Q3 Summit - Demystifying Data Science

Peter Draznik

Forked and simplified from https://www.kaggle.com/emmaren/cold-calls-data-mining-and-model-selection (https://www.kaggle.com/emmaren/cold-calls-data-mining-and-model-selection)

In [1]:

```
%matplotlib inline
import graphviz
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
from sklearn.tree import DecisionTreeClassifier, export_graphviz
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.grid_search import GridSearchCV
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.linear_model import LogisticRegression
from sklearn import svm
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, ExtraTreesClassif
ier,GradientBoostingClassifier, VotingClassifier
from sklearn.naive_bayes import GaussianNB
```

```
/opt/conda/lib/python3.6/site-packages/sklearn/cross_validation.py:41:
DeprecationWarning: This module was deprecated in version 0.18 in favo
r of the model_selection module into which all the refactored classes
and functions are moved. Also note that the interface of the new CV it
erators are different from that of this module. This module will be re
moved in 0.20.
  "This module will be removed in 0.20.", DeprecationWarning)
/opt/conda/lib/python3.6/site-packages/sklearn/grid_search.py:42: Depr
ecationWarning: This module was deprecated in version 0.18 in favor of
the model_selection module into which all the refactored classes and f
unctions are moved. This module will be removed in 0.20.
  DeprecationWarning)
```

Read-in train and test datasets

In [2]:

```
train = pd.read_csv('../input/carInsurance_train.csv')
test = pd.read_csv('../input/carInsurance_test.csv')
```

Basic Exploration of Data

In [3]:

```
print('The train dataset has %d observations and %d features' % (train
.shape[0], train.shape[1]))
print('The test dataset has %d observations and %d features' % (test.s
hape[0], test.shape[1]))
```

```
The train dataset has 4000 observations and 19 features
The test dataset has 1000 observations and 19 features
```

In [4]:

```
train.head()
```

Out[4]:

	Id	Age	Job	Marital	Education	Default	Balance	HHInsurance	CarLoan	Comm
0	1	32	management	single	tertiary	0	1218	1	0	telepho
1	2	32	blue-collar	married	primary	0	1156	1	0	NaN
2	3	29	management	single	tertiary	0	637	1	0	cellula
3	4	25	student	single	primary	0	373	1	0	cellula
4	5	30	management	married	tertiary	0	2694	0	0	cellula

In [5]:

```
# Take a peak at the non-categorical
train.describe()
```

Out[5]:

HInsurance	CarLoan	LastContactDay	NoOfContacts	DaysPassed	PrevAttempts	CarInsurance
)00.00000	4000.000000	4000.000000	4000.000000	4000.000000	4000.000000	4000.000000
49275	0.133000	15.721250	2.607250	48.706500	0.717500	0.401000
50001	0.339617	8.425307	3.064204	106.685385	2.078647	0.490162
00000	0.000000	1.000000	1.000000	-1.000000	0.000000	0.000000
00000	0.000000	8.000000	1.000000	-1.000000	0.000000	0.000000
00000	0.000000	16.000000	2.000000	-1.000000	0.000000	0.000000
00000	0.000000	22.000000	3.000000	-1.000000	0.000000	1.000000
00000	1.000000	31.000000	43.000000	854.000000	58.000000	1.000000

In [6]:

```
# Take a peak at the categorical
train.describe(include=['O'])
```

Out[6]:

	Job	Marital	Education	Communication	LastContactMonth	Outcome	CallStart

count	3981	4000	3831	3098	4000	958	4000
unique	11	3	3	2	12	3	3777
top	management	married	secondary	cellular	may	failure	17:11:04
freq	893	2304	1988	2831	1049	437	3

Begin Plotting

In [7]:

```
# merge train and test data here in order to impute missing values all a
t once
all=pd.concat([train,test],keys=('train','test'))
```

In [8]:

```
all.head()
```

Out[8]:

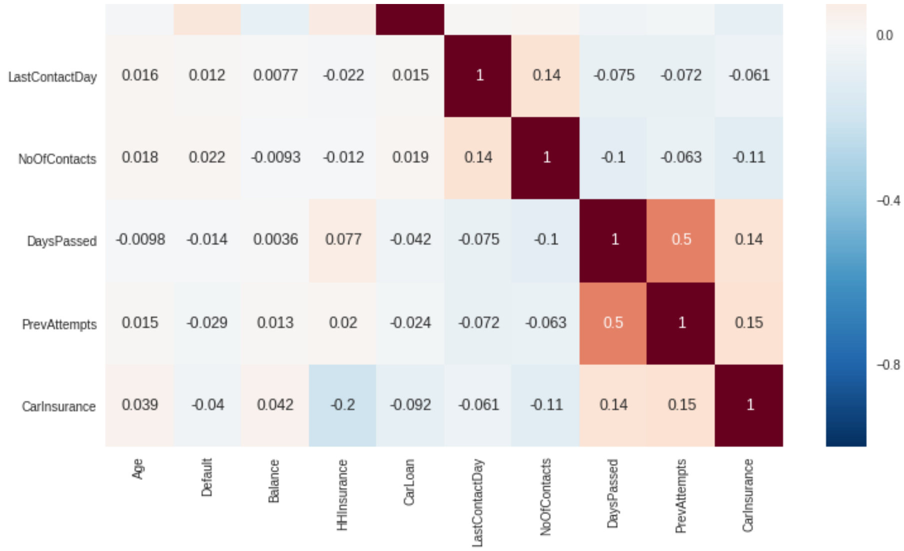
		Id	Age	Job	Marital	Education	Default	Balance	HHInsurance	CarLoan
train	0	1	32	management	single	tertiary	0	1218	1	0
	1	2	32	blue-collar	married	primary	0	1156	1	0
	2	3	29	management	single	tertiary	0	637	1	0
	3	4	25	student	single	primary	0	373	1	0
	4	5	30	management	married	tertiary	0	2694	0	0

In [9]:

```
# First check out correlations among numeric features
# Heatmap is a useful tool to get a quick understanding of which variabl
es are important
cor = all.corr()
cor = cor.drop(['Id'],axis=1).drop(['Id'],axis=0)
plt.figure(figsize=(12,12))
sns.heatmap(cor,annot=True)
```

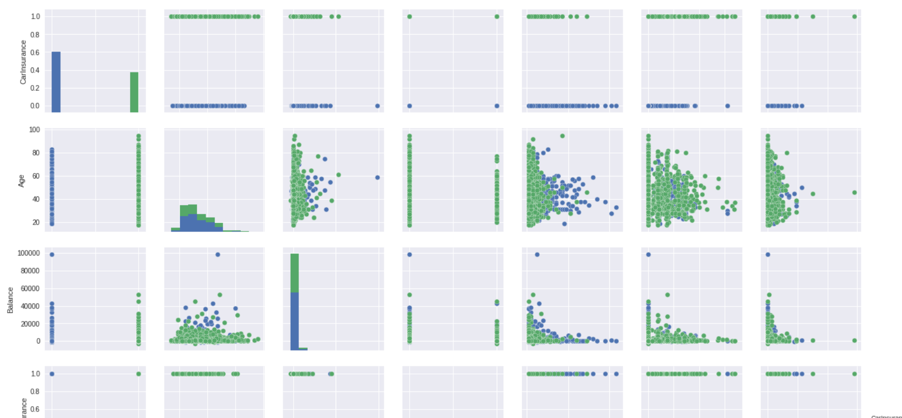
Out[9]:

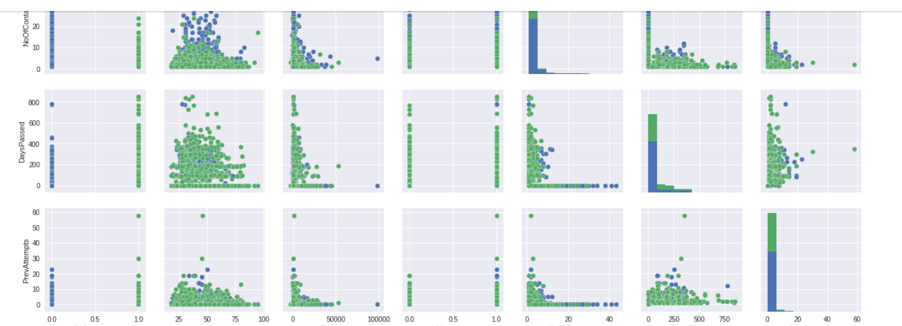
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7c86030588>
```



In [10]:

```
imp_feats = ['CarInsurance','Age','Balance','HHInsurance','NoOfContact
s','DaysPassed','PrevAttempts']
sns.pairplot(all[imp_feats],hue='CarInsurance',size=2.5)
plt.show()
```





Begin Missing Value Replacement

In [11]:

```
all.drop(['CarInsurance','Id'],axis=1,inplace=True)
print(all.shape)
```

```
(5000, 17)
```

In [12]:

```
total = all.isnull().sum()
pct = total/all.isnull().count()
NAs = pd.concat([total,pct],axis=1,keys=('Total','Pct'))
NAs[NAs.Total>0].sort_values(by='Total',ascending=False)
```

Out[12]:

	Total	Pct
Outcome	3799	0.7598
Communication	1123	0.2246
Education	216	0.0432
Job	24	0.0048

In [13]:

```
all_df = all.copy()

# Fill missing outcome as not in previous campaign
all_df[all_df['DaysPassed']==-1].count()
all_df.loc[all_df['DaysPassed']==-1,'Outcome']='NoPrev'

# Fill missing communication with none
all_df['Communication'].value_counts()
all_df['Communication'].fillna('None',inplace=True)

# Fill missing education with the most common education level by job typ
e
all_df['Education'].value_counts()
```



```
job_types = all_df.Job.value_counts().index
for job in job_types:
    mode = all_df[all_df.Job==job]['Education'].value_counts().nlarges
t(1).index
    edu_mode = np.append(edu_mode,mode)
edu_map=pd.Series(edu_mode,index=all_df.Job.value_counts().index)

# Apply the mapping to missing eductaion obs
for j in job_types:
    all_df.loc[(all_df['Education'].isnull()) & (all_df['Job']==j),'Ed
ucation'] = edu_map.loc[edu_map.index==j][0]
all_df['Education'].fillna('None',inplace=True)

# Fill missing job with none
all_df['Job'].fillna('None',inplace=True)
```

In [14]:

```
# Double check if there is still any missing value
```

```
# Double check if there is still any missing value
print("Remaining missing values: %d"%(all_df.isnull().sum().sum()))
all_df.head()
```

```
Remaining missing values: 0
```

Out[14]:

		Age	Job	Marital	Education	Default	Balance	HHInsurance	CarLoan	Con
train	0	32	management	single	tertiary	0	1218	1	0	telep
	1	32	blue-collar	married	primary	0	1156	1	0	Non
	2	29	management	single	tertiary	0	637	1	0	cellu
	3	25	student	single	primary	0	373	1	0	cellu
	4	30	management	married	tertiary	0	2694	0	0	cellu

Simplified Feature Engineering

In [15]:

```
# Get call length
all_df['CallEnd'] = pd.to_datetime(all_df['CallEnd'])
all_df['CallStart'] = pd.to_datetime(all_df['CallStart'])
all_df['CallStartHour'] = all_df['CallStart'].dt.hour
all_df['CallLength'] = ((all_df['CallEnd'] - all_df['CallStart'])/np.t
imedelta64(1,'m')).astype(float)
```

In [16]:

```
all_df['CallLengthPercent'] = all_df['CallLength']/all_df['CallLength'
].max()
```

In [17]:

```
all_df['AgePercent'] = all_df['Age']/all_df['Age'].max()
```

In [18]:

```
all_df['BalancePercent'] = all_df['Balance']/all_df['Balance'].max()
```

In [19]:

```
all_df['Education'] = all_df['Education'].replace({'None':0,'primary':
1,'secondary':2,'tertiary':3})
```

In [20]:

```
```

In [20]:

```
all_df = all_df.drop(['Age','Balance', 'CallLength', 'CallStart', 'Cal
lEnd'],axis=1)
```

In [21]:

```
all_df.head()
```

```
all_df.head()
```

Out[21]:

		Job	Marital	Education	Default	HHInsurance	CarLoan	Communication	Las
train	0	management	single	3	0	1	0	telephone	28
	1	blue-collar	married	1	0	1	0	None	26
	2	management	single	3	0	1	0	cellular	3
	3	student	single	1	0	1	0	cellular	11
	4	management	married	3	0	0	0	cellular	3

In [22]:

```
# Spilt numeric and categorical features
cat_feats = all_df.select_dtypes(include=['object']).columns
num_feats = all_df.select_dtypes(include=['float64','int64']).columns
num_df = all_df[num_feats]
print('There are %d numeric features and %d categorical features\n' %(
len(num_feats),len(cat_feats)))
print('Numeric features:\n',num_feats.values)
print('Categorical features:\n',cat_feats.values)
```

```
There are 12 numeric features and 5 categorical features

Numeric features:
 ['Education' 'Default' 'HHInsurance' 'CarLoan' 'LastContactDay'
 'NoOfContacts' 'DaysPassed' 'PrevAttempts' 'CallStartHour'
 'CallLengthPercent' 'AgePercent' 'BalancePercent']
Categorical features:
 ['Job' 'Marital' 'Communication' 'LastContactMonth' 'Outcome']
```

In [23]:

```
# One hot encoding
exlude = ['CallStart' 'CallEnd']
cat_feats = [val for val in cat_feats if val not in exlude]
cat_df = all_df[cat_feats]
cat_df = pd.get_dummies(cat_df)
cat_feats
```

Out[23]:

```
['Job', 'Marital', 'Communication', 'LastContactMonth', 'Outcome']
```

In [24]:

```
cat_df.head()
```

Out[24]:

		Job_None	Job_admin.	Job_blue-collar	Job_entrepreneur	Job_housemaid	Job_management
train	0	0	0	0	0	0	1
	1	0	0	1	0	0	0
	2	0	0	0	0	0	1
	3	0	0	0	0	0	0
	4	0	0	0	0	0	1

5 rows × 34 columns

Begin Data Splitting

In [25]:

```
# Recombine data
all_data = pd.concat([num_df,cat_df],axis=1)
all_data.head()
```

Out[25]:

		Education	Default	HHInsurance	CarLoan	LastContactDay	NoOfContacts	DaysPassed
train	0	3	0	1	0	28	2	-1
	1	1	0	1	0	26	5	-1
	2	3	0	1	0	3	1	119
	3	1	0	1	0	11	2	-1
	4	3	0	0	0	3	1	-1

5 rows × 46 columns

In [26]:

```
# Split train and test
idx=pd.IndexSlice
train_df=all_data.loc[idx[['train',],:]]
test_df=all_data.loc[idx[['test',],:]]
train_label=train['CarInsurance']
print(train_df.shape)
print(len(train_label))
print(test_df.shape)
# Train test split
x_train, x_test, y_train, y_test = train_test_split(train_df,train_lab
el,test_size = 0.3,random_state=3)
```

```
(4000, 46)
4000
(1000, 46)
```

In [27]:

```
x_test.head()
```

Out[27]:

		Education	Default	HHInsurance	CarLoan	LastContactDay	NoOfContacts	DaysPass
train	3626	3	0	0	0	12	1	-1
	3310	3	0	1	1	28	2	153
	1142	2	0	0	0	7	5	-1
	1767	1	0	0	0	7	7	-1

	2645	3	0	0	0	14	2	-1

5 rows × 46 columns

Plot Engineered Features

In [28]:

```
train_with_label = x_train.copy()
train_with_label['CarInsurance'] = y_train.values
train_with_label.head()
```

Out[28]:

		Education	Default	HHInsurance	CarLoan	LastContactDay	NoOfContacts	DaysPass
train	3209	1	0	1	1	13	5	-1
	3268	1	0	1	1	3	1	-1
	2374	2	0	0	0	30	1	-1
	885	2	0	0	0	19	1	-1
	2102	3	0	0	0	12	2	95

5 rows × 47 columns

In [29]:

```
col_names = ['CarInsurance', 'AgePercent', 'BalancePercent', 'CallLeng
thPercent', 'Education']
sns.pairplot(train_with_label[col_names],hue='CarInsurance',size=2.5)
plt.show()
```

Tools For Model Evaluation

In [30]:

```
# The confusion matrix plotting function is from the sklearn documentati
on below:
# http://scikit-learn.org/stable/auto_examples/model_selection/plot_conf
usion_matrix.html
import itertools
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]

    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1
])):
        plt.text(j, i, cm[i, j],
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')

class_names = ['Success','Failure']
```

In [31]:

```
def model_fit(model, x_train=x_train, y_train=y_train, x_test=x_test):
    clf = model.fit(x_train,y_train)
    y_pred = clf.predict(x_test)
    return {
        'clf': clf,
        'x_train':x_train,
```

```
        'y_train':y_train,
        'x_test':x_test,
        'y_pred':y_pred
    }
```

In [32]:

```
# Create a cross validation function
def get_best_model(estimator, params_grid={}):

    model = GridSearchCV(estimator = estimator,param_grid = params_gri
d,cv=3, scoring="accuracy", n_jobs= -1)
    model.fit(x_train,y_train)
    print('\n--- Best Parameters -----------------------------')
    print(model.best_params_)
    print('\n--- Best Model -----------------------------')
    best_model = model.best_estimator_
    print(best_model)
    return best_model
```

In [33]:

```
# Based off of: https://www.kaggle.com/emmaren/cold-calls-data-mining-an
d-model-selection
def model_report(clf, y_pred, y_test=y_test, class_names=['Success','F
ailure'], cv=5, feature_imp=True):
    # model report
    cm = confusion_matrix(y_test,y_pred)
    plot_confusion_matrix(cm, classes=class_names, title='Confusion ma
trix')

    print('\n--- Train Set -----------------------------')
    print('Accuracy: %.5f +/- %.4f' % (np.mean(cross_val_score(clf,x_t
rain,y_train,cv=cv)),np.std(cross_val_score(clf,x_train,y_train,cv=cv
))))
    print('AUC: %.5f +/- %.4f' % (np.mean(cross_val_score(clf,x_train,
y_train,cv=cv,scoring='roc_auc')),np.std(cross_val_score(clf,x_train,y
_train,cv=cv,scoring='roc_auc'))))
    print('\n--- Validation Set -----------------------------')
    print('Accuracy: %.5f +/- %.4f' % (np.mean(cross_val_score(clf,x_t
est,y_test,cv=cv)),np.std(cross_val_score(clf,x_test,y_test,cv=cv))))
    print('AUC: %.5f +/- %.4f' % (np.mean(cross_val_score(clf,x_test,y
_test,cv=cv,scoring='roc_auc')),np.std(cross_val_score(clf,x_test,y_te
st,cv=cv,scoring='roc_auc'))))
    print('----------------------------------------------')

    # feature importance
    if feature_imp:
        feat_imp = pd.Series(clf.feature_importances_,index=all_data.c
olumns)
        feat_imp = feat_imp.nlargest(15).sort_values()
        plt.figure()
        feat_imp.plot(kind="barh",figsize=(6,8),title="Most Important
 Features")
```

Train And Evaluate Models

Decision Tree - Unoptimized

In [34]:
```
dt = DecisionTreeClassifier()
dt.fit(x_train, y_train)
```

Out[34]:
```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=
None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=
None,
            splitter='best')
```

In [35]:
```
x_test.head()
```

Out[35]:

		Education	Default	HHInsurance	CarLoan	LastContactDay	NoOfContacts	DaysPass
train	3626	3	0	0	0	12	1	-1
	3310	3	0	1	1	28	2	153
	1142	2	0	0	0	7	5	-1
	1767	1	0	0	0	7	7	-1
	2645	3	0	0	0	14	2	-1

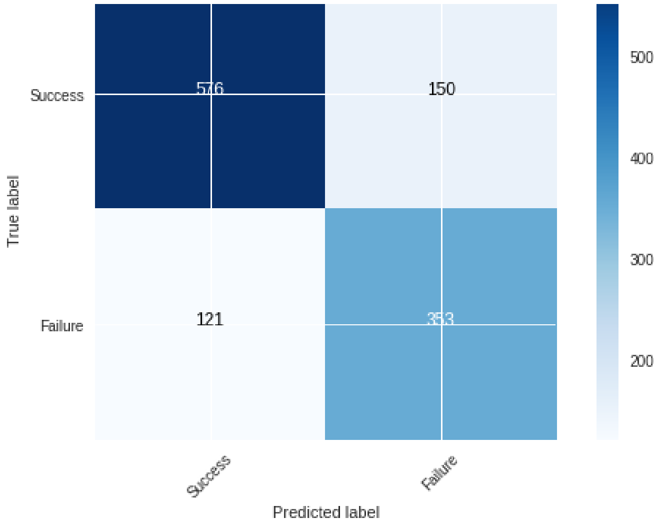
5 rows × 46 columns

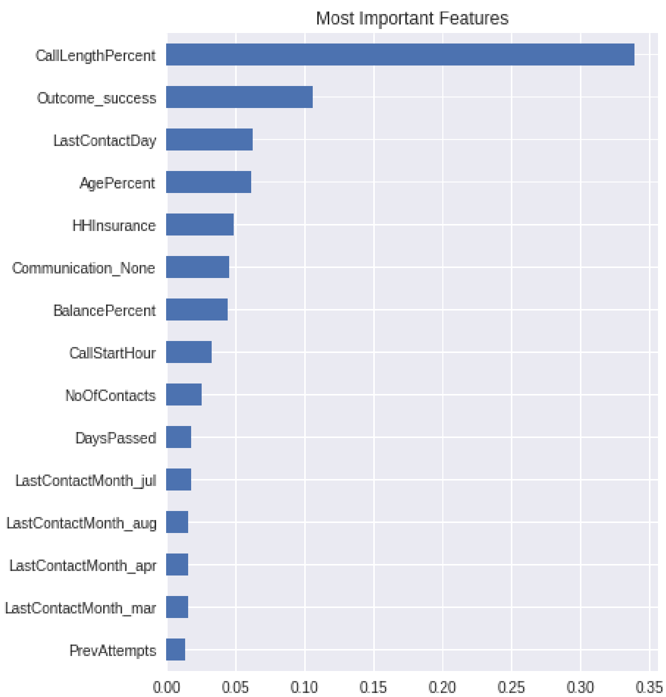
In [36]:
```
dt.predict(x_train)
```

Out[36]:
```
array([1, 0, 1, ..., 0, 1, 1])
```

In [37]:
```
model_report(dt, dt.predict(x_test), y_test)
```

```
--- Train Set ----------------------------
Accuracy: 0.79571 +/- 0.0061
AUC: 0.78181 +/- 0.0156

--- Validation Set -----------------------------
Accuracy: 0.78001 +/- 0.0168
AUC: 0.76266 +/- 0.0112
----------------------------------------------
```





Decision Tree - Optimized

In [38]:

```
parameters = {"criterion": ["gini", "entropy"],
              "min_samples_split": [2, 10, 20],
              "max_depth": [None, 2, 5, 10],
              "min_samples_leaf": [1, 5, 10],
              "max_leaf_nodes": [None, 5, 10, 20],
              }
dt_best = get_best_model(dt, parameters)
```

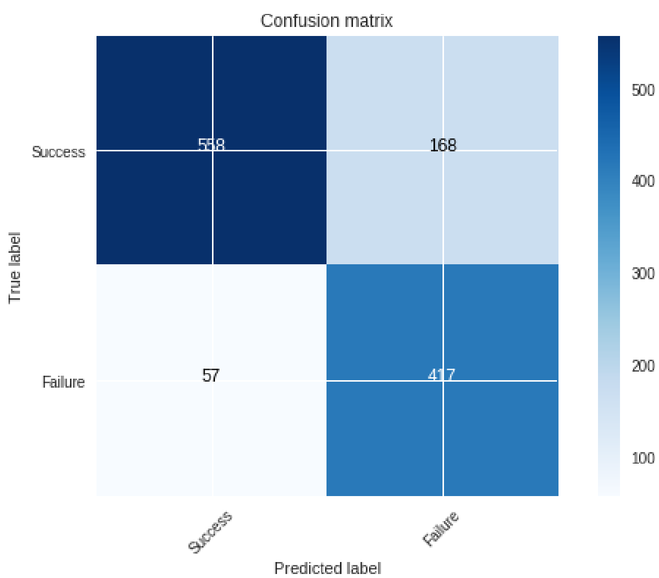
```
--- Best Parameters -----------------------------
{'criterion': 'gini', 'max_depth': 10, 'max_leaf_nodes': 20, 'min_samp
les_leaf': 1, 'min_samples_split': 2}
```

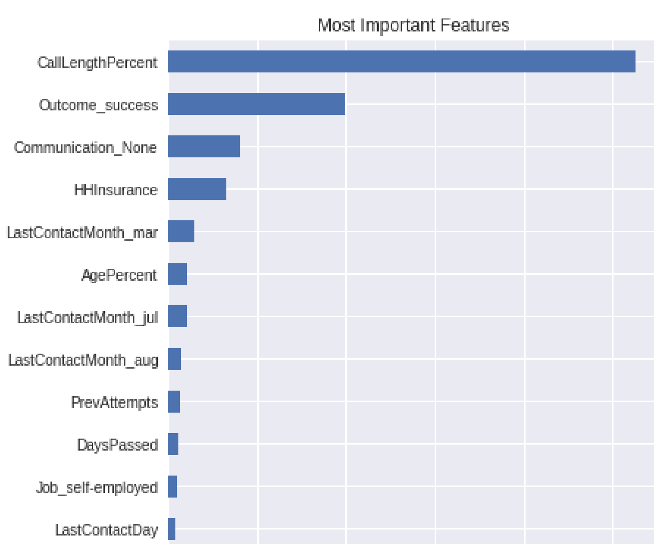
```
--- Best Model ----------------------------
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=
10,
            max_features=None, max_leaf_nodes=20,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=
None,
            splitter='best')
```

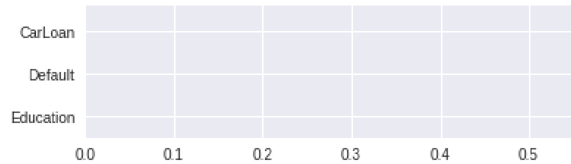
In [39]:

```
model_report(dt_best, dt_best.predict(x_test), y_test)
```

```
--- Train Set ----------------------------
Accuracy: 0.82821 +/- 0.0160
AUC: 0.88252 +/- 0.0157

--- Validation Set ----------------------------
Accuracy: 0.79080 +/- 0.0239
AUC: 0.84095 +/- 0.0101
-----------------------------------------------
```





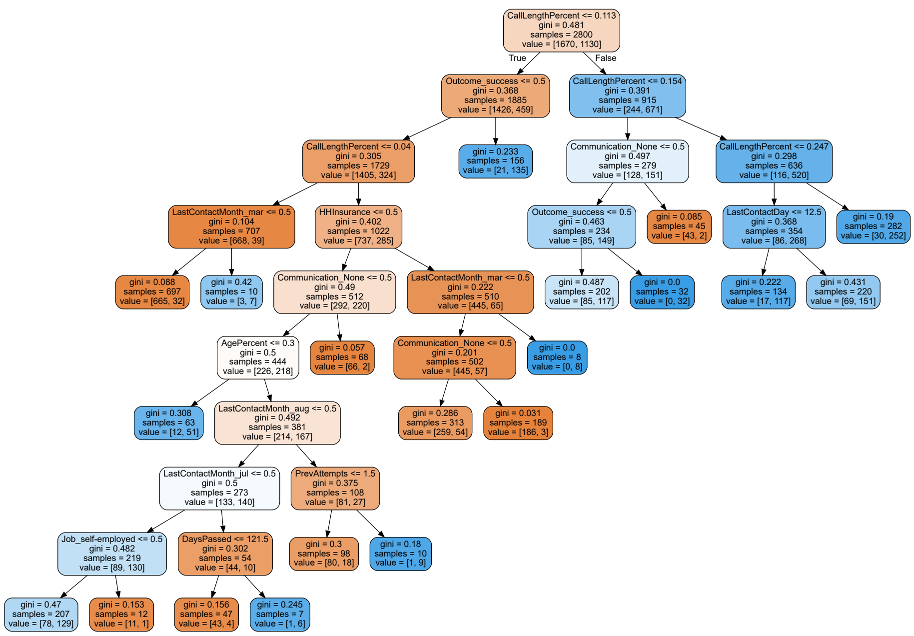


Plot Decision Tree and Save

In [40]:
```
dot_data = export_graphviz(dt_best, out_file=None, feature_names=list(
x_test.columns), filled=True, rounded=True,)
graph = graphviz.Source(dot_data)
```

In [41]:
```
graph
```

Out[41]:



K-Nearest Neighbors

In [42]:
```
knn = KNeighborsClassifier()
parameters = {'n_neighbors':[5,6,7],
              'p':[1,2],
              'weights':['uniform','distance']}
clf_knn = get_best_model(knn,parameters)
```

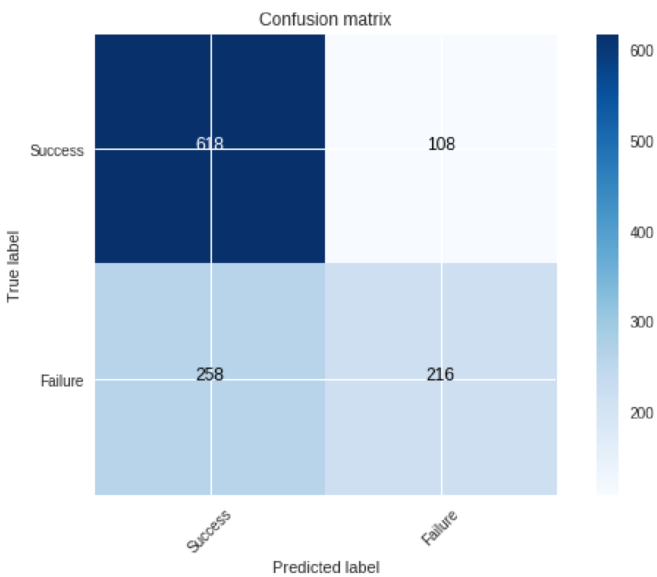
```
--- Best Parameters -----------------------------
{'n_neighbors': 7, 'p': 1, 'weights': 'distance'}

--- Best Model -----------------------------
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowsk
i',
```

```
                     metric_params=None, n_jobs=1, n_neighbors=7, p=1,
                     weights='distance')
```

In [43]:

```
model_report(clf_knn, clf_knn.predict(x_test), y_test, feature_imp=Fal
se)
```

```
--- Train Set -----------------------------
Accuracy: 0.69679 +/- 0.0223
AUC: 0.73094 +/- 0.0214

--- Validation Set -----------------------------
Accuracy: 0.65497 +/- 0.0132
AUC: 0.67858 +/- 0.0099
-----------------------------------------------
```



Naive Bayes

In [44]:

```
clf_nb = GaussianNB()
model_fit(model=clf_nb)
```

Out[44]:

```
{'clf': GaussianNB(priors=None),
 'x_test':             Education  Default  HHInsurance  CarLoan  LastC
ontactDay  \
 train 3626          3        0            0        0            12
       3310          3        0            1        1            28
       1142          2        0            0        0             7
       1767          1        0            0        0             7
       2645          3        0            0        0            14
       3756          2        0            1        1            13
       3087          3        0            0        0            14
       1105          2        0            0        0             8
       3852          2        0            0        0             2
       602           3        0            0        0            17
       1533          2        0            1        1            18
```

```
          1707      3      0      1      0      23
          250       3      0      1      0      11
          3216      2      0      0      0      22
          693       3      0      1      0      16
          2914      3      0      0      0       1
          3795      2      0      1      0       6
          2927      3      0      0      0      14
          1522      2      0      1      1      29
          883       2      0      0      1      17
          2622      2      0      0      0      18
          3416      3      0      0      0       1
          623       3      0      1      0      15
          2693      3      0      1      0       6
          198       3      0      0      0      16
          3625      3      0      0      0      20
          3657      3      0      1      0      23
          3947      2      0      1      0      13
          1240      1      0      0      0      13
          915       3      0      1      0      17
...               ...    ...    ...    ...     ...
          2830      1      0      1      0      14
          980       2      0      0      0      23
          1800      3      0      1      0      30
          459       3      0      0      0      30
          2065      3      0      0      0      21
          2705      2      0      0      1      29
          103       1      0      0      0       4
          895       3      0      0      0      30
          822       2      0      0      0      23
          3133      3      0      1      0      23
          3541      2      0      1      1      26
          3540      2      0      0      1      28
          412       1      0      1      0      18
          652       3      0      0      0      25
          1379      2      0      0      0      13
          340       2      0      0      0      20
          1300      2      0      0      0      17
          2554      3      0      0      0      20
          2838      2      0      1      0       7
          2359      3      0      1      0      17
          2287      1      1      0      1      29
          2940      3      0      1      1      28
          2370      2      0      1      0      29
          2310      3      0      0      0       4
          3394      2      0      1      0      20
          3566      2      0      0      0       6
          16        2      0      1      0       6
          2487      2      0      0      0      30
          1193      3      0      0      0       6
          443       1      0      0      0      21

            NoOfContacts  DaysPassed  PrevAttempts  CallStartHour  \
train 3626             1          -1             0             14
      3310             2         153             3             15
      1142             5          -1             0             14
      1767             7          -1             0              9
      2645             2          -1             0             15
      3756             3         370             4             15
```

```
      3087          2        -1          0          9
      1105          2        38         10          9
      3852          3        -1          0         13
      602           2        -1          0         17
      1533          5        -1          0         15
      1707          2       182          5         12
      250          11        -1          0         13
      3216         18        -1          0         11
      693           1       103          1         10
      2914          2        65          1         15
      3795          2        -1          0         16
      2927          5        -1          0         12
      1522         23        -1          0         17
      883           1        -1          0         11
      2622          3        -1          0         10
      3416          1        -1          0         15
      623           1        91          9         12
      2693          2        -1          0         13
      198           1        88          1         15
      3625          2        -1          0         16
      3657          2        -1          0          9
      3947          1       181          5         10
      1240          1        -1          0         15
      915           5        -1          0         14
...               ...       ...        ...        ...
      2830          2        -1          0         15
      980           1        -1          0         17
      1800          2         2          2         16
      459           2        -1          0         13
      2065          2        -1          0         11
      2705          2        -1          0          9
      103           2        -1          0         15
      895           1        87          1         15
      822           3        -1          0         14
      3133          1        -1          0         14
      3541          1        -1          0         17
      3540          3        -1          0         14
      412           1        -1          0         13
      652           1       544          2         16
      1379          3        -1          0         11
      340           3        -1          0         13
      1300          1        -1          0         16
      2554          2       150          2         11
      2838          1        -1          0         15
      2359          1        -1          0         17
      2287          6        -1          0         15
      2940          1        -1          0         17
      2370          2        -1          0          9
      2310          2        94          3         10
      3394          1        -1          0         12
      3566          1        -1          0         17
      16            3       362          4         11
      2487          4        -1          0         17
      1193          1        90          3         12
      443           1        -1          0         17

           CallLengthPercent      ...        LastContactMonth_jun  \
```

```
train 3626       0.107286    ...               0
      3310       0.089148    ...               0
      1142       0.075315    ...               0
      1767       0.061174    ...               0
      2645       0.214571    ...               0
      3756       0.292346    ...               0
      3087       0.154626    ...               0
      1105       0.138026    ...               0
      3852       0.025208    ...               0
      602        0.067937    ...               0
      1533       0.048263    ...               0
      1707       0.110974    ...               0
      250        0.115278    ...               1
      3216       0.006763    ...               0
      693        0.046726    ...               0
      2914       0.079926    ...               0
      3795       0.084845    ...               0
      2927       0.051952    ...               0
      1522       0.003074    ...               0
      883        0.072241    ...               0
      2622       0.030433    ...               1
      3416       0.031048    ...               0
      623        0.028589    ...               0
      2693       0.041500    ...               0
      198        0.116815    ...               0
      3625       0.283123    ...               0
      3657       0.121734    ...               0
      3947       0.094067    ...               0
      1240       0.158623    ...               0
      915        0.164771    ...               0
...                   ...    ...             ...
      2830       0.060559    ...               0
      980        0.090071    ...               0
      1800       0.123578    ...               0
      459        0.291731    ...               0
      2065       0.095297    ...               0
      2705       0.028282    ...               0
      103        0.189671    ...               0
      895        0.096834    ...               0
      822        0.048571    ...               0
      3133       0.013219    ...               0
      3541       0.058408    ...               1
      3540       0.145097    ...               0
      412        0.035352    ...               0
      652        0.139871    ...               0
      1379       0.116815    ...               0
      340        0.314479    ...               1
      1300       0.056871    ...               0
      2554       0.029511    ...               0
      2838       0.073778    ...               0
      2359       0.182601    ...               0
      2287       0.034430    ...               0
      2940       0.175838    ...               0
      2370       0.185060    ...               0
      2310       0.093760    ...               0
      3394       0.060867    ...               0
      3566       0.017830    ...               1
      16         0.028282    ...               0
```

```
       2487          0.112512       ...                            0
       1193          0.059945       ...                            0
       443           0.059945       ...                            0

             LastContactMonth_mar  LastContactMonth_may  LastContactMo
nth_nov  \
 train 3626                     0                     0
0
       3310                     0                     0
0
       1142                     0                     0
0
       1767                     0                     0
0
       2645                     0                     0
0
       3756                     0                     1
0
       3087                     0                     0
0
       1105                     0                     0
1
       3852                     0                     0
0
       602                      0                     0
0
       1533                     0                     1
0
       1707                     0                     0
0
       250                      0                     0
0
       3216                     0                     0
0
       693                      0                     0
0
       2914                     0                     0
0
       3795                     0                     0
0
       2927                     0                     0
0
       1522                     0                     0
0
       883                      0                     0
0
       2622                     0                     0
0
       3416                     0                     0
0
       623                      0                     0
0
       2693                     0                     0
0
       198                      0                     0
1
       3625                     0                     0
0
```

```
        3657                    0                    0
0
        3947                    0                    1
0
        1240                    0                    1
0
        915                     0                    0
0
 ...                         ...                  ...
...
        2830                    0                    1
0
        980                     0                    0
0
        1800                    0                    0
0
        459                     0                    0
0
        2065                    0                    0
0
        2705                    0                    0
0
        103                     0                    1
0
        895                     0                    0
0
        822                     0                    0
0
        3133                    0                    0
0
        3541                    0                    0
0
        3540                    0                    0
0
        412                     0                    1
0
        652                     0                    0
0
        1379                    0                    0
0
        340                     0                    0
0
        1300                    0                    0
0
        2554                    0                    0
0
        2838                    0                    1
0
        2359                    0                    0
0
        2287                    0                    0
0
        2940                    0                    0
0
        2370                    0                    0
0
        2310                    0                    0
0
```

```
       3394                    0                    0
0
       3566                    0                    0
0
       16                      0                    1
0
       2487                    0                    0
0
       1193                    0                    1
0
       443                     0                    0
0

             LastContactMonth_oct  LastContactMonth_sep  Outcome_NoPre
v  \
 train 3626                    0                    0
1
       3310                    0                    0
0
       1142                    0                    0
1
       1767                    0                    0
1
       2645                    0                    0
1
       3756                    0                    0
0
       3087                    0                    0
1
       1105                    0                    0
0
       3852                    0                    0
1
       602                     0                    0
1
       1533                    0                    0
1
       1707                    0                    0
0
       250                     0                    0
1
       3216                    0                    0
1
       693                     0                    1
0
       2914                    0                    0
0
       3795                    0                    0
1
       2927                    0                    0
1
       1522                    0                    0
1
       883                     0                    0
1
       2622                    0                    0
1
       3416                    1                    0
1
```

```
'
       623                        0                     0
0
       2693                       0                     0
1
       198                        0                     0
0
       3625                       0                     0
1
       3657                       0                     0
1
       3947                       0                     0
0
       1240                       0                     0
1
       915                        0                     0
1
 ...                           ...                   ...
...
       2830                       0                     0
1
       980                        0                     0
1
       1800                       0                     0
0
       459                        0                     0
1
       2065                       0                     0
1
       2705                       0                     0
1
       103                        0                     0
1
       895                        0                     0
0
       822                        0                     0
1
       3133                       0                     0
1
       3541                       0                     0
1
       3540                       0                     0
1
       412                        0                     0
1
       652                        0                     0
0
       1379                       0                     0
1
       340                        0                     0
1
       1300                       0                     0
1
       2554                       0                     0
0
       2838                       0                     0
1
       2359                       0                     0
1
```

```
       2287                 0                    0
1
       2940                 0                    0
1
       2370                 0                    0
1
       2310                 0                    0
0
       3394                 0                    0
1
       3566                 0                    0
1
       16                   0                    0
0
       2487                 0                    0
1
       1193                 0                    0
0
       443                  0                    0
1

             Outcome_failure  Outcome_other  Outcome_success
 train 3626                0              0                0
       3310                1              0                0
       1142                0              0                0
       1767                0              0                0
       2645                0              0                0
       3756                1              0                0
       3087                0              0                0
       1105                0              1                0
       3852                0              0                0
       602                 0              0                0
       1533                0              0                0
       1707                0              0                1
       250                 0              0                0
       3216                0              0                0
       693                 0              0                1
       2914                0              0                1
       3795                0              0                0
       2927                0              0                0
       1522                0              0                0
       883                 0              0                0
       2622                0              0                0
       3416                0              0                0
       623                 1              0                0
       2693                0              0                0
       198                 1              0                0
       3625                0              0                0
       3657                0              0                0
       3947                0              0                1
       1240                0              0                0
       915                 0              0                0
 ...                     ...            ...              ...
       2830                0              0                0
       980                 0              0                0
       1800                0              1                0
       459                 0              0                0
       2065                0              0                0
```

```
      2705         0         0         0
      103          0         0         0
      895          0         0         1
      822          0         0         0
      3133         0         0         0
      3541         0         0         0
      3540         0         0         0
      412          0         0         0
      652          1         0         0
      1379         0         0         0
      340          0         0         0
      1300         0         0         0
      2554         1         0         0
      2838         0         0         0
      2359         0         0         0
      2287         0         0         0
      2940         0         0         0
      2370         0         0         0
      2310         0         0         1
      3394         0         0         0
      3566         0         0         0
      16           0         1         0
      2487         0         0         0
      1193         0         1         0
      443          0         0         0

 [1200 rows x 46 columns],
 'x_train':            Education  Default  HHInsurance  CarLoan  Last
ContactDay  \
 train 3209         1        0            1        1           13
      3268         1        0            1        1            3
      2374         2        0            0        0           30
      885          2        0            0        0           19
      2102         3        0            0        0           12
      2790         2        0            0        0           19
      3178         2        0            1        0           29
      1970         2        0            0        0            9
      3206         2        0            0        0           28
      270          2        0            1        0            2
      1155         2        0            0        0           28
      3563         3        0            1        0           24
      586          2        0            1        0           30
      1120         2        0            1        1            9
      362          1        0            1        0           16
      2584         1        0            1        0            8
      2215         3        0            0        0           17
      3977         2        0            0        0            9
      3815         2        0            1        0           28
      3913         3        0            0        0            6
      1233         2        0            0        0           18
      1000         3        0            1        0           12
      837          2        0            1        1           20
      3214         3        0            0        0           23
      2911         2        0            0        0            4
      3444         2        0            1        0            9
      212          2        0            1        0           20
      131          3        0            0        0           28
      1807         3        0            0        1           20
```

```
      3935         1       0          1       0             8
...            ...     ...        ...     ...           ...
      834          3       0          1       0             4
      2710         2       0          1       0            11
      1498         2       0          0       0             6
      337          2       0          1       0            13
      3610         2       0          0       0            10
      3576         2       0          0       0            30
      2446         2       0          0       0             2
      1447         3       0          0       0            16
      2653         3       0          0       0             2
      1964         3       0          0       0            18
      1684         3       0          0       0            22
      2528         2       0          0       0            27
      3494         3       0          0       0            30
      1143         1       0          0       0            17
      2965         2       0          0       0            16
      3722         2       0          1       0            25
      1705         2       0          1       0            13
      3065         3       0          0       0            29
      2923         2       0          1       0            13
      1738         2       0          0       1            10
      2707         2       0          1       0            26
      3069         2       0          0       0            31
      789          2       0          1       0            30
      2304         3       0          0       0             4
      968          2       0          1       0             8
      3000         1       0          1       0             8
      1667         3       0          1       0            15
      3321         2       0          1       0             8
      1688         2       0          0       0            28
      1898         1       0          1       0            21

            NoOfContacts  DaysPassed  PrevAttempts  CallStartHour  \
train 3209             5          -1             0             11
      3268             1          -1             0             16
      2374             1          -1             0             14
      885              1          -1             0             17
      2102             2          95             4             12
      2790             8          -1             0             15
      3178             1          -1             0             11
      1970             1          -1             0             11
      3206             1          -1             0             16
      270              3          -1             0             12
      1155            17          -1             0             10
      3563             2          -1             0             12
      586              3          -1             0             12
      1120             2          -1             0             12
      362              2          -1             0             10
      2584             2          -1             0             15
      2215             1          -1             0             16
      3977             2          -1             0             17
      3815             6          -1             0             13
      3913             2          -1             0             13
      1233            11          -1             0              9
      1000             2          -1             0             16
      837              4          -1             0             13
      3214             3          -1             0             14
```

```
        2911          2      -1          0         10
        3444          1      -1          0         10
        212           2     350          1         13
        131           1      -1          0         14
        1807          3      -1          0         17
        3935          5      -1          0         13
...                 ...     ...        ...        ...
        834           1      -1          0         13
        2710          2      -1          0         11
        1498          1      -1          0         10
        337           1      -1          0         12
        3610          2      93          1         10
        3576          1      -1          0         10
        2446          2      -1          0         15
        1447          1      -1          0         10
        2653          1      -1          0         15
        1964          4      -1          0         14
        1684          1      -1          0         17
        2528          1      -1          0         10
        3494          1      -1          0         13
        1143          2      -1          0         12
        2965          1      -1          0         16
        3722          5      -1          0         12
        1705          1      -1          0         10
        3065         21      -1          0         17
        2923          1      -1          0         13
        1738          7      -1          0         15
        2707          1      -1          0         17
        3069          2      -1          0         15
        789           1      13          1         16
        2304          1      97          2         10
        968           1      -1          0         11
        3000          9      -1          0         17
        1667          3      -1          0         15
        3321          5      -1          0         16
        1688          2      -1          0         17
        1898          6      -1          0         12

              CallLengthPercent       ...        LastContactMonth_jun  \
train 3209             0.607132       ...                           0
      3268             0.127267       ...                           0
      2374             0.138949       ...                           0
      885              0.172456       ...                           0
      2102             0.073471       ...                           1
      2790             0.397479       ...                           0
      3178             0.084845       ...                           0
      1970             0.121426       ...                           0
      3206             0.055641       ...                           0
      270              0.025208       ...                           1
      1155             0.024285       ...                           0
      3563             0.185675       ...                           0
      586              0.029511       ...                           0
      1120             0.017522       ...                           0
      362              0.032278       ...                           0
      2584             0.075623       ...                           0
      2215             0.067630       ...                           0
      3977             0.131878       ...                           0
```

```
      3815        0.025515     ...                      0
      3913        0.053489     ...                      0
      1233        0.025822     ...                      0
      1000        0.080541     ...                      0
      837         0.014141     ...                      1
      3214        0.037504     ...                      0
      2911        0.042115     ...                      0
      3444        0.165386     ...                      0
      212         0.177990     ...                      0
      131         0.274516     ...                      0
      1807        0.172149     ...                      1
      3935        0.030741     ...                      0
...                    ...     ...                    ...
      834         0.109745     ...                      0
      2710        0.121119     ...                      1
      1498        0.098986     ...                      1
      337         0.318475     ...                      0
      3610        0.047341     ...                      0
      3576        0.040578     ...                      0
      2446        0.018137     ...                      1
      1447        0.054719     ...                      0
      2653        0.160160     ...                      0
      1964        0.011374     ...                      0
      1684        0.066093     ...                      0
      2528        0.040271     ...                      0
      3494        0.030433     ...                      0
      1143        0.040578     ...                      1
      2965        0.132493     ...                      1
      3722        0.040578     ...                      0
      1705        0.067015     ...                      0
      3065        0.078082     ...                      0
      2923        0.063019     ...                      0
      1738        0.412235     ...                      0
      2707        0.019059     ...                      0
      3069        0.045804     ...                      0
      789         0.059945     ...                      0
      2304        0.356901     ...                      1
      968         0.023056     ...                      0
      3000        0.005226     ...                      0
      1667        0.257608     ...                      0
      3321        0.091608     ...                      0
      1688        0.446050     ...                      0
      1898        0.312634     ...                      0

            LastContactMonth_mar  LastContactMonth_may  LastContactMo
nth_nov  \
 train 3209                    0                     1
0
      3268                     0                     0
0
      2374                     0                     0
0
      885                      0                     0
1
      2102                     0                     0
0
      2790                     0                     0
0
```

3178	0	1	0
1970	0	0	0
3206	0	1	0
270	0	0	0
1155	0	0	0
3563	0	0	0
586	0	1	0
1120	0	1	0
362	0	0	0
2584	0	1	0
2215	0	0	0
3977	0	0	0
3815	0	0	0
3913	0	0	0
1233	0	0	0
1000	0	1	0
837	0	0	0
3214	0	0	0
2911	0	0	0
3444	0	1	0
212	0	0	0
131	0	0	0
1807	0	0	0
3935	0	1	0
...	...	...	...
834	0	0	0
2710	0	0	0
1498	0	0	0
337	0	0	0
3610	0	0	

```
       3610                   0                    0
0
       3576                   0                    0
0
       2446                   0                    0
0
       1447                   0                    1
0
       2653                   0                    0
0
       1964                   0                    0
0
       1684                   0                    0
0
       2528                   0                    0
0
       3494                   0                    0
0
       1143                   0                    0
0
       2965                   0                    0
0
       3722                   0                    0
0
       1705                   0                    1
0
       3065                   0                    0
0
       2923                   0                    1
0
       1738                   0                    0
0
       2707                   0                    1
0
       3069                   0                    0
0
       789                    0                    0
0
       2304                   0                    0
0
       968                    0                    1
0
       3000                   0                    1
0
       1667                   0                    0
0
       3321                   0                    1
0
       1688                   0                    0
0
       1898                   0                    0
0

             LastContactMonth_oct  LastContactMonth_sep  Outcome_NoPre
v  \
 train 3209                   0                    0
1
       3268                   0                    0
1
```

```
.
        2374            0            0
1
        885             0            0
1
        2102            0            0
0
        2790            0            0
1
        3178            0            0
1
        1970            0            0
1
        3206            0            0
1
        270             0            0
1
        1155            0            0
1
        3563            0            0
1
        586             0            0
1
        1120            0            0
1
        362             0            0
1
        2584            0            0
1
        2215            0            0
1
        3977            0            0
1
        3815            0            0
1
        3913            0            0
1
        1233            0            0
1
        1000            0            0
1
        837             0            0
1
        3214            0            0
1
        2911            0            0
1
        3444            0            0
1
        212             0            0
0
        131             0            0
1
        1807            0            0
1
        3935            0            0
1
 ...                   ...          ...
...
```

834	0	0	1
2710	0	0	1
1498	0	0	1
337	0	0	1
3610	0	0	0
3576	1	0	1
2446	0	0	1
1447	0	0	1
2653	0	0	1
1964	0	0	1
1684	0	0	1
2528	1	0	1
3494	0	0	1
1143	0	0	1
2965	0	0	1
3722	0	0	1
1705	0	0	1
3065	0	0	1
2923	0	0	1
1738	0	0	1
2707	0	0	1
3069	0	0	1
789	0	0	0
2304	0	0	0
968	0	0	1
3000	0	0	1
1667	0	0	1
3321	0	0	1
1688	0	0	1

```
        1898                    0                    0
1

              Outcome_failure  Outcome_other  Outcome_success
train 3209                  0              0                0
      3268                  0              0                0
      2374                  0              0                0
      885                   0              0                0
      2102                  0              0                1
      2790                  0              0                0
      3178                  0              0                0
      1970                  0              0                0
      3206                  0              0                0
      270                   0              0                0
      1155                  0              0                0
      3563                  0              0                0
      586                   0              0                0
      1120                  0              0                0
      362                   0              0                0
      2584                  0              0                0
      2215                  0              0                0
      3977                  0              0                0
      3815                  0              0                0
      3913                  0              0                0
      1233                  0              0                0
      1000                  0              0                0
      837                   0              0                0
      3214                  0              0                0
      2911                  0              0                0
      3444                  0              0                0
      212                   1              0                0
      131                   0              0                0
      1807                  0              0                0
      3935                  0              0                0
...                       ...            ...              ...
      834                   0              0                0
      2710                  0              0                0
      1498                  0              0                0
      337                   0              0                0
      3610                  0              0                1
      3576                  0              0                0
      2446                  0              0                0
      1447                  0              0                0
      2653                  0              0                0
      1964                  0              0                0
      1684                  0              0                0
      2528                  0              0                0
      3494                  0              0                0
      1143                  0              0                0
      2965                  0              0                0
      3722                  0              0                0
      1705                  0              0                0
      3065                  0              0                0
      2923                  0              0                0
      1738                  0              0                0
      2707                  0              0                0
      3069                  0              0                0
      789                   1              0                0
```

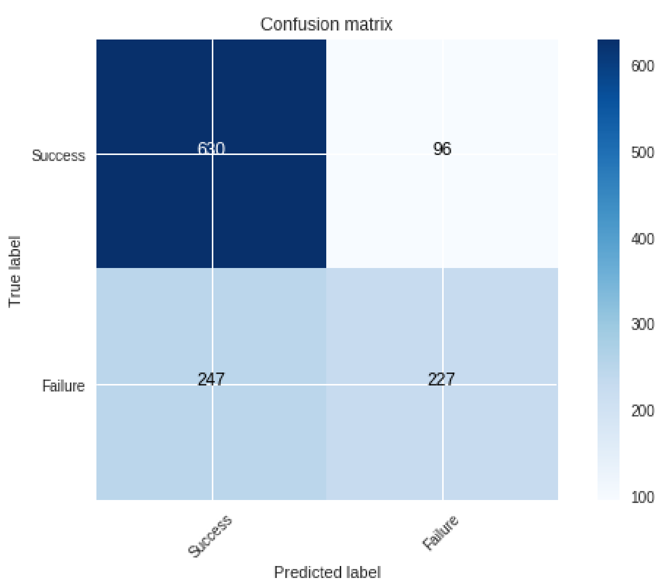
```
      2304             0            0                1
      968              0            0                0
      3000             0            0                0
      1667             0            0                0
      3321             0            0                0
      1688             0            0                0
      1898             0            0                0

[2800 rows x 46 columns],
'y_pred': array([0, 0, 1, ..., 0, 0, 0]),
'y_train': 3209    1
3268    0
2374    1
885     1
2102    1
2790    1
3178    0
1970    1
3206    0
270     0
1155    0
3563    0
586     0
1120    0
362     0
2584    0
2215    1
3977    0
3815    0
3913    0
1233    0
1000    0
837     0
3214    0
2911    0
3444    1
212     0
131     1
1807    1
3935    0
       ..
834     0
2710    0
1498    0
337     1
3610    1
3576    1
2446    0
1447    0
2653    1
1964    0
1684    0
2528    1
3494    0
1143    0
2965    0
3722    0
1705    0
```

```
3065    0
2923    0
1738    1
2707    0
3069    0
789     0
2304    1
968     0
3000    0
1667    1
3321    0
1688    1
1898    1
Name: CarInsurance, Length: 2800, dtype: int64}
```

In [45]:

```
model_report(clf_nb, clf_nb.predict(x_test), y_test, feature_imp=False
)
```

```
--- Train Set -----------------------------
Accuracy: 0.72964 +/- 0.0203
AUC: 0.79652 +/- 0.0115

--- Validation Set -----------------------------
Accuracy: 0.72416 +/- 0.0125
AUC: 0.76857 +/- 0.0135
-----------------------------------------------
```



Logistic Regression

In [46]:

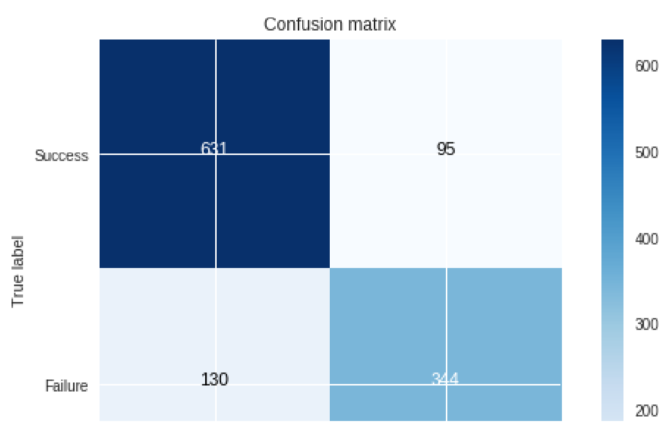
```
lg = LogisticRegression(random_state=3)
parameters = {'C':[0.8,0.9,1],
              'penalty':['l1','l2']}
clf_lg = get_best_model(lg,parameters)
```

```
--- Best Parameters -----------------------------
{'C': 0.8, 'penalty': 'l1'}

--- Best Model -----------------------------
LogisticRegression(C=0.8, class_weight=None, dual=False, fit_intercept
=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs
=1,
          penalty='l1', random_state=3, solver='liblinear', tol=0.000
1,
          verbose=0, warm_start=False)
```

In [47]:

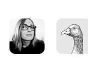
```
model_report(clf_lg, clf_lg.predict(x_test), y_test, feature_imp=False
)
```

```
--- Train Set -----------------------------
Accuracy: 0.82179 +/- 0.0121
AUC: 0.90752 +/- 0.0106

--- Validation Set -----------------------------
Accuracy: 0.79585 +/- 0.0161
AUC: 0.89650 +/- 0.0117
-----------------------------------------------
```



Did you find this Kernel useful?
Show your appreciation with an upvote

2

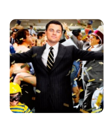


Data

Data Sources

Car Insurance Cold Calls

carInsurance_test.csv 19 columns



Car Insurance Cold Calls

We help the guys and girls at the front to get out of Cold Call Hell

Last Updated: 2 years ago (Version 1)

carInsurance_train.csv 19 columns

DSS_DMC_Description.pdf

About this Dataset

Introduction

Here you find a very simple, beginner-friendly data set. No sparse matrices, no fancy tools needed to understand what's going on. Just a couple of rows and columns. Super simple stuff. As explained below, this data set is used for a competition. As it turns out, this competition tends to reveal a common truth in data science: KISS - Keep It Simple Stupid

What is so special about this data set is, given it's simplicity, it pays off to use "simple" classifiers as well. This year's competition was won by a C5.0 . Can you do better?

Description

We are looking at cold call results. Turns out, same salespeople called existing insurance customers up and tried to sell car insurance. What you have are details about the called customers. Their age, job, marital status, whether the have home insurance, a car loan, etc. As I said, super simple.

What I would love to see is some of you applying some crazy XGBoost classifiers, which we can square off against some logistic regressions. It would be curious to see what comes out on top. Thank you for your time, I hope you enjoy using the data set.

Acknowledgements

Thanks goes to the Decision Science and Systems Chair of Technical University of Munich (TUM) for getting the data set

Comments (0)

Click here to comment...