In [1]:

```
%reload_ext autoreload
%autoreload 2
%matplotlib inline
```

In [2]:

```
from fastai.imports import *
from fastai.structured import *
```

```
/Users/saran/anaconda3/lib/python3.6/site-
packages/sklearn/ensemble/weight_boosting.py:29: DeprecationWarning: numpy.core.umath
_tests is an internal NumPy module and should not be imported. It will be removed in
a future NumPy release.
  from numpy.core.umath_tests import inner1d
```

In [3]:

```
from sklearn import metrics
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from datetime import datetime
```

In [4]:

```
pd.set_option('display.max_columns', 500)
pd.set_option('display.max_rows', 500)
```

In [5]:

```
PATH = 'Datasets/'
```

In [6]:

```
df_raw = pd.read_csv(f'{PATH}train.csv', low_memory=False, parse_dates=[1], index_col
=0)
```

In [7]:

```
df_raw.head().T
```

Out[7]:

Id	0	1	2	3	4
Open Date	1999-07-17 00:00:00	2008-02-14 00:00:00	2013-03-09 00:00:00	2012-02-02 00:00:00	2009-05-09 00:00:00
City	İstanbul	Ankara	Diyarbakır	Tokat	Gaziantep
City Group	Big Cities	Big Cities	Other	Other	Other

Type Id	IL 0	FC 1	IL 2	IL 3	IL 4
P1	4	4	2	6	3
P2	5	5	4	4.5	4
P3	4	4	2	6	3
P4	4	4	5	6	4
P5	2	1	2	4	2
P6	2	2	3	4	2
P7	5	5	5	10	5
P8	4	5	5	8	5
P9	5	5	5	10	5
P10	5	5	5	10	5
P11	3	1	2	8	2
P12	5	5	5	10	5
P13	5	5	5	7.5	5
P14	1	0	0	6	2
P15	2	0	0	4	1
P16	2	0	0	9	2
P17	2	0	0	3	1
P18	4	0	0	12	4
P19	5	3	1	20	2
P20	4	2	1	12	2
P21	1	1	1	6	1
P22	3	3	1	1	2
P23	3	2	1	10	1
P24	1	0	0	2	2
P25	1	0	0	2	3
P26	1	0	0	2.5	3
P27	4	0	0	2.5	5
P28	2	3	1	2.5	1
P29	3	3	3	7.5	3
P30	5	0	0	25	5
P31	3	0	0	12	1
P32	4	0	0	10	3

Id	0	1	2	3	4
P33	5	0	0	6	2
P34	5	0	0	18	3
P35	4	0	0	12	4
P36	3	0	0	12	3
P37	4	0	0	6	3
revenue	5.65375e+06	6.92313e+06	2.05538e+06	2.67551e+06	4.31672e+06

In [8]:

```
df_raw.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 137 entries, 0 to 136
Data columns (total 42 columns):
Open Date     137 non-null datetime64[ns]
City          137 non-null object
City Group    137 non-null object
Type          137 non-null object
P1            137 non-null int64
P2            137 non-null float64
P3            137 non-null float64
P4            137 non-null float64
P5            137 non-null int64
P6            137 non-null int64
P7            137 non-null int64
P8            137 non-null int64
P9            137 non-null int64
P10           137 non-null int64
P11           137 non-null int64
P12           137 non-null int64
P13           137 non-null float64
P14           137 non-null int64
P15           137 non-null int64
P16           137 non-null int64
P17           137 non-null int64
P18           137 non-null int64
P19           137 non-null int64
P20           137 non-null int64
P21           137 non-null int64
P22           137 non-null int64
P23           137 non-null int64
P24           137 non-null int64
P25           137 non-null int64
P26           137 non-null float64
P27           137 non-null float64
P28           137 non-null float64
P29           137 non-null float64
P30           137 non-null int64
P31           137 non-null int64
P32           137 non-null int64
```

```
P33             137 non-null int64
P34             137 non-null int64
P35             137 non-null int64
P36             137 non-null int64
P37             137 non-null int64
revenue         137 non-null float64
dtypes: datetime64[ns](1), float64(9), int64(29), object(3)
memory usage: 46.0+ KB
```

In [9]:

```
df_raw.shape
```

Out[9]:

```
(137, 42)
```

In [10]:

```
df_raw["YearsOpen"] = (datetime.now()-df_raw['Open Date']).astype('timedelta64[D]')/365
```

In [11]:

```
df_raw.head().T
```

Out[11]:

Id	0	1	2	3	4
Open Date	1999-07-17 00:00:00	2008-02-14 00:00:00	2013-03-09 00:00:00	2012-02-02 00:00:00	2009-05-09 00:00:00
City	İstanbul	Ankara	Diyarbakır	Tokat	Gaziantep
City Group	Big Cities	Big Cities	Other	Other	Other
Type	IL	FC	IL	IL	IL
P1	4	4	2	6	3
P2	5	5	4	4.5	4
P3	4	4	2	6	3
P4	4	4	5	6	4
P5	2	1	2	4	2
P6	2	2	3	4	2
P7	5	5	5	10	5
P8	4	5	5	8	5
P9	5	5	5	10	5
P10	5	5	5	10	5

Id	0	1	2	3	4
P11	3	1	2	8	2
P12	5	5	5	10	5
P13	5	5	5	7.5	5
P14	1	0	0	6	2
P15	2	0	0	4	1
P16	2	0	0	9	2
P17	2	0	0	3	1
P18	4	0	0	12	4
P19	5	3	1	20	2
P20	4	2	1	12	2
P21	1	1	1	6	1
P22	3	3	1	1	2
P23	3	2	1	10	1
P24	1	0	0	2	2
P25	1	0	0	2	3
P26	1	0	0	2.5	3
P27	4	0	0	2.5	5
P28	2	3	1	2.5	1
P29	3	3	3	7.5	3
P30	5	0	0	25	5
P31	3	0	0	12	1
P32	4	0	0	10	3
P33	5	0	0	6	2
P34	5	0	0	18	3
P35	4	0	0	12	4
P36	3	0	0	12	3
P37	4	0	0	6	3
revenue	5.65375e+06	6.92313e+06	2.05538e+06	2.67551e+06	4.31672e+06
YearsOpen	19.1315	10.5452	5.47671	6.57534	9.31233

In [12]:

```
df_raw.drop(columns=['Open Date'], inplace=True)
```

In [13]:

```
df_raw.head().T
```

Out[13]:

Id	0	1	2	3	4
City	İstanbul	Ankara	Diyarbakır	Tokat	Gaziantep
City Group	Big Cities	Big Cities	Other	Other	Other
Type	IL	FC	IL	IL	IL
P1	4	4	2	6	3
P2	5	5	4	4.5	4
P3	4	4	2	6	3
P4	4	4	5	6	4
P5	2	1	2	4	2
P6	2	2	3	4	2
P7	5	5	5	10	5
P8	4	5	5	8	5
P9	5	5	5	10	5
P10	5	5	5	10	5
P11	3	1	2	8	2
P12	5	5	5	10	5
P13	5	5	5	7.5	5
P14	1	0	0	6	2
P15	2	0	0	4	1
P16	2	0	0	9	2
P17	2	0	0	3	1
P18	4	0	0	12	4
P19	5	3	1	20	2
P20	4	2	1	12	2
P21	1	1	1	6	1
P22	3	3	1	1	2
P23	3	2	1	10	1
P24	1	0	0	2	2
P25	1	0	0	2	3
P26	1	0	0	2.5	3
P27	4	0	0	2.5	5

Id	0	1	2	3	4
P28	2	3	1	2.5	1
P29	3	3	3	7.5	3
P30	5	0	0	25	5
P31	3	0	0	12	1
P32	4	0	0	10	3
P33	5	0	0	6	2
P34	5	0	0	18	3
P35	4	0	0	12	4
P36	3	0	0	12	3
P37	4	0	0	6	3
revenue	5.65375e+06	6.92313e+06	2.05538e+06	2.67551e+06	4.31672e+06
YearsOpen	19.1315	10.5452	5.47671	6.57534	9.31233

In [14]:

```
#train_cats(df_raw)
df_raw['Type'] = LabelEncoder().fit_transform(df_raw['Type'])
df_raw['City Group'] = LabelEncoder().fit_transform(df_raw['City Group'])
```

In [15]:

```
df_raw.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 137 entries, 0 to 136
Data columns (total 42 columns):
City           137 non-null object
City Group     137 non-null int64
Type           137 non-null int64
P1             137 non-null int64
P2             137 non-null float64
P3             137 non-null float64
P4             137 non-null float64
P5             137 non-null int64
P6             137 non-null int64
P7             137 non-null int64
P8             137 non-null int64
P9             137 non-null int64
P10            137 non-null int64
P11            137 non-null int64
P12            137 non-null int64
P13            137 non-null float64
P14            137 non-null int64
P15            137 non-null int64
P16            137 non-null int64
P17            137 non-null int64
P18            137 non-null int64
```

```
P18             137 non-null int64
P19             137 non-null int64
P20             137 non-null int64
P21             137 non-null int64
P22             137 non-null int64
P23             137 non-null int64
P24             137 non-null int64
P25             137 non-null int64
P26             137 non-null float64
P27             137 non-null float64
P28             137 non-null float64
P29             137 non-null float64
P30             137 non-null int64
P31             137 non-null int64
P32             137 non-null int64
P33             137 non-null int64
P34             137 non-null int64
P35             137 non-null int64
P36             137 non-null int64
P37             137 non-null int64
revenue         137 non-null float64
YearsOpen       137 non-null float64
dtypes: float64(10), int64(31), object(1)
memory usage: 46.0+ KB
```

In [16]:

```
df_raw.drop(columns=['City'], inplace=True)
```

In [17]:

```
df_raw.apply(lambda x: x.isnull().values.sum())
```

Out[17]:

```
City Group     0
Type           0
P1             0
P2             0
P3             0
P4             0
P5             0
P6             0
P7             0
P8             0
P9             0
P10            0
P11            0
P12            0
P13            0
P14            0
P15            0
P16            0
P17            0
```

```
P18           0
P19           0
P20           0
P21           0
P22           0
P23           0
P24           0
P25           0
P26           0
P27           0
P28           0
P29           0
P30           0
P31           0
P32           0
P33           0
P34           0
P35           0
P36           0
P37           0
revenue       0
YearsOpen     0
dtype: int64
```

In [18]:

```
os.makedirs(f'{PATH}/tmp', exist_ok=True)
```

In [19]:

```
df_raw = df_raw.reset_index()
```

In [20]:

```
df_raw.to_feather(f'{PATH}tmp/TFI.ft')
```

In [21]:

```
df_raw = pd.read_feather(f'{PATH}tmp/TFI.ft')
```

In [22]:

```
df_raw.head().T
```

Out[22]:

	0	1	2	3	4
Id	0.000000e+00	1.000000e+00	2.000000e+00	3.000000e+00	4.000000e+00
City Group	0.000000e+00	0.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
Type	2.000000e+00	1.000000e+00	2.000000e+00	2.000000e+00	2.000000e+00

	0	1	2	3	4
P1	4.000000e+00	4.000000e+00	2.000000e+00	6.000000e+00	3.000000e+00
P2	5.000000e+00	5.000000e+00	4.000000e+00	4.500000e+00	4.000000e+00
P3	4.000000e+00	4.000000e+00	2.000000e+00	6.000000e+00	3.000000e+00
P4	4.000000e+00	4.000000e+00	5.000000e+00	6.000000e+00	4.000000e+00
P5	2.000000e+00	1.000000e+00	2.000000e+00	4.000000e+00	2.000000e+00
P6	2.000000e+00	2.000000e+00	3.000000e+00	4.000000e+00	2.000000e+00
P7	5.000000e+00	5.000000e+00	5.000000e+00	1.000000e+01	5.000000e+00
P8	4.000000e+00	5.000000e+00	5.000000e+00	8.000000e+00	5.000000e+00
P9	5.000000e+00	5.000000e+00	5.000000e+00	1.000000e+01	5.000000e+00
P10	5.000000e+00	5.000000e+00	5.000000e+00	1.000000e+01	5.000000e+00
P11	3.000000e+00	1.000000e+00	2.000000e+00	8.000000e+00	2.000000e+00
P12	5.000000e+00	5.000000e+00	5.000000e+00	1.000000e+01	5.000000e+00
P13	5.000000e+00	5.000000e+00	5.000000e+00	7.500000e+00	5.000000e+00
P14	1.000000e+00	0.000000e+00	0.000000e+00	6.000000e+00	2.000000e+00
P15	2.000000e+00	0.000000e+00	0.000000e+00	4.000000e+00	1.000000e+00
P16	2.000000e+00	0.000000e+00	0.000000e+00	9.000000e+00	2.000000e+00
P17	2.000000e+00	0.000000e+00	0.000000e+00	3.000000e+00	1.000000e+00
P18	4.000000e+00	0.000000e+00	0.000000e+00	1.200000e+01	4.000000e+00
P19	5.000000e+00	3.000000e+00	1.000000e+00	2.000000e+01	2.000000e+00
P20	4.000000e+00	2.000000e+00	1.000000e+00	1.200000e+01	2.000000e+00
P21	1.000000e+00	1.000000e+00	1.000000e+00	6.000000e+00	1.000000e+00
P22	3.000000e+00	3.000000e+00	1.000000e+00	1.000000e+00	2.000000e+00
P23	3.000000e+00	2.000000e+00	1.000000e+00	1.000000e+01	1.000000e+00
P24	1.000000e+00	0.000000e+00	0.000000e+00	2.000000e+00	2.000000e+00
P25	1.000000e+00	0.000000e+00	0.000000e+00	2.000000e+00	3.000000e+00
P26	1.000000e+00	0.000000e+00	0.000000e+00	2.500000e+00	3.000000e+00
P27	4.000000e+00	0.000000e+00	0.000000e+00	2.500000e+00	5.000000e+00
P28	2.000000e+00	3.000000e+00	1.000000e+00	2.500000e+00	1.000000e+00
P29	3.000000e+00	3.000000e+00	3.000000e+00	7.500000e+00	3.000000e+00
P30	5.000000e+00	0.000000e+00	0.000000e+00	2.500000e+01	5.000000e+00
P31	3.000000e+00	0.000000e+00	0.000000e+00	1.200000e+01	1.000000e+00
P32	4.000000e+00	0.000000e+00	0.000000e+00	1.000000e+01	3.000000e+00
P33	5.000000e+00	0.000000e+00	0.000000e+00	6.000000e+00	2.000000e+00
P34	5.000000e+00	0.000000e+00	0.000000e+00	1.800000e+01	3.000000e+00

	0	1	2	3	4
P35	4.000000e+00	0.000000e+00	0.000000e+00	1.200000e+01	4.000000e+00
P36	3.000000e+00	0.000000e+00	0.000000e+00	1.200000e+01	3.000000e+00
P37	4.000000e+00	0.000000e+00	0.000000e+00	6.000000e+00	3.000000e+00
revenue	5.653753e+06	6.923131e+06	2.055379e+06	2.675511e+06	4.316715e+06
YearsOpen	1.913151e+01	1.054521e+01	5.476712e+00	6.575342e+00	9.312329e+00

In [23]:

```
y = df_raw['revenue'].copy()
```

In [24]:

```
df = df_raw.drop(columns=['revenue'])
```

In [25]:

```
df.head().T
```

Out[25]:

	0	1	2	3	4
Id	0.000000	1.000000	2.000000	3.000000	4.000000
City Group	0.000000	0.000000	1.000000	1.000000	1.000000
Type	2.000000	1.000000	2.000000	2.000000	2.000000
P1	4.000000	4.000000	2.000000	6.000000	3.000000
P2	5.000000	5.000000	4.000000	4.500000	4.000000
P3	4.000000	4.000000	2.000000	6.000000	3.000000
P4	4.000000	4.000000	5.000000	6.000000	4.000000
P5	2.000000	1.000000	2.000000	4.000000	2.000000
P6	2.000000	2.000000	3.000000	4.000000	2.000000
P7	5.000000	5.000000	5.000000	10.000000	5.000000
P8	4.000000	5.000000	5.000000	8.000000	5.000000
P9	5.000000	5.000000	5.000000	10.000000	5.000000
P10	5.000000	5.000000	5.000000	10.000000	5.000000
P11	3.000000	1.000000	2.000000	8.000000	2.000000
P12	5.000000	5.000000	5.000000	10.000000	5.000000
P13	5.000000	5.000000	5.000000	7.500000	5.000000
P14	1.000000	0.000000	0.000000	6.000000	2.000000
P15	2.000000	0.000000	0.000000	4.000000	1.000000

P16	2.0000000	0.0000001	0.000000	9.0000003	2.000000
P17	2.000000	0.000000	0.000000	3.000000	1.000000
P18	4.000000	0.000000	0.000000	12.000000	4.000000
P19	5.000000	3.000000	1.000000	20.000000	2.000000
P20	4.000000	2.000000	1.000000	12.000000	2.000000
P21	1.000000	1.000000	1.000000	6.000000	1.000000
P22	3.000000	3.000000	1.000000	1.000000	2.000000
P23	3.000000	2.000000	1.000000	10.000000	1.000000
P24	1.000000	0.000000	0.000000	2.000000	2.000000
P25	1.000000	0.000000	0.000000	2.000000	3.000000
P26	1.000000	0.000000	0.000000	2.500000	3.000000
P27	4.000000	0.000000	0.000000	2.500000	5.000000
P28	2.000000	3.000000	1.000000	2.500000	1.000000
P29	3.000000	3.000000	3.000000	7.500000	3.000000
P30	5.000000	0.000000	0.000000	25.000000	5.000000
P31	3.000000	0.000000	0.000000	12.000000	1.000000
P32	4.000000	0.000000	0.000000	10.000000	3.000000
P33	5.000000	0.000000	0.000000	6.000000	2.000000
P34	5.000000	0.000000	0.000000	18.000000	3.000000
P35	4.000000	0.000000	0.000000	12.000000	4.000000
P36	3.000000	0.000000	0.000000	12.000000	3.000000
P37	4.000000	0.000000	0.000000	6.000000	3.000000
YearsOpen	19.131507	10.545205	5.476712	6.575342	9.312329

In [26]:

```
model = RandomForestRegressor(n_jobs=-1)
```

In [27]:

```
model.fit(df, y)
```

Out[27]:

```
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,
           max_features='auto', max_leaf_nodes=None,
           min_impurity_decrease=0.0, min_impurity_split=None,
           min_samples_leaf=1, min_samples_split=2,
           min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=-1,
           oob_score=False, random_state=None, verbose=0, warm_start=False)
```

In [28]:

```
model.score(df, y)
```

Out[28]:

```
0.8322358961786487
```

In [29]:

```
X_temp, X_test, y_temp, y_test = train_test_split(df, y, test_size=0.2, random_state=
42)
```

In [30]:

```
X_train, X_val, y_train, y_val = train_test_split(X_temp, y_temp, test_size=0.2, rand
om_state=42)
```

In [31]:

```
X_train.shape, X_val.shape, X_test.shape
```

Out[31]:

```
((87, 41), (22, 41), (28, 41))
```

In [32]:

```
y_train.shape, y_val.shape, y_test.shape
```

Out[32]:

```
((87,), (22,), (28,))
```

In [33]:

```
model = RandomForestRegressor(n_jobs=-1, n_estimators=250, max_features=2)
```

In [34]:

```
%time model.fit(X_train, y_train)
```

```
CPU times: user 280 ms, sys: 132 ms, total: 413 ms
Wall time: 370 ms
```

Out[34]:

```
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,
           max_features=2, max_leaf_nodes=None, min_impurity_decrease=0.0,
           min_impurity_split=None, min_samples_leaf=1,
           min_samples_split=2, min_weight_fraction_leaf=0.0,
           n_estimators=250, n_jobs=-1, oob_score=False, random_state=None,
           verbose=0, warm_start=False)
```

In [35]:

```
model.score(X_train, y_train)
```

Out[35]:

```
0.8645887242479635
```

In [36]:

```
model.score(X_val, y_val)
```

Out[36]:

```
0.013965005208134706
```

In [37]:

```
model.score(X_test, y_test)
```

Out[37]:

```
0.07594714470024222
```