4_categorical_data_exercise

May 20, 2019

```
In [ ]: import pandas as pd
        import numpy as np

In [ ]: edges = pd.DataFrame({'source': [0, 1, 2],
                              'target': [2, 2, 3],
                                  'weight': [3, 4, 5],
                                  'color': ['red', 'blue', 'blue']})

        edges

In [ ]: edges.dtypes

In [ ]: edges["color"]

In [ ]: pd.get_dummies(edges)

In [ ]:

In [ ]: pd.get_dummies(edges["color"])

In [ ]: pd.get_dummies(edges[["color"]])

In [ ]: weight_dict = {3:"M", 4:"L", 5:"XL"}
        edges["weight_sign"] = edges["weight"].map(weight_dict)
        edges

In [ ]: weight_sign = pd.get_dummies(edges["weight_sign"])
        weight_sign

In [ ]: pd.concat([edges, weight_sign], axis=1)

In [ ]: pd.get_dummies(edges).values

In [ ]: # Example from - https://chrisalbon.com/python/pandas_binning_data.html

        raw_data = {'regiment': ['Nighthawks', 'Nighthawks', 'Nighthawks', 'Nighthawks', \
                                 'Dragoons', 'Dragoons', 'Dragoons', 'Dragoons', 'Scouts',\
                                 'Scouts', 'Scouts', 'Scouts'],
                'company': ['1st', '1st', '2nd', '2nd', '1st', '1st', '2nd', '2nd',\
```

```
                        '1st', '1st', '2nd', '2nd'],
                'name': ['Miller', 'Jacobson', 'Ali', 'Milner', 'Cooze', 'Jacon', 'Ryaner',\
                         'Sone', 'Sloan', 'Piger', 'Riani', 'Ali'],
                'preTestScore': [4, 24, 31, 2, 3, 4, 24, 31, 2, 3, 2, 3],
                'postTestScore': [25, 94, 57, 62, 70, 25, 94, 57, 62, 70, 62, 70]}
        df = pd.DataFrame(raw_data, columns = ['regiment', 'company', 'name', 'preTestScore', '
                                               'postTestScore'])
        df
```

```
In [ ]: bins = [0, 25, 50, 75, 100] # Define bins as 0 to 25, 25 to 50, 60 to 75, 75 to 100
        group_names = ['Low', 'Okay', 'Good', 'Great']
        categories = pd.cut(df['postTestScore'], bins, labels=group_names)
        categories
```

```
In [ ]: df['categories'] = pd.cut(df['postTestScore'], bins, labels=group_names)
        pd.value_counts(df['categories'])
```

```
In [ ]: pd.get_dummies(df)
```

```
In [ ]:
```

0.0.1 using scikit-learn preprocessing

```
In [ ]: raw_example = df.as_matrix()
        raw_example[:10]
```

```
In [ ]: data = raw_example.copy()
```

```
In [ ]: from sklearn import preprocessing
        le = preprocessing.LabelEncoder()
```

```
In [ ]: raw_example[:,0]
```

```
In [ ]: le.fit(raw_example[:,0])
```

```
In [ ]: le.classes_
```

```
In [ ]: le.transform(raw_example[:,0])
```

```
In [ ]: data[:,0] = le.transform(raw_example[:,0])
        data[:3]
```

```
In [ ]: label_column = [0,1,2,5]
        label_enconder_list = []
        for column_index in  label_column:
            le = preprocessing.LabelEncoder()
            le.fit(raw_example[:,column_index])
            data[:,column_index] = le.transform(raw_example[:,column_index])
            label_enconder_list.append(le)
            del le
        data[:3]
```

```
In [ ]: label_enconder_list[0].transform(raw_example[:10,0])

In [ ]: one_hot_enc = preprocessing.OneHotEncoder()
        data[:,0].reshape(-1,1)

In [ ]: one_hot_enc.fit(data[:,0].reshape(-1,1))

In [ ]: one_hot_enc.n_values_

In [ ]: one_hot_enc.active_features_

In [ ]: data[:,0].reshape(-1,1)

In [ ]: onehotlabels = one_hot_enc.transform(data[:,0].reshape(-1,1)).toarray()
        onehotlabels

In [ ]:

In [ ]:
```