ML_6_XGBoost

March 19, 2019

```
In [1]: """In this assignment students need to predict whether a person makes over 50K per yea
        or not from classic adult dataset using XGBoost. The description of the dataset is as
        follows:"""

        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
        %matplotlib inline
        train_set = pd.read_csv('http://archive.ics.uci.edu/ml/machine-learning-databases/adul
        test_set = pd.read_csv('http://archive.ics.uci.edu/ml/machine-learning-databases/adult
        col_labels = ['age', 'workclass', 'fnlwgt', 'education', 'education_num', 'marital_sta
        'native_country', 'wage_class']
        train_set.columns = col_labels
        test_set.columns = col_labels
        train_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
age               32561 non-null int64
workclass         32561 non-null object
fnlwgt            32561 non-null int64
education         32561 non-null object
education_num     32561 non-null int64
marital_status    32561 non-null object
occupation        32561 non-null object
relationship      32561 non-null object
race              32561 non-null object
sex               32561 non-null object
capital_gain      32561 non-null int64
capital_loss      32561 non-null int64
hours_per_week    32561 non-null int64
native_country    32561 non-null object
wage_class        32561 non-null object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

```
In [3]: # data cleaning

        df = pd.concat([train_set,test_set],axis=0)
        df['wage_class'] = df['wage_class'].apply(lambda x : 1 if x=='>50K' else 0) # converti
        df.head()
```

```
Out[3]:    age         workclass  fnlwgt  education  education_num  \
        0   39         State-gov   77516  Bachelors             13
        1   50  Self-emp-not-inc   83311  Bachelors             13
        2   38           Private  215646    HS-grad              9
        3   53           Private  234721       11th              7
        4   28           Private  338409  Bachelors             13

               marital_status         occupation   relationship   race     sex  \
        0        Never-married       Adm-clerical  Not-in-family  White    Male
        1   Married-civ-spouse    Exec-managerial        Husband  White    Male
        2             Divorced  Handlers-cleaners  Not-in-family  White    Male
        3   Married-civ-spouse  Handlers-cleaners        Husband  Black    Male
        4   Married-civ-spouse     Prof-specialty           Wife  Black  Female

           capital_gain  capital_loss  hours_per_week native_country  wage_class
        0          2174             0              40  United-States           0
        1             0             0              13  United-States           0
        2             0             0              40  United-States           0
        3             0             0              40  United-States           0
        4             0             0              40           Cuba           0
```

```
In [5]: # Remove unknown values and spaces

        df.replace(' ?', np.nan, inplace=True)
        for col in df.columns:
            if type(df[col][0]) == str:
                print("Working on " + col)
                df[col] = df[col].apply(lambda val: val.replace(" ",""))
        df.head()
```

```
Out[5]:    age         workclass  fnlwgt  education  education_num  \
        0   39         State-gov   77516  Bachelors             13
        1   50  Self-emp-not-inc   83311  Bachelors             13
        2   38           Private  215646    HS-grad              9
        3   53           Private  234721       11th              7
        4   28           Private  338409  Bachelors             13

               marital_status         occupation   relationship   race     sex  \
        0        Never-married       Adm-clerical  Not-in-family  White    Male
        1   Married-civ-spouse    Exec-managerial        Husband  White    Male
        2             Divorced  Handlers-cleaners  Not-in-family  White    Male
        3   Married-civ-spouse  Handlers-cleaners        Husband  Black    Male
        4   Married-civ-spouse      Prof-specialty           Wife    Black    Female

           capital_gain  capital_loss  hours_per_week  native_country  wage_class
        0          2174             0              40   United-States           0
        1             0             0              13   United-States           0
        2             0             0              40   United-States           0
        3             0             0              40   United-States           0
        4             0             0              40            Cuba           0
```

```
In [7]: # Convert categorical variables into int.

        df = pd.concat([df, pd.get_dummies(df['workclass'],prefix='workclass',prefix_sep=':')]
        df.drop('workclass',axis=1,inplace=True)

        df = pd.concat([df, pd.get_dummies(df['marital_status'],prefix='marital_status',prefix_
        df.drop('marital_status',axis=1,inplace=True)

        df = pd.concat([df, pd.get_dummies(df['occupation'],prefix='occupation',prefix_sep=':'
        df.drop('occupation',axis=1,inplace=True)

        df = pd.concat([df, pd.get_dummies(df['relationship'],prefix='relationship',prefix_sep
        df.drop('relationship',axis=1,inplace=True)

        df = pd.concat([df, pd.get_dummies(df['race'],prefix='race',prefix_sep=':')], axis=1)
        df.drop('race',axis=1,inplace=True)

        df = pd.concat([df, pd.get_dummies(df['sex'],prefix='sex',prefix_sep=':')], axis=1)
        df.drop('sex',axis=1,inplace=True)

        df = pd.concat([df, pd.get_dummies(df['native_country'],prefix='native_country',prefix_
        df.drop('native_country',axis=1,inplace=True)

        df.drop('education', axis=1,inplace=True)

        df.head()
```

```
Out[7]:    age  fnlwgt  education_num  capital_gain  capital_loss  hours_per_week  \
        0   39   77516             13          2174             0              40
        1   50   83311             13             0             0              13
        2   38  215646              9             0             0              40
        3   53  234721              7             0             0              40
        4   28  338409             13             0             0              40

           wage_class  workclass: Federal-gov  workclass: Local-gov  \
        0           0                       0                     0
        1           0                       0                     0
        2           0                       0                     0
        3           0                       0                     0
```

```
4           0                       0                     0

   workclass: Never-worked             ...              \
0                        0             ...
1                        0             ...
2                        0             ...
3                        0             ...
4                        0             ...

   native_country: Portugal  native_country: Puerto-Rico  \
0                         0                            0
1                         0                            0
2                         0                            0
3                         0                            0
4                         0                            0

   native_country: Scotland  native_country: South  native_country: Taiwan  \
0                         0                      0                       0
1                         0                      0                       0
2                         0                      0                       0
3                         0                      0                       0
4                         0                      0                       0

   native_country: Thailand  native_country: Trinadad&Tobago  \
0                         0                                0
1                         0                                0
2                         0                                0
3                         0                                0
4                         0                                0

   native_country: United-States  native_country: Vietnam  \
0                              1                        0
1                              1                        0
2                              1                        0
3                              1                        0
4                              0                        0

   native_country: Yugoslavia
0                           0
1                           0
2                           0
3                           0
4                           0

[5 rows x 98 columns]
```

```
In [12]: # Splitting into dependent and independent variables.
```

```
from sklearn import preprocessing

X = np.array(df.drop(['wage_class'], 1))
y = np.array(df['wage_class'])
X = preprocessing.scale(X)    #Standard Scaling

#Splitting into train and test part

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# Applying XGBoost Classifier model

import xgboost as xgb
from sklearn.metrics import accuracy_score

model = xgb.XGBClassifier(learning_rate=0.1,
                           n_estimators=500,
                           max_depth=5,
                           min_child_weight=4
                           )
model.fit(X_train, y_train)
predictions = model.predict(X_test)
XGBA = accuracy_score(y_test, predictions)
print("The Accuracy  is {}".format(XGBA))
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:595: DataConversionWarni
  warnings.warn(msg, DataConversionWarning)
```

```
The Accuracy  is 1.0
```