

```
...,
[ 68., 144., 193., ..., 0., 0., 1.],
[ 57., 130., 131., ..., 0., 0., 1.],
[ 57., 130., 236., ..., 1., 0., 0.]]
```

When we use `get_dummies`, if we do not supply a list of categorical variables using the `columns` option, it will look for columns that have an `object` or `category dtype` and encode them. Since some of the categories in this data set actually have integer data types we needed to indicate them specifically, or they would not be encoded.³

We will examine the distribution of classification errors, taking the classification error as one minus the accuracy score. The following function `evaluate` will perform a requested number of train/test splits and then return a list of all of the errors for further processing.

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
def evaluate(X, Y, MODEL, nsplits=100, PRINT=True):
    errs=[]
    for j in range(nsplits):
        XTRAIN,XTEST,YTRAIN,YTEST=train_test_split(X,Y)
        model=MODEL
        model.fit(XTRAIN,YTRAIN)
        YP=model.predict(XTEST)
        errs.append(1-accuracy_score(YTEST,YP))
    if PRINT:
        print("Mean error=%7.5f std=%7.5f" \
              %(np.mean(errs), np.std(errs)))
    return errs
```

We need to load all of the required methods.

```
from sklearn.linear_model import LogisticRegression as LR
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier as KNN
from sklearn.discriminant_analysis import \
    LinearDiscriminantAnalysis as LDA
from sklearn.discriminant_analysis import \
    QuadraticDiscriminantAnalysis as QDA
from sklearn.svm import SVC
from sklearn.neural_network import MLPClassifier as ANN
from sklearn.ensemble import RandomForestClassifier as RF
from sklearn.tree import DecisionTreeClassifier as DT
```

³For documentation see https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.get_dummies.html