Bulldozer_Bluebook_fastai_deep_learning

March 4, 2019

In [0]:
```
import pandas as pd
import re
from sklearn.ensemble import RandomForestRegressor, RandomForestClassifier
from IPython.display import display
import numpy as np
import math
from sklearn import metrics
from pandas.api.types import is_string_dtype, is_numeric_dtype
import matplotlib.pyplot as plt
from sklearn.ensemble import forest
import scipy
from scipy.cluster import hierarchy as hc
from fastai.tabular import *
```

In [0]:
```
#Put at beginning of every notebook to map Google drive to Colab
from google.colab import drive
drive.mount('/content/gdrive', force_remount=True)
root_dir = "/content/gdrive/My Drive/"
base_dir = root_dir + 'fastai-v3/'
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-

Enter your authorization code:
ůůůůůůůůůů
Mounted at /content/gdrive
```

In [0]:
```
path = Path(base_dir + 'data/bulldozer')
```

In [0]:
```
def rmse(x,y):
    return math.sqrt(((x-y)**2).mean())

def print_score(m):
    res = [rmse(m.predict(X_train), y_train), rmse(m.predict(X_valid), y_valid),
                m.score(X_train, y_train), m.score(X_valid, y_valid)]
    if hasattr(m, 'oob_score_'): res.append(m.oob_score_)
    print(res)
```

```python
def split_vals(a,n):
    return a[:n].copy(), a[n:].copy()

def get_oob(df):
    m = RandomForestRegressor(n_estimators=40, min_samples_leaf=5, max_features=0.6, n_
    x, _ = split_vals(df, n_trn)
    m.fit(x, y_train)
    return m.oob_score_

def add_datepart(df, fldname, drop=True, time=False):
    fld = df[fldname]
    fld_dtype = fld.dtype
    if isinstance(fld_dtype, pd.core.dtypes.dtypes.DatetimeTZDtype):
        fld_dtype = np.datetime64

    if not np.issubdtype(fld_dtype, np.datetime64):
        df[fldname] = fld = pd.to_datetime(fld, infer_datetime_format=True)
    targ_pre = re.sub('[Dd]ate$', '', fldname)
    attr = ['Year', 'Month', 'Week', 'Day', 'Dayofweek', 'Dayofyear',
            'Is_month_end', 'Is_month_start', 'Is_quarter_end', 'Is_quarter_start', 'Is
    if time: attr = attr + ['Hour', 'Minute', 'Second']
    for n in attr: df[targ_pre + n] = getattr(fld.dt, n.lower())
    df[targ_pre + 'Elapsed'] = fld.astype(np.int64) // 10 ** 9
    if drop: df.drop(fldname, axis=1, inplace=True)

def train_cats(df):
    for n,c in df.items():
        if is_string_dtype(c): df[n] = c.astype('category').cat.as_ordered()

def fix_missing(df, col, name, na_dict):
    if is_numeric_dtype(col):
        if pd.isnull(col).sum() or (name in na_dict):
            df[name+'_na'] = pd.isnull(col)
            filler = na_dict[name] if name in na_dict else col.median()
            df[name] = col.fillna(filler)
            na_dict[name] = filler
    return na_dict

def proc_df(df, y_fld=None, skip_flds=None, ignore_flds=None, do_scale=False, na_dict=N
            preproc_fn=None, max_n_cat=None, subset=None, mapper=None):
    if not ignore_flds: ignore_flds=[]
    if not skip_flds: skip_flds=[]
    if subset: df = get_sample(df,subset)
    else: df = df.copy()
    ignored_flds = df.loc[:, ignore_flds]
    df.drop(ignore_flds, axis=1, inplace=True)
    if preproc_fn: preproc_fn(df)
    if y_fld is None: y = None
```

```
    else:
        if not is_numeric_dtype(df[y_fld]): df[y_fld] = df[y_fld].cat.codes
        y = df[y_fld].values
        skip_flds += [y_fld]
    df.drop(skip_flds, axis=1, inplace=True)

    if na_dict is None: na_dict = {}
    else: na_dict = na_dict.copy()
    na_dict_initial = na_dict.copy()
    for n,c in df.items(): na_dict = fix_missing(df, c, n, na_dict)
    if len(na_dict_initial.keys()) > 0:
        df.drop([a + '_na' for a in list(set(na_dict.keys()) - set(na_dict_initial.keys
    if do_scale: mapper = scale_vars(df, mapper)
    for n,c in df.items(): numericalize(df, c, n, max_n_cat)
    df = pd.get_dummies(df, dummy_na=True)
    df = pd.concat([ignored_flds, df], axis=1)
    res = [df, y, na_dict]
    if do_scale: res = res + [mapper]
    return res

def numericalize(df, col, name, max_n_cat):
    if not is_numeric_dtype(col) and ( max_n_cat is None or col.nunique()>max_n_cat):
        df[name] = col.cat.codes+1
```

In [0]:
```
train_df = pd.read_csv(path/'Train.csv', low_memory=False, parse_dates=['saledate'])
```

In [0]:
```
train_df.shape
```

Out[0]: (401125, 53)

In [0]:
```
#Do some pre-processing
#Change SalePrice to log because the evaluation is for RMSLE
train_df.SalePrice = np.log(train_df.SalePrice)
#Change dates to date parts
add_datepart(train_df, 'saledate')
#Add a column for age of bulldozer
train_df['age'] = train_df['saleYear'] - train_df['YearMade']
```

In [0]:
```
dep_var = 'SalePrice'
cat_names = ['SalesID','MachineID', 'ModelID', 'datasource', 'auctioneerID', 'YearMade
       'fiModelDescriptor', 'ProductSize', 'fiProductClassDesc', 'state', 'ProductGrou
       'Turbocharged', 'Blade_Extension', 'Blade_Width', 'Enclosure_Type', 'Engine_Hor
       'Grouser_Tracks', 'Hydraulics_Flow', 'Track_Type', 'Undercarriage_Pad_Width', '
       'Differential_Type', 'Steering_Controls', 'saleYear', 'saleMonth', 'saleWeek',
       'saleIs_quarter_start', 'saleIs_year_end', 'saleIs_year_start']
cont_names = ['MachineHoursCurrentMeter', 'saleElapsed', 'age']
```

In [0]:
```
#Change string variables to category type
train_cats(df_raw)
```

```
#Specify order for variable UsageBand and change to codes
df_raw.UsageBand.cat.set_categories(['High', 'Medium', 'Low'], ordered=True, inplace=Tr
df_raw.UsageBand = df_raw.UsageBand.cat.codes
#Change categories to code and missing values to 0, replace missing numeric values with
#add column to indicate replaced missing values and separate the dependent variable as
df, y, nas = proc_df(df_raw, 'SalePrice')
```

```
---------------------------------------------------------------------------

NameError                                 Traceback (most recent call last)

<ipython-input-9-afbec81aab7f> in <module>()
----> 1 train_cats(df_raw)
      2 #Specify order for variable UsageBand and change to codes
      3 df_raw.UsageBand.cat.set_categories(['High', 'Medium', 'Low'], ordered=True, inplac
      4 df_raw.UsageBand = df_raw.UsageBand.cat.codes
      5 #Change categories to code and missing values to 0, replace missing numeric values


NameError: name 'train_cats' is not defined
```

In [0]:
```
df = pd.DataFrame({'col1': [1, 2, 3], 'col2': ['a', 'b', 'a'], 'col3': [0.5, 1.2, 7.5]
df
```

Out[0]:

	col1	col2	col3	col4
0	1	a	0.5	ab
1	2	b	1.2	o
2	3	a	7.5	o

In [0]:
```
path = untar_data(URLs.ADULT_SAMPLE)
df = pd.read_csv(path/'adult.csv')
train_df, valid_df = df.iloc[:800].copy(), df.iloc[800:1000].copy()
train_df.head()
```

Out[0]:

	age	workclass	fnlwgt	education	education-num \
0	49	Private	101320	Assoc-acdm	12.0
1	44	Private	236746	Masters	14.0
2	38	Private	96185	HS-grad	NaN
3	38	Self-emp-inc	112847	Prof-school	15.0
4	42	Self-emp-not-inc	82297	7th-8th	NaN

	marital-status	occupation	relationship	race \
0	Married-civ-spouse	NaN	Wife	White
1	Divorced	Exec-managerial	Not-in-family	White
2	Divorced	NaN	Unmarried	Black
3	Married-civ-spouse	Prof-specialty	Husband	Asian-Pac-Islander
4	Married-civ-spouse	Other-service	Wife	Black

	sex	capital-gain	capital-loss	hours-per-week	native-country	salary
0	Female	0	1902	40	United-States	>=50k
1	Male	10520	0	45	United-States	>=50k
2	Female	0	0	32	United-States	<50k
3	Male	0	0	40	United-States	>=50k
4	Female	0	0	50	United-States	<50k

In [0]:
```
train_df.shape
```

Out[0]:
```
(800, 15)
```

In [0]:
```
valid_df.shape
```

Out[0]:
```
(200, 15)
```

In [0]:
```
train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 800 entries, 0 to 799
Data columns (total 15 columns):
age               800 non-null int64
workclass         800 non-null object
fnlwgt            800 non-null int64
education         800 non-null object
education-num     406 non-null float64
marital-status    800 non-null object
occupation        388 non-null object
relationship      800 non-null object
race              800 non-null object
sex               800 non-null object
capital-gain      800 non-null int64
capital-loss      800 non-null int64
hours-per-week    800 non-null int64
native-country    800 non-null object
salary            800 non-null object
dtypes: float64(1), int64(5), object(9)
memory usage: 93.8+ KB
```

In [0]:
```
cont_var, cat_var = cont_cat_split(train_df, dep_var='salary')
```

In [0]:
```
cont_var
```

Out[0]:
```
['age',
 'fnlwgt',
 'education-num',
 'capital-gain',
 'capital-loss',
 'hours-per-week']
```

```
In [0]: cat_var

Out[0]: ['workclass',
         'education',
         'marital-status',
         'occupation',
         'relationship',
         'race',
         'sex',
         'native-country']

In [0]: categorified = Categorify(cat_var, cont_var)

In [0]: categorified(train_df)

In [0]: train_df

Out[0]:      age          workclass   fnlwgt     education  education-num  \
        0     49            Private   101320    Assoc-acdm           12.0
        1     44            Private   236746       Masters           14.0
        2     38            Private    96185       HS-grad            NaN
        3     38       Self-emp-inc   112847   Prof-school           15.0
        4     42   Self-emp-not-inc    82297       7th-8th            NaN
        5     20            Private    63210       HS-grad            9.0
        6     49            Private    44434  Some-college           10.0
        7     37            Private   138940          11th            7.0
        8     46            Private   328216       HS-grad            9.0
        9     36       Self-emp-inc   216711       HS-grad            NaN
        10    23            Private   529223     Bachelors           13.0
        11    18            Private   216284          11th            NaN
        12    30            Private   151989     Assoc-voc            NaN
        13    30            Private    55291     Bachelors            NaN
        14    43            Private    84661     Assoc-voc            NaN
        15    51            Private   284329       HS-grad            9.0
        16    38            Private   170174          10th            NaN
        17    35            Private   261293       Masters           14.0
        18    56          State-gov   274111       Masters           14.0
        19    45            Private   267967     Bachelors            NaN
        20    40            Private   188942  Some-college            NaN
        21    26            Private   746432       HS-grad            9.0
        22    46            Private   117605           9th            NaN
        23    29            Private  1268339       HS-grad            NaN
        24    49            Private   247294       HS-grad            9.0
        25    55       Self-emp-inc   222615       Masters           14.0
        26    47   Self-emp-not-inc   213745  Some-college            NaN
        27    41       Self-emp-inc   151089  Some-college            NaN
        28    27            Private   153078   Prof-school            NaN
        29    42            Private    70055          11th            7.0
        ..   ...                ...      ...           ...            ...
```

770	68	Private	128472	Doctorate	NaN
771	37	Private	162494	Bachelors	13.0
772	48	Self-emp-not-inc	197702	Some-college	10.0
773	40	Local-gov	141649	Assoc-voc	11.0
774	42	Private	208875	Some-college	10.0
775	25	Private	521400	5th-6th	3.0
776	64	State-gov	114650	9th	NaN
777	22	?	165065	Some-college	NaN
778	38	Private	298841	HS-grad	9.0
779	62	Private	211408	Assoc-voc	11.0
780	47	Self-emp-inc	206947	Assoc-acdm	12.0
781	30	Private	186932	Bachelors	NaN
782	48	?	63466	HS-grad	NaN
783	29	Self-emp-not-inc	322238	HS-grad	9.0
784	44	Local-gov	144778	Bachelors	13.0
785	70	?	173736	Bachelors	13.0
786	23	Local-gov	212803	Bachelors	NaN
787	29	Private	22641	HS-grad	NaN
788	49	Private	122385	Masters	14.0
789	23	Local-gov	40021	Some-college	10.0
790	62	Private	210464	HS-grad	9.0
791	31	State-gov	207505	Doctorate	16.0
792	29	Local-gov	190330	Some-college	NaN
793	65	Private	105252	HS-grad	9.0
794	18	?	175648	11th	7.0
795	47	Local-gov	172246	Masters	NaN
796	30	Private	114691	HS-grad	9.0
797	51	Private	41474	10th	6.0
798	26	Private	68895	HS-grad	NaN
799	60	Self-emp-not-inc	166153	Some-college	10.0

	marital-status	occupation	relationship \
0	Married-civ-spouse	NaN	Wife
1	Divorced	Exec-managerial	Not-in-family
2	Divorced	NaN	Unmarried
3	Married-civ-spouse	Prof-specialty	Husband
4	Married-civ-spouse	Other-service	Wife
5	Never-married	Handlers-cleaners	Own-child
6	Divorced	NaN	Other-relative
7	Married-civ-spouse	NaN	Husband
8	Married-civ-spouse	Craft-repair	Husband
9	Married-civ-spouse	NaN	Husband
10	Never-married	NaN	Own-child
11	Never-married	Adm-clerical	Own-child
12	Married-civ-spouse	NaN	Wife
13	Married-civ-spouse	NaN	Husband
14	Married-civ-spouse	Sales	Husband
15	Widowed	NaN	Unmarried

```
16       Married-civ-spouse   Machine-op-inspct         Husband
17            Never-married                 NaN   Not-in-family
18                 Divorced                 NaN   Not-in-family
19       Married-civ-spouse      Prof-specialty         Husband
20       Married-civ-spouse                 NaN            Wife
21            Never-married   Handlers-cleaners       Own-child
22                 Divorced               Sales   Not-in-family
23    Married-spouse-absent                 NaN       Own-child
24       Married-civ-spouse        Craft-repair         Husband
25       Married-civ-spouse     Exec-managerial         Husband
26                 Divorced                 NaN       Unmarried
27       Married-civ-spouse                 NaN         Husband
28            Never-married      Prof-specialty       Own-child
29       Married-civ-spouse                 NaN         Husband
..                      ...                 ...             ...
770      Married-civ-spouse      Prof-specialty         Husband
771           Never-married        Adm-clerical   Not-in-family
772      Married-civ-spouse    Transport-moving         Husband
773      Married-civ-spouse     Protective-serv         Husband
774      Married-civ-spouse                 NaN            Wife
775           Never-married   Machine-op-inspct  Other-relative
776      Married-civ-spouse                 NaN         Husband
777           Never-married                 NaN       Own-child
778                Divorced        Adm-clerical       Own-child
779      Married-civ-spouse        Tech-support         Husband
780                 Widowed                 NaN   Not-in-family
781      Married-civ-spouse     Exec-managerial         Husband
782   Married-spouse-absent                 NaN       Unmarried
783           Never-married     Farming-fishing   Not-in-family
784      Married-civ-spouse      Prof-specialty         Husband
785      Married-civ-spouse                   ?         Husband
786           Never-married                 NaN   Not-in-family
787      Married-civ-spouse   Machine-op-inspct         Husband
788      Married-civ-spouse     Exec-managerial         Husband
789                Divorced        Adm-clerical       Unmarried
790           Never-married       Other-service  Other-relative
791      Married-civ-spouse      Prof-specialty         Husband
792           Never-married                 NaN       Own-child
793      Married-civ-spouse        Tech-support         Husband
794           Never-married                 NaN       Own-child
795      Married-civ-spouse      Prof-specialty         Husband
796           Never-married                 NaN       Own-child
797      Married-civ-spouse                 NaN         Husband
798           Never-married        Adm-clerical   Not-in-family
799      Married-civ-spouse               Sales         Husband

                  race      sex  capital-gain  capital-loss  hours-per-week  \
0                White   Female             0          1902              40
```

1	White	Male	10520	0	45
2	Black	Female	0	0	32
3	Asian-Pac-Islander	Male	0	0	40
4	Black	Female	0	0	50
5	White	Male	0	0	15
6	White	Male	0	0	35
7	White	Male	0	0	40
8	White	Male	0	0	40
9	White	Male	99999	0	50
10	Black	Male	0	0	10
11	White	Female	0	0	20
12	White	Female	0	0	40
13	White	Male	0	0	40
14	White	Male	0	0	45
15	White	Male	0	0	40
16	White	Male	0	0	40
17	White	Male	0	0	60
18	White	Male	0	1669	40
19	White	Male	0	0	45
20	Black	Female	0	0	40
21	Black	Male	0	0	48
22	White	Male	0	0	35
23	Black	Male	0	0	40
24	White	Male	0	0	45
25	White	Male	0	0	60
26	White	Female	0	0	45
27	White	Male	0	0	50
28	Asian-Pac-Islander	Male	0	0	40
29	White	Male	0	0	45
..	...	...	...	...	...
770	White	Male	0	0	55
771	White	Female	0	0	40
772	White	Male	0	0	40
773	White	Male	0	0	40
774	White	Female	0	0	40
775	White	Male	0	0	40
776	White	Male	0	0	40
777	White	Female	0	0	40
778	White	Female	0	0	40
779	White	Male	0	0	40
780	White	Female	0	0	67
781	White	Male	0	0	70
782	White	Female	0	0	32
783	White	Male	0	0	40
784	White	Male	0	0	40
785	White	Male	0	0	6
786	White	Female	0	0	35
787	Amer-Indian-Eskimo	Male	0	0	45

```
788               White     Male            0          0          40
789               White   Female            0          0          70
790               Black   Female            0          0          38
791               White     Male            0       1977          70
792               White   Female            0          0          10
793               White     Male            0          0          40
794               White     Male            0          0          40
795               White     Male            0          0          60
796               White     Male            0          0          40
797               White     Male            0          0          40
798               White     Male            0          0          50
799               White     Male            0          0          50

     native-country salary
0     United-States  >=50k
1     United-States  >=50k
2     United-States   <50k
3     United-States  >=50k
4     United-States   <50k
5     United-States   <50k
6     United-States   <50k
7     United-States   <50k
8     United-States  >=50k
9                 ?  >=50k
10    United-States   <50k
11    United-States   <50k
12    United-States   <50k
13    United-States  >=50k
14    United-States   <50k
15    United-States   <50k
16    United-States  >=50k
17    United-States   <50k
18    United-States   <50k
19    United-States  >=50k
20      Puerto-Rico   <50k
21    United-States   <50k
22    United-States   <50k
23    United-States   <50k
24    United-States  >=50k
25    United-States   <50k
26    United-States   <50k
27    United-States   <50k
28    United-States   <50k
29    United-States   <50k
..              ...    ...
770   United-States  >=50k
771   United-States   <50k
772   United-States   <50k
```

```
773   United-States  >=50k
774     El-Salvador  >=50k
775          Mexico   <50k
776   United-States   <50k
777           Italy   <50k
778   United-States   <50k
779   United-States  >=50k
780   United-States   <50k
781   United-States   <50k
782   United-States   <50k
783   United-States   <50k
784   United-States  >=50k
785   United-States   <50k
786   United-States   <50k
787   United-States   <50k
788   United-States  >=50k
789   United-States   <50k
790   United-States   <50k
791   United-States  >=50k
792   United-States   <50k
793   United-States  >=50k
794   United-States   <50k
795   United-States  >=50k
796   United-States   <50k
797          Mexico   <50k
798          Mexico   <50k
799   United-States   <50k

[800 rows x 15 columns]
```

In [0]:
```
train_df.dtypes
```

Out[0]:
```
age                  int64
workclass         category
fnlwgt               int64
education         category
education-num      float64
marital-status    category
occupation        category
relationship      category
race              category
sex               category
capital-gain         int64
capital-loss         int64
hours-per-week       int64
native-country    category
salary              object
dtype: object
```

```
In [0]:
```