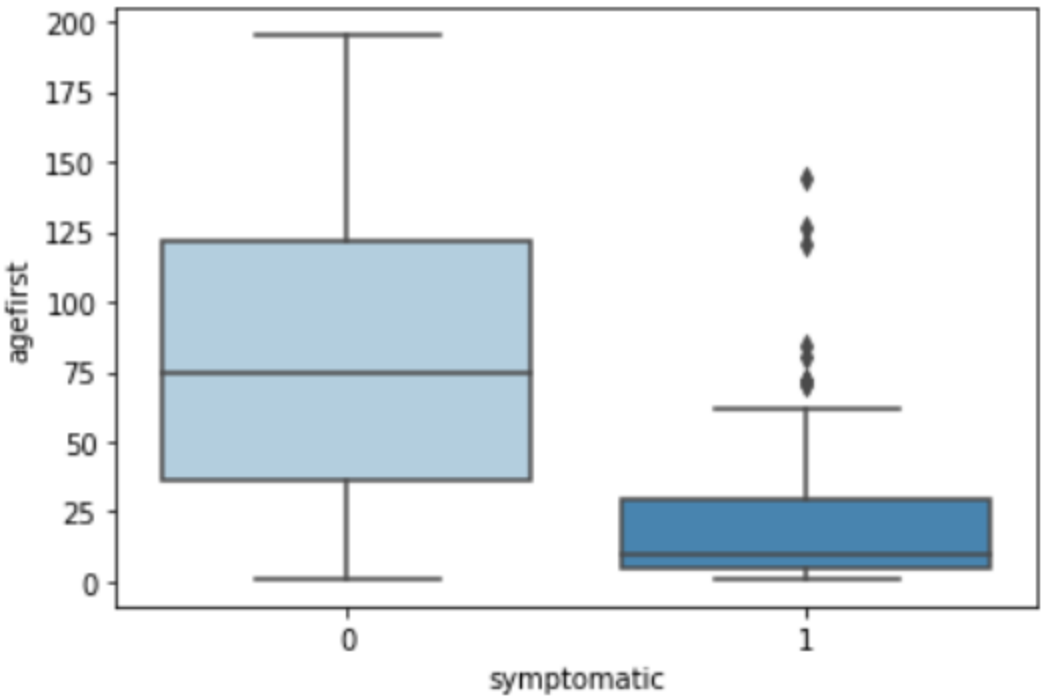


Figure 5. The side-by-side boxplots of age at first seizure (in months) for two levels of symptomatic epilepsy status

Results: Logit

Model:	Logit	Pseudo R-squared:	inf
Dependent Variable:	y	AIC:	123637.9815
Date:	2020-04-20 10:05	BIC:	123655.2502
No. Observations:	554	Log-Likelihood:	-61815.
Df Model:	3	LL-Null:	0.0000
Df Residuals:	550	LLR p-value:	1.0000
Converged:	1.0000	Scale:	1.0000
No. Iterations:	5.0000		

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
intercept	-1.2718	0.1902	-6.6883	0.0000	-1.6445	-0.8991
agefirst	0.0068	0.0019	3.6345	0.0003	0.0032	0.0105
symptomatic	2.0616	0.3563	5.7853	0.0000	1.3631	2.7600
interaction	-0.0070	0.0081	-0.8682	0.3853	-0.0230	0.0089

Figure 6. The summary of the GLM with an interaction term

Since the patients are independent of each other and the diagnostic plot (Residuals vs. Fitted) in Figure 7 shows no anomalies – the residuals are randomly scattered around the mean – the model assumptions are satisfied, so it is valid to proceed with predicting the remission status using the reduced model. The prediction accuracy will be displayed later in the analysis after the prediction accuracies for other techniques are obtained as well.

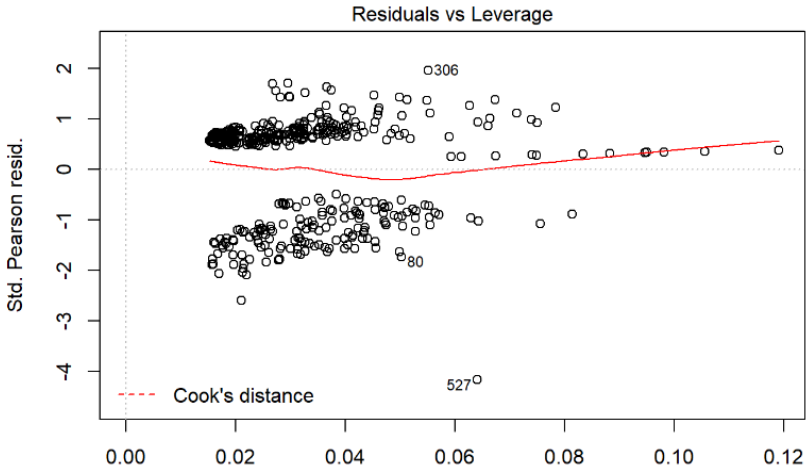


Figure 7. The residuals versus fitted plot

Unlike the logistic model which uses only two variables, the classification tree finds the two other variables to be important: 'fu', total follow-up from first seizure to last contact in months, and 'edum', the education level of mother.

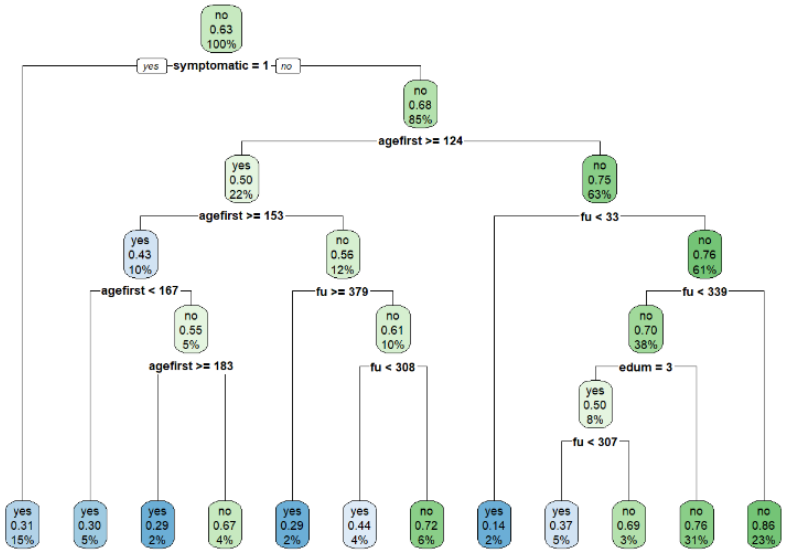


Figure 8. The original classification tree

Although, the decision trees are prone to overfitting and perhaps that is the reason why these variables are included in the tree, but it is important to understand why 'fu' can be quite misleading: as it is usually the case with medical trials, some patients tend to withdraw from a study if they recover or instead if they did not, but stopped attending doctor's appointments because they do

not help anymore. For that reason, it is reasonable to use a pruned tree which also relies on 'symptomatic' in predicting the terminal remission which confirms the results of the logistic regression

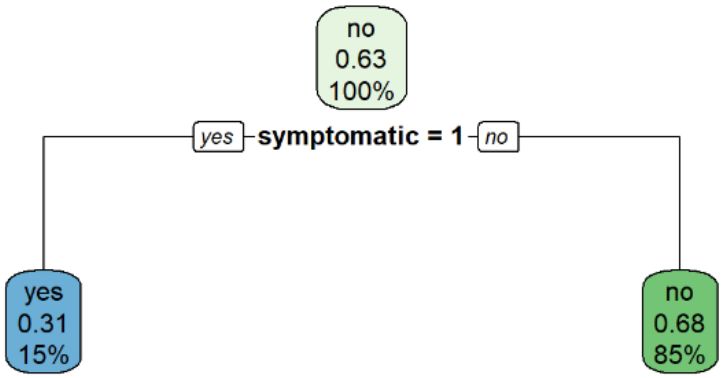


Figure 9. The pruned classification tree

The last step will be to grow a random forest: unlike the decision trees, a random forest's implementation in Python is not able to handle missing values. For that reason, the size of the train set is slightly reduced by removing the 'fu' and 'generalized' variables which have lots of missing values and were previously proven to have no explanatory power. The classification accuracy rates are given in the following table:

Technique	Classification Accuracy Rate
Logistic Regression	68.345%
Classification Tree	61.871%
Classification Tree (Pruned)	67.625%
Random Forest	62.589%

It is important to note that the above techniques all provide similar results, and all agree on the important variables.

A classification modelling problem involves classifying observations into classes of a categorical response. At the same time, a model could predict the probability of assigning an observation to each of the possible classes. The logistic regression returns well calibrated predictions by default as it directly optimizes log-loss, shown in Figure 8.

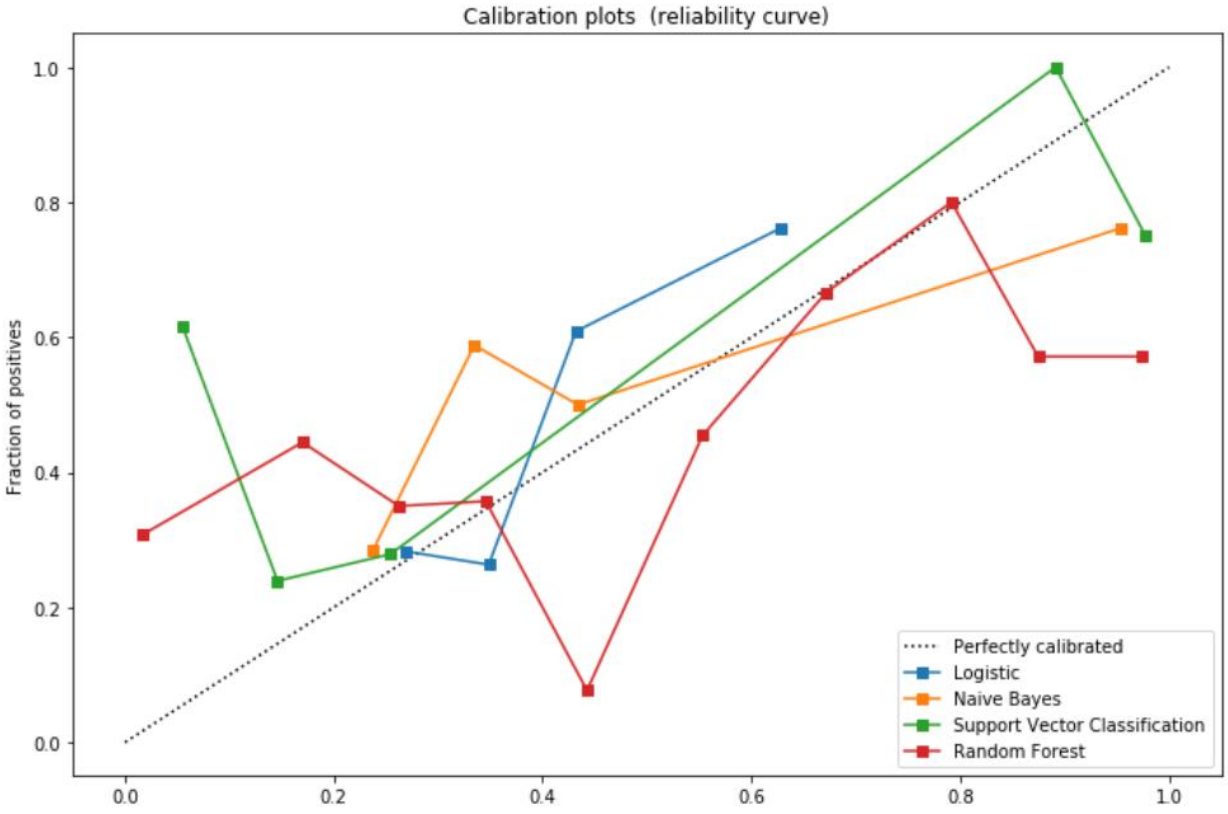


Figure 10. The calibration plots

Conclusion

The three techniques used in the analysis – Logistic Regression, Classification Tree, and Random Forest – all agree on that the terminal remission status is highly affected by two major factors: the status of symptomatic epilepsy, whether epilepsy was caused by an injury to the brain or not; and the age at the first seizure, which is recorded in months. A patient is more likely to recover from epilepsy if they are diagnosed with symptomatic epilepsy, and if the first seizure occurs at a younger age. The three techniques all provided quite similar classification accuracy rates, meaning that the results are quite stable and reliable.

Data Analysis for Epileptic Seizure Prediction: Python code

Arman Kerimbek Supervisor: Dr. Bruce Smith Dalhousie University, Halifax, Nova Scotia

In [3]:
```
#this is a package used to silence warnings
import warnings
warnings.filterwarnings('ignore')
```

In [4]:
```
#importing packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib as mpl
import seaborn as sns
import statsmodels.api as sm
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
from patsy import dmatrices
import sklearn
from sklearn import datasets
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import LinearSVC
from sklearn.calibration import calibration_curve
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
from sklearn import tree
from sklearn.externals.six import StringIO
from IPython.display import Image
from sklearn.tree import export_graphviz
import pydotplus
```

Exploratory Data Analysis

In [5]:
```
#reading in the data from a camfield.csv file
data = pd.read_csv('camfield.csv', header = 0)
#exploring the dataset
data.head()
```

Out[5]:

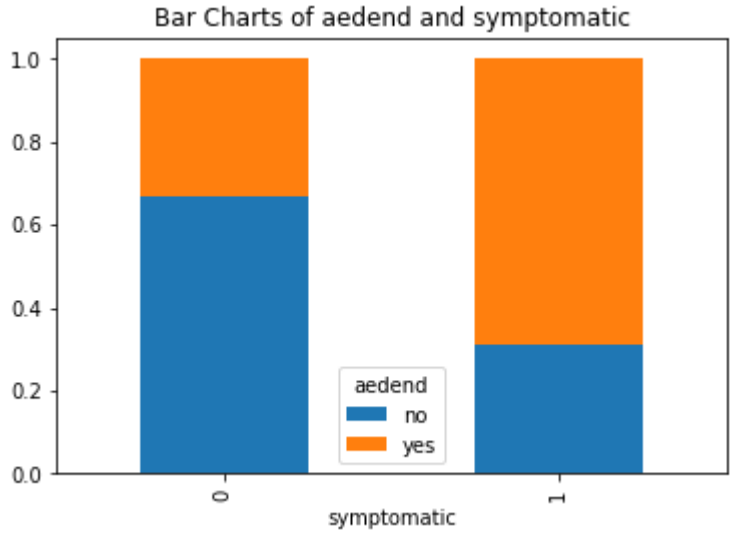
	Column1	groupID	nogroup	dob	sex	ca
0	1	700.0	NaN	NaN	female	515
1	2	NaN	NaN	##########...	male	80
2	3	NaN	NaN	12421987200	female	250
3	4	NaN	NaN	##########...	female	6
4	5	777.0	NaN	NaN	female	478

5 rows × 462 columns

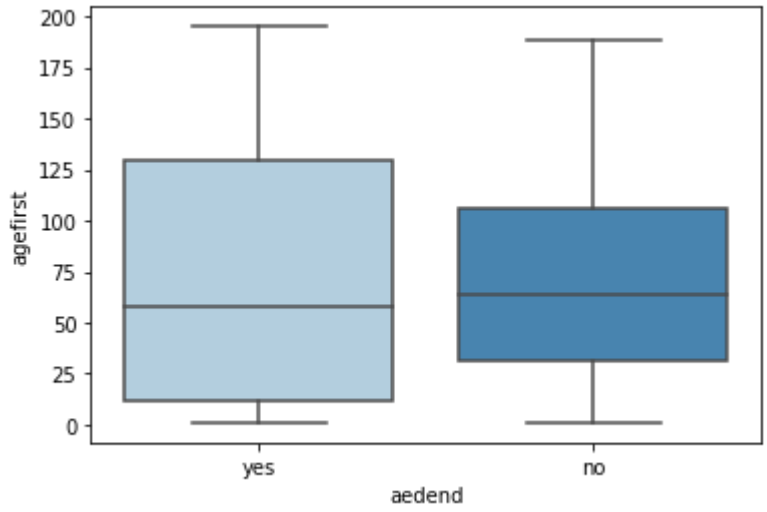
In [6]:
```
#getting the dimensions
data.shape
#removing the first column
data = data.drop(columns = "Column1")
#checking
data.shape
```

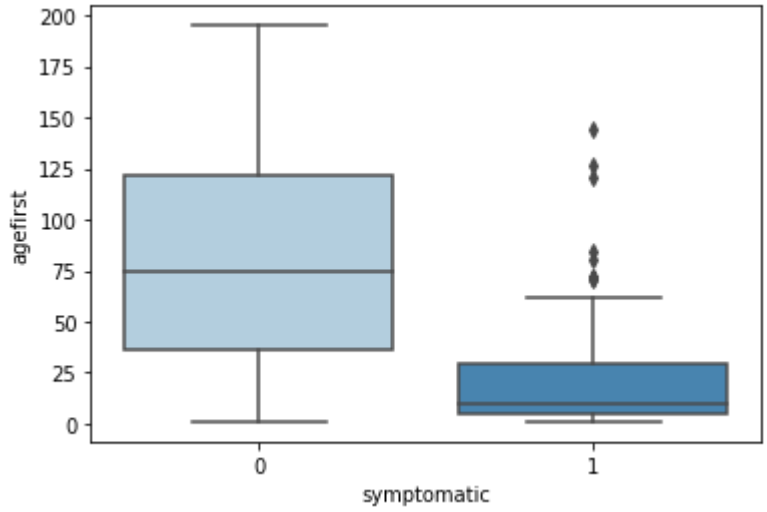
Out[6]: (554, 461)

In [7]:
```
#some visualization tools
%matplotlib inline
table = pd.crosstab(data.symptomatic, data.aedend)
table.div(table.sum(1).astype(float), axis=0).plot(kind='bar', stacked=True)
plt.title('Bar Charts of aedend and symptomatic')
plt.savefig('aedend_and_symptomatic')
```



In [8]:
```
#getting the needed columns
small_data = pd.concat([data.aedend, data.agefirst], axis = 1)
#making some boxplots
sns.boxplot(x = data.aedend, y = data.agefirst, palette = "Blues")
plt.show()
sns.boxplot(x = data.symptomatic, y = data.agefirst, palette = "Blues")
plt.show()
```





Generalized Logistic Regression

In [9]:
```
#modifying the variables
data.aedend = pd.get_dummies(data.aedend).yes
data.sex = pd.get_dummies(data.sex).male
data['intercept'] = 1
data['interaction'] = data.agefirst*data.symptomatic
```

In [17]:
```
#the logistic regression model with all the variables
logit_model = sm.Logit(data.aedend.to_numpy(), data[['intercept','agefirst',
'symptomatic',
                                                     'sex', 'focal', 'fu', 'po
or', 'adequate', 'generalized']].to_numpy())
result = logit_model.fit()
print(result.summary2(xname = ['intercept','agefirst', 'symptomatic',
                                                     'sex', 'focal', 'fu', 'po
or', 'adequate', 'generalized']))
```

```
Optimization terminated successfully.
         Current function value: 114.569145
         Iterations 5
                         Results: Logit
=================================================================
Model:              Logit            Pseudo R-squared: inf
Dependent Variable: y                AIC:              126960.6128
Date:               2020-04-20 10:06 BIC:              126999.4673
No. Observations:   554              Log-Likelihood:   -63471.
Df Model:           8                LL-Null:          0.0000
Df Residuals:       545              LLR p-value:      1.0000
Converged:          1.0000           Scale:            1.0000
No. Iterations:     5.0000
-----------------------------------------------------------------
                 Coef.   Std.Err.     z     P>|z|    [0.025  0.975]
-----------------------------------------------------------------
intercept       -0.8361    0.5116  -1.6344  0.1022  -1.8388  0.1665
agefirst         0.0066    0.0019   3.4623  0.0005   0.0029  0.0103
symptomatic      1.7685    0.5168   3.4223  0.0006   0.7557  2.7813
sex             -0.0287    0.1866  -0.1538  0.8777  -0.3944  0.3370
focal           -0.0806    0.4769  -0.1689  0.8659  -1.0153  0.8542
fu              -0.0014    0.0009  -1.5571  0.1194  -0.0031  0.0004
poor            -0.0339    0.2283  -0.1487  0.8818  -0.4813  0.4135
adequate         0.0458    0.2273   0.2014  0.8404  -0.3998  0.4914
generalized      0.2808    0.5271   0.5328  0.5942  -0.7522  1.3139
=================================================================
```

In [16]:
```
#The reduced model
logit_model = sm.Logit(data.aedend.to_numpy(), data[['intercept','agefirst',
'symptomatic']].to_numpy())
result = logit_model.fit()
print(result.summary2(xname = ['intercept','agefirst', 'symptomatic']))
```

```
Optimization terminated successfully.
         Current function value: 110.948478
         Iterations 5
                         Results: Logit
=================================================================
Model:              Logit            Pseudo R-squared: inf
Dependent Variable: y                AIC:              122936.9132
Date:               2020-04-20 10:05 BIC:              122949.8647
No. Observations:   554              Log-Likelihood:   -61465.
Df Model:           2                LL-Null:          0.0000
Df Residuals:       551              LLR p-value:      1.0000
Converged:          1.0000           Scale:            1.0000
No. Iterations:     5.0000
-----------------------------------------------------------------
                Coef.   Std.Err.     z     P>|z|    [0.025   0.975]
-----------------------------------------------------------------
intercept      -1.2420    0.1862  -6.6686  0.0000  -1.6070  -0.8770
agefirst        0.0065    0.0018   3.5375  0.0004   0.0029   0.0101
symptomatic     1.8814    0.2834   6.6384  0.0000   1.3259   2.4369
=================================================================
```

In [15]:
```
#The model with interaction
logit_model = sm.Logit(data.aedend.to_numpy(), data[['intercept','agefirst',
'symptomatic', 'interaction']].to_numpy())
result = logit_model.fit()
print(result.summary2(xname = ['intercept','agefirst', 'symptomatic', 'interac
tion']))
```

```
Optimization terminated successfully.
         Current function value: 111.579406
         Iterations 5
                         Results: Logit
=================================================================
Model:              Logit            Pseudo R-squared: inf
Dependent Variable: y                AIC:              123637.9815
Date:               2020-04-20 10:05 BIC:              123655.2502
No. Observations:   554              Log-Likelihood:   -61815.
Df Model:           3                LL-Null:          0.0000
Df Residuals:       550              LLR p-value:      1.0000
Converged:          1.0000           Scale:            1.0000
No. Iterations:     5.0000
-----------------------------------------------------------------
                Coef.   Std.Err.     z     P>|z|    [0.025   0.975]
-----------------------------------------------------------------
intercept      -1.2718    0.1902  -6.6883  0.0000  -1.6445  -0.8991
agefirst        0.0068    0.0019   3.6345  0.0003   0.0032   0.0105
symptomatic     2.0616    0.3563   5.7853  0.0000   1.3631   2.7600
interaction    -0.0070    0.0081  -0.8682  0.3853  -0.0230   0.0089
=================================================================
```

In [19]:
```
#making predictions
x_train, x_test, y_train, y_test = train_test_split(data[['intercept','agefirs
t', 'symptomatic']].to_numpy(),
                                                    data.aedend.to_numpy(), te
st_size = 0.25, random_state = 0)
logisticRegr = LogisticRegression()
logisticRegr.fit(x_train, y_train)
predictions = logisticRegr.predict(x_test)
score = logisticRegr.score(x_test, y_test)
print(score)
```

```
0.6834532374100719
```

Classification Tree

In [20]:
```
#making a classifcation tree
x_train, x_test, y_train, y_test = train_test_split(data[['agefirst', 'symptom
atic', 'sex', 'focal', 'fu', 'poor',
                                                         'adequate', 'general
ized']].to_numpy(),
                                                 data.aedend.to_numpy(), te
st_size = 0.25, random_state = 0)
#making predctions
dt = DecisionTreeClassifier()
dt.fit(x_train, y_train)
y_pred = dt.predict(x_test)
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
```

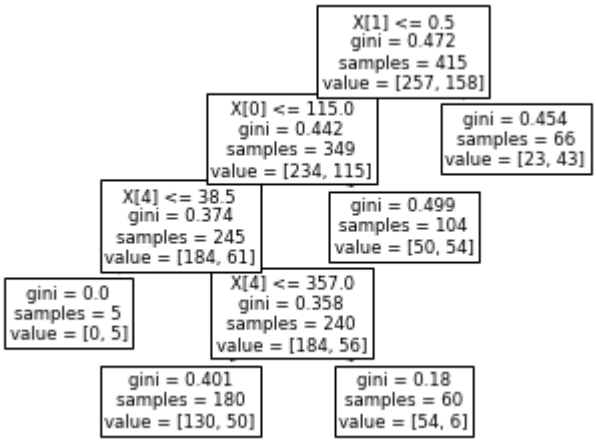
```
Accuracy: 0.6187050359712231
```

In [23]:
```
#pruning the above tree
dt = DecisionTreeClassifier(max_leaf_nodes=5)
dt.fit(x_train, y_train)
y_pred = dt.predict(x_test)
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.6762589928057554
```

In [24]:
```
tree.plot_tree(dt.fit(x_train, y_train))
```

Out[24]:
```
[Text(223.20000000000002, 195.696, 'X[1] <= 0.5\ngini = 0.472\nsamples = 415
\nvalue = [257, 158]'),
 Text(167.4, 152.208, 'X[0] <= 115.0\ngini = 0.442\nsamples = 349\nvalue = [2
34, 115]'),
 Text(111.60000000000001, 108.72, 'X[4] <= 38.5\ngini = 0.374\nsamples = 245
\nvalue = [184, 61]'),
 Text(55.800000000000004, 65.232, 'gini = 0.0\nsamples = 5\nvalue = [0, 5]'),
 Text(167.4, 65.232, 'X[4] <= 357.0\ngini = 0.358\nsamples = 240\nvalue = [18
4, 56]'),
 Text(111.60000000000001, 21.744, 'gini = 0.401\nsamples = 180\nvalue = [130,
50]'),
 Text(223.20000000000002, 21.744, 'gini = 0.18\nsamples = 60\nvalue = [54,
6]'),
 Text(223.20000000000002, 108.72, 'gini = 0.499\nsamples = 104\nvalue = [50,
54]'),
 Text(279.0, 152.208, 'gini = 0.454\nsamples = 66\nvalue = [23, 43]')]
```



Random Forest

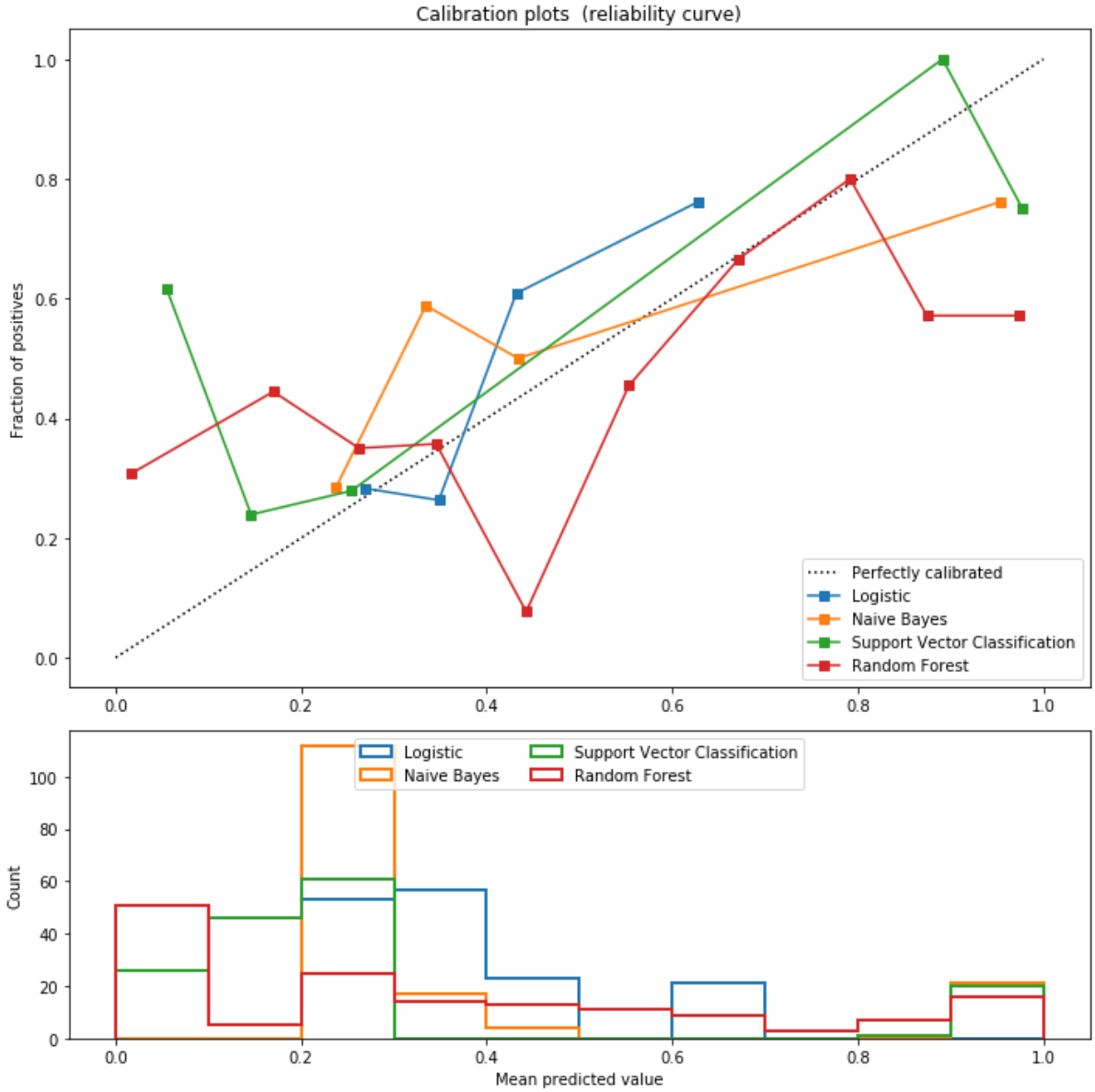
In [26]:
```
# Instantiate model with 500 decision trees
rf = RandomForestClassifier(n_estimators=1000)
# Train the model on training data
rf.fit(x_train, y_train)
y_pred = rf.predict(x_test)
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.6258992805755396
```

Probability Calibration

In [40]:

```
X_train = data[['intercept','agefirst', 'symptomatic']].to_numpy()[:400]
X_test = data[['intercept','agefirst', 'symptomatic']].to_numpy()[400:]
y_train = data.aedend.to_numpy()[:400]
y_test = data.aedend.to_numpy()[400:]

# Create classifiers
lr = LogisticRegression()
gnb = GaussianNB()
svc = LinearSVC(C=1.0)
rfc = RandomForestClassifier()


# #############################################################################
#
# Plot calibration plots

plt.figure(figsize=(10, 10))
ax1 = plt.subplot2grid((3, 1), (0, 0), rowspan=2)
ax2 = plt.subplot2grid((3, 1), (2, 0))

ax1.plot([0, 1], [0, 1], "k:", label="Perfectly calibrated")
for clf, name in [(lr, 'Logistic'),
                  (gnb, 'Naive Bayes'),
                  (svc, 'Support Vector Classification'),
                  (rfc, 'Random Forest')]:
    clf.fit(X_train, y_train)
    if hasattr(clf, "predict_proba"):
        prob_pos = clf.predict_proba(X_test)[:, 1]
    else:  # use decision function
        prob_pos = clf.decision_function(X_test)
        prob_pos = \
            (prob_pos - prob_pos.min()) / (prob_pos.max() - prob_pos.min())
    fraction_of_positives, mean_predicted_value = \
        calibration_curve(y_test, prob_pos, n_bins=10)

    ax1.plot(mean_predicted_value, fraction_of_positives, "s-",
             label="%s" % (name, ))

    ax2.hist(prob_pos, range=(0, 1), bins=10, label=name,
             histtype="step", lw=2)

ax1.set_ylabel("Fraction of positives")
ax1.set_ylim([-0.05, 1.05])
ax1.legend(loc="lower right")
ax1.set_title('Calibration plots  (reliability curve)')

ax2.set_xlabel("Mean predicted value")
ax2.set_ylabel("Count")
ax2.legend(loc="upper center", ncol=2)

plt.tight_layout()
plt.show()
```


Calibration plots (reliability curve)
Fraction of positives
Perfectly calibrated
Logistic
Naive Bayes
Support Vector Classification
Random Forest
Count
Mean predicted value