Brock University Library
Digital Scholarship Lab

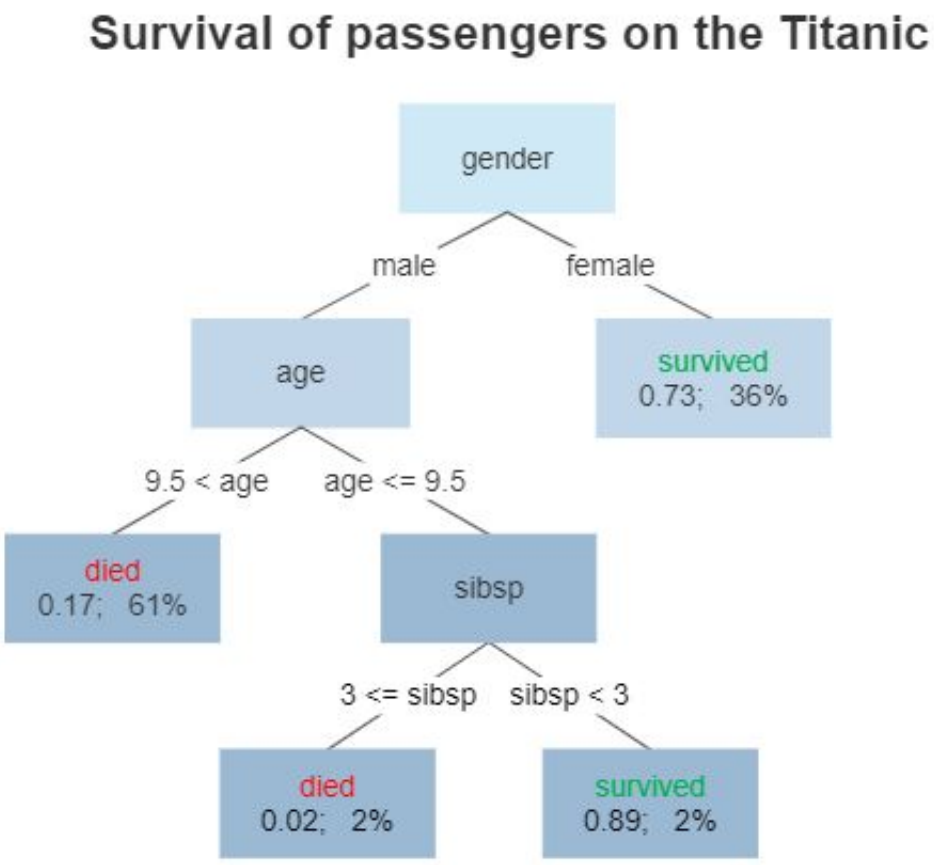
Machine Learning with Python: An (Easy) Introduction

Connect to the Workshop `http://bit.ly/dslmachine`

Steps Involved in a Machine Learning Project

1. Getting your data and cleaning it up
2. Identify what parts of your data are **features**
3. Identify what is your **target variable** that you'll guess based on your features
4. Split your data in **training and testing sets**
5. **Train** your model against the training set
6. **Validate** your model against the testing set

Scikit and Decision Trees



Libraries to Include

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import tree
```

Loading and formating the data

```
data =
pd.read_csv("https://brockdsl.github.io/Python_2.0_Workshop/canadian_toy_dataset.
csv")
data.columns = ["city","gender","age","income","ill"]

#Instead of yes/no we'll use a 0 or 1
data["ill"].replace({"No":0, "Yes":1},inplace=True)

#We change categorical values in numeric ones using `dummies`
data = pd.get_dummies(data, columns=['city','gender'])
```

Building and Running the Model

```
#all of our `indication` columns are features
features = ["age",\
            "income",\
            "city_Edmonton",\
            "city_Halifax",\
            "city_Montreal",
            "city_Ottawa",\
            "city_Regina",\
            "city_Toronto",
            "city_Vancouver",\
            "city_Waterloo",\
            "gender_Female",\
            "gender_Male"]
X = data[features]

#We want to target the ill column
y = data.ill
```

Training and Testing

```
#Training and test together make up 100% of the data!
test_percent = 30
train_percent = 100 - test_percent

X_train, X_test, y_train, y_test = train_test_split(X, \
                                                    y, \
                                                    test_size=test_percent/100.0,
                                                   random_state=10)

# Create Decision Tree classifer object
treeClass = DecisionTreeClassifier()

# Train
treeClass = treeClass.fit(X_train,y_train)

#Predict
y_pred = treeClass.predict(X_test)
```

Checking our model

```
from sklearn import metrics
metrics.accuracy_score(y_test,y_pred)
```

Links

A Gentle Introduction to Scikit-Learn

Data Science Handbook by Field Cady

Deep Learning with Python