Decision Trees & scikit-learn

Sarah Jane Delany

Click to edit Master subtitle style

scikit-learn - the problem with category data

- Category data must be converted to numbers
  - **`LabelEncoder`** does this
    - but not as we would want

```
label_encoder = LabelEncoder()
labelE = df.apply(label_encoder.fit_transform)
labelE
```

	Pet	Trans	Gender
0	cat	bike	Female
1	dog	car	Female
2	cat	car	Male
3	ferret	bike	Female

	Pet	Trans	Gender
0	0	0	0
1	1	1	0
2	0	1	1
3	2	0	0

OneHot Encoding

- Two options:
  - Use sklearn **`OneHotEncoder`**
    - Creates a transformation object
    - Can be applied to other data
  - Use pandas **`get_dummies`** function

	Pet	Trans	Gender
0	cat	bike	Female
1	dog	car	Female
2	cat	car	Male
3	ferret	bike	Female

```
df = pd.get_dummies(df,drop_first=True)
```

	Pet_cat	Pet_dog	Pet_ferret	Transp_bike	Transt_car	Gender_Female	Gender_Male
0	1	0	0	1	0	1	0
1	0	1	0	0	1	1	0
2	1	0	0	0	1	0	1
3	0	0	1	1	0	1	0

Pandas get_dummies

- A column for each category is not necessary

```
df = pd.get_dummies(df)
```

	Pet_cat	Pet_dog	Pet_ferret	Transport_bike	Transport_car	Gender_Female	Gender_Male
0	1	0	0	1	0	1	0
1	0	1	0	0	1	1	0
2	1	0	0	0	1	0	1
3	0	0	1	1	0	1	0

```
df = pd.get_dummies(df,drop_first=True)
```

	Pet_dog	Pet_ferret	Transport_car	Gender_Male
0	0	0	0	0
1	1	0	1	0
2	0	0	1	1
3	0	1	0	0

One-Hot Encoding

- sklearn **`OneHotEncoder`** produces a numpy array

```
from sklearn.preprocessing import OneHotEncoder
onehot_encoder = OneHotEncoder(sparse=False)
dfOH = onehot_encoder.fit_transform(df)
dfOH
Out[14]:
array([[1., 0., 0., 1., 0., 1., 0.],
       [0., 1., 0., 0., 1., 1., 0.],
       [1., 0., 0., 0., 1., 0., 1.],
       [0., 0., 1., 1., 0., 1., 0.]])
In [15]:
onehot_encoder.get_feature_names()
Out[15]:
array(['x0_cat', 'x0_dog', 'x0_ferret', 'x1_bike', 'x1_car', 'x2_Female',
       'x2_Male'], dtype=object)
```

Important: we have a handle for a OneHotEncoder object that can be applied to other data

Restaurant data

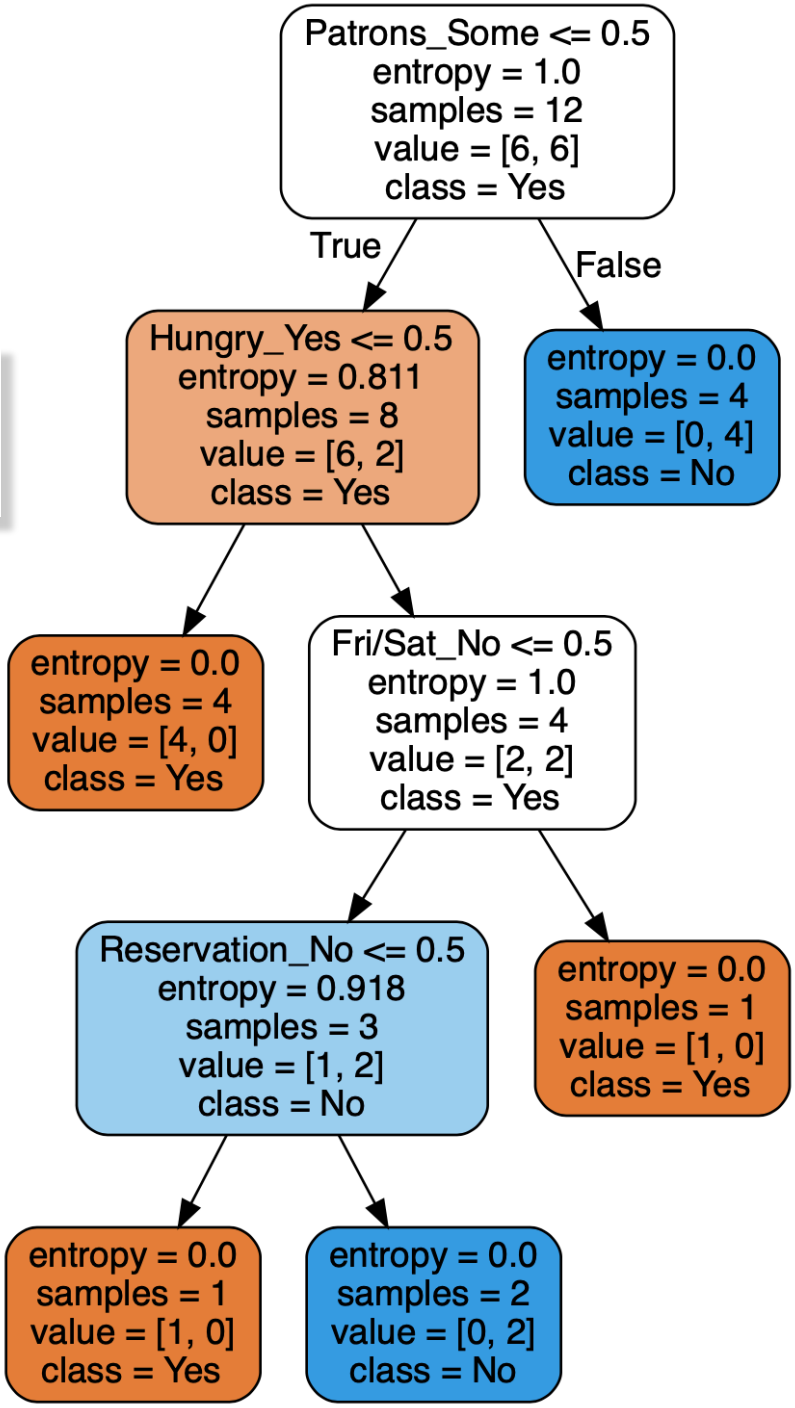
No	Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Raining	Reserv	Type	WaitEst	WillWait?
1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	Yes
2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	No
3	No	Yes	No	No	Some	$	No	No	Burger	0-10	Yes
4	Yes	No	Yes	Yes	Full	$	No	No	Thai	10-30	Yes
5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	No

```
onehot_encoder = OneHotEncoder(sparse=False)
restOH = onehot_encoder.fit(restaurant)
restOH_data = restOH.transform(restaurant)

restOH.get_feature_names(restaurant.columns)
Out[21]:
array(['Alternate_No', 'Alternate_Yes', 'Bar_No', 'Bar_Yes', 'Fri/Sat_No',
       'Fri/Sat_Yes', 'Hungry_No', 'Hungry_Yes', 'Patrons_Full',
       'Patrons_None', 'Patrons_Some', 'Price_$', 'Price_$$', 'Price_$$$',
       'Raining_No', 'Raining_Yes', 'Reservation_No', 'Reservation_Yes',
       'Type_Burger', 'Type_French', 'Type_Italian', 'Type_Thai',
       'WaitEst_0-10', 'WaitEst_10-30', 'WaitEst_30-60', 'WaitEst_>60'],
      dtype=object)
```

Restaurant data

```
rtree = DecisionTreeClassifier(
                        criterion='entropy')
rtreeOH = rtree.fit(restOH_data,y)
```



Sadly: these OneHotEncoded trees are really hard to read.