Big_Mart_Sales_Project

January 21, 2020

[1]:

```python
import numpy as np
import pandas as pd
from pandas import Series, DataFrame
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
from sklearn.metrics import accuracy_score
# import test and train file
train = pd.read_csv('mart_train.csv')
test = pd.read_csv('mart_test.csv')
```

[2]:

```python
train.head()
```

[2]:

```
  Item_Identifier  Item_Weight Item_Fat_Content  Item_Visibility  \
0           FDA15         9.30          Low Fat         0.016047
1           DRC01         5.92          Regular         0.019278
2           FDN15        17.50          Low Fat         0.016760
3           FDX07        19.20          Regular         0.000000
4           NCD19         8.93          Low Fat         0.000000

               Item_Type  Item_MRP Outlet_Identifier  \
0                  Dairy  249.8092            OUT049
1            Soft Drinks   48.2692            OUT018
2                   Meat  141.6180            OUT049
3  Fruits and Vegetables  182.0950            OUT010
4              Household   53.8614            OUT013

   Outlet_Establishment_Year Outlet_Size Outlet_Location_Type  \
0                       1999      Medium               Tier 1
1                       2009      Medium               Tier 3
2                       1999      Medium               Tier 1
3                       1998         NaN               Tier 3
4                       1987        High               Tier 3

         Outlet_Type  Item_Outlet_Sales
0  Supermarket Type1          3735.1380
1  Supermarket Type2           443.4228
```

```
2  Supermarket Type1          2097.2700
3      Grocery Store           732.3800
4  Supermarket Type1           994.7052
```

[3]:
```
train['Item_Fat_Content'].value_counts()
```

[3]:
```
Low Fat    5089
Regular    2889
LF          316
reg         117
low fat     112
Name: Item_Fat_Content, dtype: int64
```

[4]:
```
test.head()
```

[4]:
```
  Item_Identifier  Item_Weight Item_Fat_Content  Item_Visibility    Item_Type  \
0           FDW58       20.750          Low Fat         0.007565  Snack Foods
1           FDW14        8.300              reg         0.038428        Dairy
2           NCN55       14.600          Low Fat         0.099575       Others
3           FDQ58        7.315          Low Fat         0.015388  Snack Foods
4           FDY38          NaN          Regular         0.118599        Dairy

   Item_MRP Outlet_Identifier  Outlet_Establishment_Year Outlet_Size  \
0  107.8622            OUT049                       1999      Medium
1   87.3198            OUT017                       2007         NaN
2  241.7538            OUT010                       1998         NaN
3  155.0340            OUT017                       2007         NaN
4  234.2300            OUT027                       1985      Medium

  Outlet_Location_Type        Outlet_Type
0               Tier 1  Supermarket Type1
1               Tier 2  Supermarket Type1
2               Tier 3      Grocery Store
3               Tier 2  Supermarket Type1
4               Tier 3  Supermarket Type3
```

[5]:
```
# importing linear regression from sklearn
from sklearn.linear_model import LinearRegression
lreg = LinearRegression()
# Import LabelEncoder
from sklearn import preprocessing
#creating labelEncoder
le = preprocessing.LabelEncoder()
# Converting train data string labels into numbers and filling Na values of␣
 ↪Item_Weight By Mean Values According to Fat_Content.
train['Outlet_Location_Type'] = le.fit_transform(train['Outlet_Location_Type'])

train['Item_Fat_Content'].replace(['LF','reg','low fat'],['Low␣
 ↪Fat','Regular','Low Fat'],inplace = True)
```

```
train['Item_Weight'] = train.groupby('Item_Fat_Content')['Item_Weight'].
 ↪transform(lambda x: x.fillna(x.mean()))
train['Item_Fat_Content'] = le.fit_transform(train['Item_Fat_Content'])
# Converting test data string labels into numbers and filling Na values of␣
 ↪Item_Weight By Mean Values According to Fat_Content.
test['Outlet_Location_Type'] = le.fit_transform(test['Outlet_Location_Type'])

test['Item_Fat_Content'].replace(['LF','reg','low fat'],['Low␣
 ↪Fat','Regular','Low Fat'],inplace = True)
test['Item_Weight'] = test.groupby('Item_Fat_Content')['Item_Weight'].
 ↪transform(lambda x: x.fillna(x.mean()))
test['Item_Fat_Content'] = le.fit_transform(test['Item_Fat_Content'])
```

[6]:

```
#splitting into training and cv for cross validation
X = train.loc[:
 ↪,['Outlet_Establishment_Year','Item_Visibility','Outlet_Location_Type','Item_Weight','Item_
X1 = test.loc[:
 ↪,['Outlet_Establishment_Year','Item_Visibility','Outlet_Location_Type','Item_Weight','Item_
x_train = X
y_train = train['Item_Outlet_Sales']
x_cv = X1

# training the model
lreg.fit(x_train,y_train)
# predicting on cv
pred = lreg.predict(x_cv)
# Writing pred values in solution file
test['Item_Outlet_Sales'] = pred
test.to_csv("solution.csv")
```

[7]:

```
solution = pd.read_csv('solution.csv')
solution.head()
```

[7]:

```
   Unnamed: 0 Item_Identifier  Item_Weight  Item_Fat_Content  Item_Visibility  \
0           0           FDW58    20.750000                 0         0.007565
1           1           FDW14     8.300000                 1         0.038428
2           2           NCN55    14.600000                 0         0.099575
3           3           FDQ58     7.315000                 0         0.015388
4           4           FDY38    12.394528                 1         0.118599

     Item_Type  Item_MRP Outlet_Identifier  Outlet_Establishment_Year  \
0  Snack Foods  107.8622            OUT049                       1999
1        Dairy   87.3198            OUT017                       2007
2       Others  241.7538            OUT010                       1998
3  Snack Foods  155.0340            OUT017                       2007
4        Dairy  234.2300            OUT027                       1985

  Outlet_Size  Outlet_Location_Type        Outlet_Type  Item_Outlet_Sales
```

```
0      Medium                     0  Supermarket Type1        1676.288689
1         NaN                     1  Supermarket Type1        1403.109730
2         NaN                     2      Grocery Store        3722.079225
3         NaN                     1  Supermarket Type1        2480.455810
4      Medium                     2  Supermarket Type3        3748.898724
```