Blue Book for Bulldozers

September 30, 2018

1 Blue Book for Bulldozers

```
In [64]: %load_ext autoreload
         %autoreload 2
         %matplotlib inline
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

```
In [65]: from fastai.imports import *
         from fastai.structured import *

         from pandas_summary import DataFrameSummary
         from sklearn.ensemble import RandomForestRegressor, RandomForestClassifier
         from IPython.display import display

         from sklearn import metrics
```

```
In [66]: # point path to data
         PATH = "data/bulldozers/"
```

```
In [67]: # use bash command to list out data
         !ls {PATH}
```

```
Data Dictionary.xlsx  random_forest_benchmark_test.csv        Train.csv
Machine_Appendix.csv  Test.csv                                Valid.csv
median_benchmark.csv  tmp                                ValidSolution.csv
models                      TrainAndValid.csv
```

```
In [68]: # notice parse_dates, this is an important step when you have a
         # date column thats not already split into its constituent parts
         df_raw = pd.read_csv(f'{PATH}Train.csv', low_memory=False,
                              parse_dates=["saledate"])
```

In any sort of data science work, it's **important to look at your data**, to make sure you understand the format, how it's stored, what type of values it holds, etc. Even if you've read descriptions about your data, the actual data may not be what you expect.

```
In [69]: def display_all(df):
             with pd.option_context("display.max_rows", 1000, "display.max_columns", 1000):
                 display(df)
```

```
In [70]: display_all(df_raw.tail().T)
```

```
                                                                    401120  \
SalesID                                                            6333336
SalePrice                                                            10500
MachineID                                                          1840702
ModelID                                                              21439
datasource                                                             149
auctioneerID                                                             1
YearMade                                                              2005
MachineHoursCurrentMeter                                               NaN
UsageBand                                                              NaN
saledate                                               2011-11-02 00:00:00
fiModelDesc                                                          35NX2
fiBaseModel                                                             35
fiSecondaryDesc                                                         NX
fiModelSeries                                                            2
fiModelDescriptor                                                      NaN
ProductSize                                                           Mini
fiProductClassDesc         Hydraulic Excavator, Track - 3.0 to 4.0 Metric...
state                                                             Maryland
ProductGroup                                                           TEX
ProductGroupDesc                                          Track Excavators
Drive_System                                                           NaN
Enclosure                                                            EROPS
Forks                                                                  NaN
Pad_Type                                                               NaN
Ride_Control                                                           NaN
Stick                                                                  NaN
Transmission                                                           NaN
Turbocharged                                                           NaN
Blade_Extension                                                        NaN
Blade_Width                                                            NaN
Enclosure_Type                                                         NaN
Engine_Horsepower                                                      NaN
Hydraulics                                                       Auxiliary
Pushblock                                                              NaN
Ripper                                                                 NaN
Scarifier                                                              NaN
Tip_Control                                                            NaN
Tire_Size                                                              NaN
Coupler                                               None or Unspecified
Coupler_System                                                         NaN
Grouser_Tracks                                                         NaN
```

```
Hydraulics_Flow                                               NaN
Track_Type                                                  Steel
Undercarriage_Pad_Width                       None or Unspecified
Stick_Length                                  None or Unspecified
Thumb                                         None or Unspecified
Pattern_Changer                               None or Unspecified
Grouser_Type                                               Double
Backhoe_Mounting                                              NaN
Blade_Type                                                    NaN
Travel_Controls                                               NaN
Differential_Type                                             NaN
Steering_Controls                                             NaN

                                                           401121  \
SalesID                                                   6333337
SalePrice                                                   11000
MachineID                                                 1830472
ModelID                                                     21439
datasource                                                    149
auctioneerID                                                    1
YearMade                                                     2005
MachineHoursCurrentMeter                                      NaN
UsageBand                                                     NaN
saledate                                      2011-11-02 00:00:00
fiModelDesc                                                 35NX2
fiBaseModel                                                    35
fiSecondaryDesc                                                NX
fiModelSeries                                                   2
fiModelDescriptor                                             NaN
ProductSize                                                  Mini
fiProductClassDesc        Hydraulic Excavator, Track - 3.0 to 4.0 Metric...
state                                                    Maryland
ProductGroup                                                  TEX
ProductGroupDesc                                 Track Excavators
Drive_System                                                  NaN
Enclosure                                                   EROPS
Forks                                                         NaN
Pad_Type                                                      NaN
Ride_Control                                                  NaN
Stick                                                         NaN
Transmission                                                  NaN
Turbocharged                                                  NaN
Blade_Extension                                               NaN
Blade_Width                                                   NaN
Enclosure_Type                                                NaN
Engine_Horsepower                                             NaN
Hydraulics                                               Standard
Pushblock                                                     NaN
```

```
Ripper                                                        NaN
Scarifier                                                     NaN
Tip_Control                                                   NaN
Tire_Size                                                     NaN
Coupler                                       None or Unspecified
Coupler_System                                                NaN
Grouser_Tracks                                                NaN
Hydraulics_Flow                                               NaN
Track_Type                                                  Steel
Undercarriage_Pad_Width                       None or Unspecified
Stick_Length                                  None or Unspecified
Thumb                                         None or Unspecified
Pattern_Changer                               None or Unspecified
Grouser_Type                                               Double
Backhoe_Mounting                                              NaN
Blade_Type                                                    NaN
Travel_Controls                                               NaN
Differential_Type                                             NaN
Steering_Controls                                             NaN

                                                           401122  \
SalesID                                                   6333338
SalePrice                                                   11500
MachineID                                                 1887659
ModelID                                                     21439
datasource                                                    149
auctioneerID                                                    1
YearMade                                                     2005
MachineHoursCurrentMeter                                      NaN
UsageBand                                                     NaN
saledate                                      2011-11-02 00:00:00
fiModelDesc                                                 35NX2
fiBaseModel                                                    35
fiSecondaryDesc                                                NX
fiModelSeries                                                   2
fiModelDescriptor                                             NaN
ProductSize                                                  Mini
fiProductClassDesc          Hydraulic Excavator, Track - 3.0 to 4.0 Metric...
state                                                    Maryland
ProductGroup                                                  TEX
ProductGroupDesc                                 Track Excavators
Drive_System                                                  NaN
Enclosure                                                   EROPS
Forks                                                         NaN
Pad_Type                                                      NaN
Ride_Control                                                  NaN
Stick                                                         NaN
Transmission                                                  NaN
```

```
Turbocharged                                                              NaN
Blade_Extension                                                           NaN
Blade_Width                                                               NaN
Enclosure_Type                                                            NaN
Engine_Horsepower                                                         NaN
Hydraulics                                                          Auxiliary
Pushblock                                                                 NaN
Ripper                                                                    NaN
Scarifier                                                                 NaN
Tip_Control                                                               NaN
Tire_Size                                                                 NaN
Coupler                                                 None or Unspecified
Coupler_System                                                            NaN
Grouser_Tracks                                                            NaN
Hydraulics_Flow                                                           NaN
Track_Type                                                              Steel
Undercarriage_Pad_Width                                 None or Unspecified
Stick_Length                                            None or Unspecified
Thumb                                                   None or Unspecified
Pattern_Changer                                         None or Unspecified
Grouser_Type                                                           Double
Backhoe_Mounting                                                          NaN
Blade_Type                                                                NaN
Travel_Controls                                                           NaN
Differential_Type                                                         NaN
Steering_Controls                                                         NaN

                                                                       401123  \
SalesID                                                               6333341
SalePrice                                                                9000
MachineID                                                             1903570
ModelID                                                                 21435
datasource                                                                149
auctioneerID                                                                2
YearMade                                                                 2005
MachineHoursCurrentMeter                                                  NaN
UsageBand                                                                 NaN
saledate                                                  2011-10-25 00:00:00
fiModelDesc                                                              30NX
fiBaseModel                                                                30
fiSecondaryDesc                                                            NX
fiModelSeries                                                             NaN
fiModelDescriptor                                                         NaN
ProductSize                                                              Mini
fiProductClassDesc        Hydraulic Excavator, Track - 2.0 to 3.0 Metric...
state                                                                 Florida
ProductGroup                                                              TEX
ProductGroupDesc                                             Track Excavators
```

Drive_System	NaN
Enclosure	EROPS
Forks	NaN
Pad_Type	NaN
Ride_Control	NaN
Stick	NaN
Transmission	NaN
Turbocharged	NaN
Blade_Extension	NaN
Blade_Width	NaN
Enclosure_Type	NaN
Engine_Horsepower	NaN
Hydraulics	Standard
Pushblock	NaN
Ripper	NaN
Scarifier	NaN
Tip_Control	NaN
Tire_Size	NaN
Coupler	None or Unspecified
Coupler_System	NaN
Grouser_Tracks	NaN
Hydraulics_Flow	NaN
Track_Type	Steel
Undercarriage_Pad_Width	None or Unspecified
Stick_Length	None or Unspecified
Thumb	None or Unspecified
Pattern_Changer	None or Unspecified
Grouser_Type	Double
Backhoe_Mounting	NaN
Blade_Type	NaN
Travel_Controls	NaN
Differential_Type	NaN
Steering_Controls	NaN

	401124
SalesID	6333342
SalePrice	7750
MachineID	1926965
ModelID	21435
datasource	149
auctioneerID	2
YearMade	2005
MachineHoursCurrentMeter	NaN
UsageBand	NaN
saledate	2011-10-25 00:00:00
fiModelDesc	30NX
fiBaseModel	30
fiSecondaryDesc	NX

```
fiModelSeries                                                              NaN
fiModelDescriptor                                                          NaN
ProductSize                                                               Mini
fiProductClassDesc        Hydraulic Excavator, Track - 2.0 to 3.0 Metric...
state                                                                  Florida
ProductGroup                                                               TEX
ProductGroupDesc                                              Track Excavators
Drive_System                                                               NaN
Enclosure                                                                EROPS
Forks                                                                      NaN
Pad_Type                                                                   NaN
Ride_Control                                                               NaN
Stick                                                                      NaN
Transmission                                                               NaN
Turbocharged                                                               NaN
Blade_Extension                                                            NaN
Blade_Width                                                                NaN
Enclosure_Type                                                             NaN
Engine_Horsepower                                                          NaN
Hydraulics                                                            Standard
Pushblock                                                                  NaN
Ripper                                                                     NaN
Scarifier                                                                  NaN
Tip_Control                                                                NaN
Tire_Size                                                                  NaN
Coupler                                                  None or Unspecified
Coupler_System                                                             NaN
Grouser_Tracks                                                             NaN
Hydraulics_Flow                                                            NaN
Track_Type                                                               Steel
Undercarriage_Pad_Width                                  None or Unspecified
Stick_Length                                             None or Unspecified
Thumb                                                    None or Unspecified
Pattern_Changer                                          None or Unspecified
Grouser_Type                                                            Double
Backhoe_Mounting                                                           NaN
Blade_Type                                                                 NaN
Travel_Controls                                                            NaN
Differential_Type                                                          NaN
Steering_Controls                                                          NaN
```

In [71]:
```
# the function above is importtant because if you notice
# this option won't show every column
df_raw
```

Out[71]:
```
        SalesID  SalePrice  MachineID  ModelID  datasource  auctioneerID  \
0       1139246      66000     999089     3157         121           3.0
```

1	1139248	57000	117657	77	121	3.0
2	1139249	10000	434808	7009	121	3.0
3	1139251	38500	1026470	332	121	3.0
4	1139253	11000	1057373	17311	121	3.0
5	1139255	26500	1001274	4605	121	3.0
6	1139256	21000	772701	1937	121	3.0
7	1139261	27000	902002	3539	121	3.0
8	1139272	21500	1036251	36003	121	3.0
9	1139275	65000	1016474	3883	121	3.0
10	1139278	24000	1024998	4605	121	3.0
11	1139282	22500	319906	5255	121	3.0
12	1139283	36000	1052214	2232	121	3.0
13	1139284	30500	1068082	3542	121	3.0
14	1139290	28000	1058450	5162	121	3.0
15	1139291	19000	1004810	4604	121	3.0
16	1139292	13500	1026973	9510	121	3.0
17	1139299	9500	1002713	21442	121	3.0
18	1139301	12500	125790	7040	121	3.0
19	1139304	11500	1011914	3177	121	3.0
20	1139311	41000	1014135	8867	121	3.0
21	1139333	34500	999192	3350	121	3.0
22	1139344	26000	1044500	7040	121	3.0
23	1139346	73000	821452	85	121	3.0
24	1139348	33000	294562	3542	121	3.0
25	1139351	12500	833838	7009	121	3.0
26	1139354	15500	565440	7040	121	3.0
27	1139356	53000	1004127	25458	121	3.0
28	1139357	46000	44800	19167	121	3.0
29	1139358	89000	1018076	1333	121	3.0
...	...	...	...	...	...	...
401095	6333259	10500	1872639	21437	149	1.0
401096	6333260	10000	1816341	21437	149	2.0
401097	6333261	8500	1843949	21437	149	1.0
401098	6333262	10500	1791341	21437	149	2.0
401099	6333263	11000	1833174	21437	149	1.0
401100	6333264	10500	1791370	21437	149	2.0
401101	6333270	10000	1799208	21437	149	1.0
401102	6333272	10500	1927142	21437	149	2.0
401103	6333273	12500	1789856	21437	149	2.0
401104	6333275	10500	1924623	21437	149	2.0
401105	6333276	10000	1835350	21437	149	2.0
401106	6333278	10500	1944702	21437	149	2.0
401107	6333279	12500	1866563	21437	149	2.0
401108	6333280	10500	1851633	21437	149	2.0
401109	6333281	10500	1798958	21437	149	2.0
401110	6333282	10500	1878866	21437	149	2.0
401111	6333283	10000	1874235	21437	149	2.0
401112	6333284	10500	1887654	21437	149	2.0

401113	6333285	10500	1817165	21437	149	2.0
401114	6333287	12500	1918242	21437	149	2.0
401115	6333290	10000	1843374	21437	149	2.0
401116	6333302	8500	1825337	21437	149	2.0
401117	6333307	10000	1821747	21437	149	2.0
401118	6333311	9500	1828862	21437	149	2.0
401119	6333335	8500	1798293	21435	149	2.0
401120	6333336	10500	1840702	21439	149	1.0
401121	6333337	11000	1830472	21439	149	1.0
401122	6333338	11500	1887659	21439	149	1.0
401123	6333341	9000	1903570	21435	149	2.0
401124	6333342	7750	1926965	21435	149	2.0

	YearMade	MachineHoursCurrentMeter	UsageBand	saledate	\
0	2004	68.0	Low	2006-11-16	
1	1996	4640.0	Low	2004-03-26	
2	2001	2838.0	High	2004-02-26	
3	2001	3486.0	High	2011-05-19	
4	2007	722.0	Medium	2009-07-23	
5	2004	508.0	Low	2008-12-18	
6	1993	11540.0	High	2004-08-26	
7	2001	4883.0	High	2005-11-17	
8	2008	302.0	Low	2009-08-27	
9	1000	20700.0	Medium	2007-08-09	
10	2004	1414.0	Medium	2008-08-21	
11	1998	2764.0	Low	2006-08-24	
12	1998	0.0	NaN	2005-10-20	
13	2001	1921.0	Medium	2006-01-26	
14	2004	320.0	Low	2006-01-03	
15	1999	2450.0	Medium	2006-11-16	
16	1999	1972.0	Low	2007-06-14	
17	2003	0.0	NaN	2010-01-28	
18	2001	994.0	Low	2006-03-09	
19	1991	8005.0	Medium	2005-11-17	
20	2000	3259.0	Medium	2006-05-18	
21	1000	16328.0	Medium	2006-10-19	
22	2005	109.0	Low	2007-10-25	
23	1996	17033.0	High	2006-10-19	
24	2001	1877.0	Medium	2004-05-20	
25	2003	1028.0	Medium	2006-03-09	
26	2003	356.0	Low	2006-03-09	
27	2000	0.0	NaN	2007-02-22	
28	2004	904.0	Low	2007-08-09	
29	1998	10466.0	Medium	2006-06-01	
...	...	...	...	...	
401095	2003	NaN	NaN	2011-12-14	
401096	2004	NaN	NaN	2011-09-15	
401097	2005	NaN	NaN	2011-10-28	

```
401098       2004                   NaN       NaN 2011-08-16
401099       2004                   NaN       NaN 2011-12-14
401100       2004                   NaN       NaN 2011-08-16
401101       2004                   NaN       NaN 2011-12-14
401102       2005                   NaN       NaN 2011-08-16
401103       2005                   NaN       NaN 2011-09-15
401104       2005                   NaN       NaN 2011-08-16
401105       2005                   NaN       NaN 2011-10-25
401106       2005                   NaN       NaN 2011-08-16
401107       2005                   NaN       NaN 2011-09-15
401108       2005                   NaN       NaN 2011-08-16
401109       2005                   NaN       NaN 2011-08-16
401110       2005                   NaN       NaN 2011-09-15
401111       2005                   NaN       NaN 2011-10-25
401112       2005                   NaN       NaN 2011-10-25
401113       2005                   NaN       NaN 2011-10-25
401114       2005                   NaN       NaN 2011-11-15
401115       2005                   NaN       NaN 2011-10-25
401116       2005                   NaN       NaN 2011-10-25
401117       2005                   NaN       NaN 2011-10-25
401118       2006                   NaN       NaN 2011-10-25
401119       2005                   NaN       NaN 2011-10-25
401120       2005                   NaN       NaN 2011-11-02
401121       2005                   NaN       NaN 2011-11-02
401122       2005                   NaN       NaN 2011-11-02
401123       2005                   NaN       NaN 2011-10-25
401124       2005                   NaN       NaN 2011-10-25

               ...    Undercarriage_Pad_Width        Stick_Length  \
0              ...                        NaN                 NaN
1              ...                        NaN                 NaN
2              ...                        NaN                 NaN
3              ...                        NaN                 NaN
4              ...                        NaN                 NaN
5              ...                        NaN                 NaN
6              ...       None or Unspecified  None or Unspecified
7              ...                        NaN                 NaN
8              ...       None or Unspecified  None or Unspecified
9              ...                        NaN                 NaN
10             ...                        NaN                 NaN
11             ...                        NaN                 NaN
12             ...       None or Unspecified              11' 0"
13             ...                        NaN                 NaN
14             ...                        NaN                 NaN
15             ...                        NaN                 NaN
16             ...       None or Unspecified  None or Unspecified
17             ...                    16 inch  None or Unspecified
18             ...       None or Unspecified  None or Unspecified
```

```
19            ...                   NaN                  NaN
20            ...               32 inch  None or Unspecified
21            ...                   NaN                  NaN
22            ...   None or Unspecified  None or Unspecified
23            ...                   NaN                  NaN
24            ...                   NaN                  NaN
25            ...                   NaN                  NaN
26            ...   None or Unspecified  None or Unspecified
27            ...   None or Unspecified  None or Unspecified
28            ...                   NaN                  NaN
29            ...   None or Unspecified              15' 9"
...           ...                   ...                  ...
401095        ...   None or Unspecified  None or Unspecified
401096        ...   None or Unspecified  None or Unspecified
401097        ...   None or Unspecified  None or Unspecified
401098        ...   None or Unspecified  None or Unspecified
401099        ...   None or Unspecified  None or Unspecified
401100        ...   None or Unspecified  None or Unspecified
401101        ...   None or Unspecified  None or Unspecified
401102        ...   None or Unspecified  None or Unspecified
401103        ...   None or Unspecified  None or Unspecified
401104        ...   None or Unspecified  None or Unspecified
401105        ...   None or Unspecified  None or Unspecified
401106        ...   None or Unspecified  None or Unspecified
401107        ...   None or Unspecified  None or Unspecified
401108        ...   None or Unspecified  None or Unspecified
401109        ...   None or Unspecified  None or Unspecified
401110        ...   None or Unspecified  None or Unspecified
401111        ...   None or Unspecified  None or Unspecified
401112        ...   None or Unspecified  None or Unspecified
401113        ...   None or Unspecified  None or Unspecified
401114        ...   None or Unspecified  None or Unspecified
401115        ...   None or Unspecified  None or Unspecified
401116        ...   None or Unspecified  None or Unspecified
401117        ...   None or Unspecified  None or Unspecified
401118        ...   None or Unspecified  None or Unspecified
401119        ...   None or Unspecified  None or Unspecified
401120        ...   None or Unspecified  None or Unspecified
401121        ...   None or Unspecified  None or Unspecified
401122        ...   None or Unspecified  None or Unspecified
401123        ...   None or Unspecified  None or Unspecified
401124        ...   None or Unspecified  None or Unspecified

                    Thumb      Pattern_Changer Grouser_Type  \
0                     NaN                  NaN          NaN
1                     NaN                  NaN          NaN
2                     NaN                  NaN          NaN
3                     NaN                  NaN          NaN
```

4	NaN	NaN	NaN
5	NaN	NaN	NaN
6	None or Unspecified	None or Unspecified	Double
7	NaN	NaN	NaN
8	None or Unspecified	None or Unspecified	Double
9	NaN	NaN	NaN
10	NaN	NaN	NaN
11	NaN	NaN	NaN
12	None or Unspecified	None or Unspecified	Double
13	NaN	NaN	NaN
14	NaN	NaN	NaN
15	NaN	NaN	NaN
16	None or Unspecified	None or Unspecified	Double
17	None or Unspecified	None or Unspecified	Double
18	None or Unspecified	Yes	Double
19	NaN	NaN	NaN
20	None or Unspecified	None or Unspecified	Double
21	NaN	NaN	NaN
22	None or Unspecified	None or Unspecified	Double
23	NaN	NaN	NaN
24	NaN	NaN	NaN
25	NaN	NaN	NaN
26	None or Unspecified	Yes	Double
27	None or Unspecified	None or Unspecified	Double
28	NaN	NaN	NaN
29	None or Unspecified	Yes	Double
...	...	...	...
401095	None or Unspecified	None or Unspecified	Double
401096	None or Unspecified	None or Unspecified	Double
401097	None or Unspecified	None or Unspecified	Double
401098	None or Unspecified	None or Unspecified	Double
401099	None or Unspecified	None or Unspecified	Double
401100	None or Unspecified	None or Unspecified	Double
401101	None or Unspecified	None or Unspecified	Double
401102	None or Unspecified	None or Unspecified	Double
401103	None or Unspecified	None or Unspecified	Double
401104	None or Unspecified	None or Unspecified	Double
401105	None or Unspecified	None or Unspecified	Double
401106	None or Unspecified	None or Unspecified	Double
401107	None or Unspecified	None or Unspecified	Double
401108	None or Unspecified	None or Unspecified	Double
401109	None or Unspecified	None or Unspecified	Double
401110	None or Unspecified	None or Unspecified	Double
401111	None or Unspecified	None or Unspecified	Double
401112	None or Unspecified	None or Unspecified	Double
401113	None or Unspecified	None or Unspecified	Double
401114	None or Unspecified	None or Unspecified	Double
401115	None or Unspecified	None or Unspecified	Double

```
401116  None or Unspecified  None or Unspecified       Double
401117  None or Unspecified  None or Unspecified       Double
401118  None or Unspecified  None or Unspecified       Double
401119  None or Unspecified  None or Unspecified       Double
401120  None or Unspecified  None or Unspecified       Double
401121  None or Unspecified  None or Unspecified       Double
401122  None or Unspecified  None or Unspecified       Double
401123  None or Unspecified  None or Unspecified       Double
401124  None or Unspecified  None or Unspecified       Double

           Backhoe_Mounting Blade_Type      Travel_Controls Differential_Type  \
0                       NaN        NaN                  NaN          Standard
1                       NaN        NaN                  NaN          Standard
2                       NaN        NaN                  NaN               NaN
3                       NaN        NaN                  NaN               NaN
4                       NaN        NaN                  NaN               NaN
5                       NaN        NaN                  NaN               NaN
6                       NaN        NaN                  NaN               NaN
7                       NaN        NaN                  NaN               NaN
8                       NaN        NaN                  NaN               NaN
9                       NaN        NaN                  NaN          Standard
10                      NaN        NaN                  NaN               NaN
11      None or Unspecified        PAT  None or Unspecified               NaN
12                      NaN        NaN                  NaN               NaN
13                      NaN        NaN                  NaN               NaN
14                      NaN        NaN                  NaN               NaN
15                      NaN        NaN                  NaN               NaN
16                      NaN        NaN                  NaN               NaN
17                      NaN        NaN                  NaN               NaN
18                      NaN        NaN                  NaN               NaN
19                      NaN        NaN                  NaN               NaN
20                      NaN        NaN                  NaN               NaN
21                      NaN        NaN                  NaN               NaN
22                      NaN        NaN                  NaN               NaN
23                      NaN        NaN                  NaN          Standard
24                      NaN        NaN                  NaN               NaN
25                      NaN        NaN                  NaN               NaN
26                      NaN        NaN                  NaN               NaN
27                      NaN        NaN                  NaN               NaN
28                      NaN        NaN                  NaN               NaN
29                      NaN        NaN                  NaN               NaN
...                     ...        ...                  ...               ...
401095                  NaN        NaN                  NaN               NaN
401096                  NaN        NaN                  NaN               NaN
401097                  NaN        NaN                  NaN               NaN
401098                  NaN        NaN                  NaN               NaN
401099                  NaN        NaN                  NaN               NaN
401100                  NaN        NaN                  NaN               NaN
```

401101	NaN	NaN	NaN	NaN
401102	NaN	NaN	NaN	NaN
401103	NaN	NaN	NaN	NaN
401104	NaN	NaN	NaN	NaN
401105	NaN	NaN	NaN	NaN
401106	NaN	NaN	NaN	NaN
401107	NaN	NaN	NaN	NaN
401108	NaN	NaN	NaN	NaN
401109	NaN	NaN	NaN	NaN
401110	NaN	NaN	NaN	NaN
401111	NaN	NaN	NaN	NaN
401112	NaN	NaN	NaN	NaN
401113	NaN	NaN	NaN	NaN
401114	NaN	NaN	NaN	NaN
401115	NaN	NaN	NaN	NaN
401116	NaN	NaN	NaN	NaN
401117	NaN	NaN	NaN	NaN
401118	NaN	NaN	NaN	NaN
401119	NaN	NaN	NaN	NaN
401120	NaN	NaN	NaN	NaN
401121	NaN	NaN	NaN	NaN
401122	NaN	NaN	NaN	NaN
401123	NaN	NaN	NaN	NaN
401124	NaN	NaN	NaN	NaN

	Steering_Controls
0	Conventional
1	Conventional
2	NaN
3	NaN
4	NaN
5	NaN
6	NaN
7	NaN
8	NaN
9	Conventional
10	NaN
11	NaN
12	NaN
13	NaN
14	NaN
15	NaN
16	NaN
17	NaN
18	NaN
19	NaN
20	NaN
21	NaN

```
22                   NaN
23          Conventional
24                   NaN
25                   NaN
26                   NaN
27                   NaN
28                   NaN
29                   NaN
...                  ...
401095               NaN
401096               NaN
401097               NaN
401098               NaN
401099               NaN
401100               NaN
401101               NaN
401102               NaN
401103               NaN
401104               NaN
401105               NaN
401106               NaN
401107               NaN
401108               NaN
401109               NaN
401110               NaN
401111               NaN
401112               NaN
401113               NaN
401114               NaN
401115               NaN
401116               NaN
401117               NaN
401118               NaN
401119               NaN
401120               NaN
401121               NaN
401122               NaN
401123               NaN
401124               NaN

[401125 rows x 53 columns]
```

```
In [72]: display_all(df_raw.describe(include='all').T)
```

```
                          count unique  \
SalesID                  401125    NaN
SalePrice                401125    NaN
MachineID                401125    NaN
```

ModelID	401125	NaN
datasource	401125	NaN
auctioneerID	380989	NaN
YearMade	401125	NaN
MachineHoursCurrentMeter	142765	NaN
UsageBand	69639	3
saledate	401125	3919
fiModelDesc	401125	4999
fiBaseModel	401125	1950
fiSecondaryDesc	263934	175
fiModelSeries	56908	122
fiModelDescriptor	71919	139
ProductSize	190350	6
fiProductClassDesc	401125	74
state	401125	53
ProductGroup	401125	6
ProductGroupDesc	401125	6
Drive_System	104361	4
Enclosure	400800	6
Forks	192077	2
Pad_Type	79134	4
Ride_Control	148606	3
Stick	79134	2
Transmission	183230	8
Turbocharged	79134	2
Blade_Extension	25219	2
Blade_Width	25219	6
Enclosure_Type	25219	3
Engine_Horsepower	25219	2
Hydraulics	320570	12
Pushblock	25219	2
Ripper	104137	4
Scarifier	25230	2
Tip_Control	25219	3
Tire_Size	94718	17
Coupler	213952	3
Coupler_System	43458	2
Grouser_Tracks	43362	2
Hydraulics_Flow	43362	3
Track_Type	99153	2
Undercarriage_Pad_Width	99872	19
Stick_Length	99218	29
Thumb	99288	3
Pattern_Changer	99218	3
Grouser_Type	99153	3
Backhoe_Mounting	78672	2
Blade_Type	79833	10
Travel_Controls	79834	7

```
Differential_Type          69411      4
Steering_Controls          69369      5

                                                                          top  \
SalesID                                                                   NaN
SalePrice                                                                 NaN
MachineID                                                                 NaN
ModelID                                                                   NaN
datasource                                                                NaN
auctioneerID                                                              NaN
YearMade                                                                  NaN
MachineHoursCurrentMeter                                                  NaN
UsageBand                                                              Medium
saledate                                                  2009-02-16 00:00:00
fiModelDesc                                                              310G
fiBaseModel                                                               580
fiSecondaryDesc                                                             C
fiModelSeries                                                              II
fiModelDescriptor                                                           L
ProductSize                                                            Medium
fiProductClassDesc        Backhoe Loader - 14.0 to 15.0 Ft Standard Digg...
state                                                                 Florida
ProductGroup                                                              TEX
ProductGroupDesc                                             Track Excavators
Drive_System                                                  Two Wheel Drive
Enclosure                                                               OROPS
Forks                                                     None or Unspecified
Pad_Type                                                  None or Unspecified
Ride_Control                                                               No
Stick                                                                Standard
Transmission                                                         Standard
Turbocharged                                              None or Unspecified
Blade_Extension                                           None or Unspecified
Blade_Width                                                               14'
Enclosure_Type                                            None or Unspecified
Engine_Horsepower                                                          No
Hydraulics                                                            2 Valve
Pushblock                                                 None or Unspecified
Ripper                                                    None or Unspecified
Scarifier                                                 None or Unspecified
Tip_Control                                               None or Unspecified
Tire_Size                                                 None or Unspecified
Coupler                                                   None or Unspecified
Coupler_System                                            None or Unspecified
Grouser_Tracks                                            None or Unspecified
Hydraulics_Flow                                                      Standard
Track_Type                                                              Steel
Undercarriage_Pad_Width                                   None or Unspecified
```

```
Stick_Length                                                None or Unspecified
Thumb                                                       None or Unspecified
Pattern_Changer                                             None or Unspecified
Grouser_Type                                                             Double
Backhoe_Mounting                                            None or Unspecified
Blade_Type                                                                  PAT
Travel_Controls                                             None or Unspecified
Differential_Type                                                      Standard
Steering_Controls                                                  Conventional

                            freq                first                 last  \
SalesID                      NaN                  NaN                  NaN
SalePrice                    NaN                  NaN                  NaN
MachineID                    NaN                  NaN                  NaN
ModelID                      NaN                  NaN                  NaN
datasource                   NaN                  NaN                  NaN
auctioneerID                 NaN                  NaN                  NaN
YearMade                     NaN                  NaN                  NaN
MachineHoursCurrentMeter     NaN                  NaN                  NaN
UsageBand                  33985                  NaN                  NaN
saledate                    1932  1989-01-17 00:00:00  2011-12-30 00:00:00
fiModelDesc                 5039                  NaN                  NaN
fiBaseModel                19798                  NaN                  NaN
fiSecondaryDesc            43235                  NaN                  NaN
fiModelSeries              13202                  NaN                  NaN
fiModelDescriptor          15875                  NaN                  NaN
ProductSize                62274                  NaN                  NaN
fiProductClassDesc         56166                  NaN                  NaN
state                      63944                  NaN                  NaN
ProductGroup              101167                  NaN                  NaN
ProductGroupDesc          101167                  NaN                  NaN
Drive_System               46139                  NaN                  NaN
Enclosure                 173932                  NaN                  NaN
Forks                     178300                  NaN                  NaN
Pad_Type                   70614                  NaN                  NaN
Ride_Control               77685                  NaN                  NaN
Stick                      48829                  NaN                  NaN
Transmission              140328                  NaN                  NaN
Turbocharged               75211                  NaN                  NaN
Blade_Extension            24692                  NaN                  NaN
Blade_Width                 9615                  NaN                  NaN
Enclosure_Type             21923                  NaN                  NaN
Engine_Horsepower          23937                  NaN                  NaN
Hydraulics                141404                  NaN                  NaN
Pushblock                  19463                  NaN                  NaN
Ripper                     83452                  NaN                  NaN
Scarifier                  12719                  NaN                  NaN
Tip_Control                16207                  NaN                  NaN
```

Tire_Size	46339	NaN	NaN
Coupler	184582	NaN	NaN
Coupler_System	40430	NaN	NaN
Grouser_Tracks	40515	NaN	NaN
Hydraulics_Flow	42784	NaN	NaN
Track_Type	84880	NaN	NaN
Undercarriage_Pad_Width	79651	NaN	NaN
Stick_Length	78820	NaN	NaN
Thumb	83093	NaN	NaN
Pattern_Changer	90255	NaN	NaN
Grouser_Type	84653	NaN	NaN
Backhoe_Mounting	78652	NaN	NaN
Blade_Type	38612	NaN	NaN
Travel_Controls	69923	NaN	NaN
Differential_Type	68073	NaN	NaN
Steering_Controls	68679	NaN	NaN

	mean	std	min	25% \
SalesID	1.91971e+06	909021	1.13925e+06	1.41837e+06
SalePrice	31099.7	23036.9	4750	14500
MachineID	1.2179e+06	440992	0	1.0887e+06
ModelID	6889.7	6221.78	28	3259
datasource	134.666	8.96224	121	132
auctioneerID	6.55604	16.9768	0	1
YearMade	1899.16	291.797	1000	1985
MachineHoursCurrentMeter	3457.96	27590.3	0	0
UsageBand	NaN	NaN	NaN	NaN
saledate	NaN	NaN	NaN	NaN
fiModelDesc	NaN	NaN	NaN	NaN
fiBaseModel	NaN	NaN	NaN	NaN
fiSecondaryDesc	NaN	NaN	NaN	NaN
fiModelSeries	NaN	NaN	NaN	NaN
fiModelDescriptor	NaN	NaN	NaN	NaN
ProductSize	NaN	NaN	NaN	NaN
fiProductClassDesc	NaN	NaN	NaN	NaN
state	NaN	NaN	NaN	NaN
ProductGroup	NaN	NaN	NaN	NaN
ProductGroupDesc	NaN	NaN	NaN	NaN
Drive_System	NaN	NaN	NaN	NaN
Enclosure	NaN	NaN	NaN	NaN
Forks	NaN	NaN	NaN	NaN
Pad_Type	NaN	NaN	NaN	NaN
Ride_Control	NaN	NaN	NaN	NaN
Stick	NaN	NaN	NaN	NaN
Transmission	NaN	NaN	NaN	NaN
Turbocharged	NaN	NaN	NaN	NaN
Blade_Extension	NaN	NaN	NaN	NaN
Blade_Width	NaN	NaN	NaN	NaN

Enclosure_Type	NaN	NaN	NaN	NaN
Engine_Horsepower	NaN	NaN	NaN	NaN
Hydraulics	NaN	NaN	NaN	NaN
Pushblock	NaN	NaN	NaN	NaN
Ripper	NaN	NaN	NaN	NaN
Scarifier	NaN	NaN	NaN	NaN
Tip_Control	NaN	NaN	NaN	NaN
Tire_Size	NaN	NaN	NaN	NaN
Coupler	NaN	NaN	NaN	NaN
Coupler_System	NaN	NaN	NaN	NaN
Grouser_Tracks	NaN	NaN	NaN	NaN
Hydraulics_Flow	NaN	NaN	NaN	NaN
Track_Type	NaN	NaN	NaN	NaN
Undercarriage_Pad_Width	NaN	NaN	NaN	NaN
Stick_Length	NaN	NaN	NaN	NaN
Thumb	NaN	NaN	NaN	NaN
Pattern_Changer	NaN	NaN	NaN	NaN
Grouser_Type	NaN	NaN	NaN	NaN
Backhoe_Mounting	NaN	NaN	NaN	NaN
Blade_Type	NaN	NaN	NaN	NaN
Travel_Controls	NaN	NaN	NaN	NaN
Differential_Type	NaN	NaN	NaN	NaN
Steering_Controls	NaN	NaN	NaN	NaN

	50%	75%	max
SalesID	1.63942e+06	2.24271e+06	6.33334e+06
SalePrice	24000	40000	142000
MachineID	1.27949e+06	1.46807e+06	2.48633e+06
ModelID	4604	8724	37198
datasource	132	136	172
auctioneerID	2	4	99
YearMade	1995	2000	2013
MachineHoursCurrentMeter	0	3025	2.4833e+06
UsageBand	NaN	NaN	NaN
saledate	NaN	NaN	NaN
fiModelDesc	NaN	NaN	NaN
fiBaseModel	NaN	NaN	NaN
fiSecondaryDesc	NaN	NaN	NaN
fiModelSeries	NaN	NaN	NaN
fiModelDescriptor	NaN	NaN	NaN
ProductSize	NaN	NaN	NaN
fiProductClassDesc	NaN	NaN	NaN
state	NaN	NaN	NaN
ProductGroup	NaN	NaN	NaN
ProductGroupDesc	NaN	NaN	NaN
Drive_System	NaN	NaN	NaN
Enclosure	NaN	NaN	NaN
Forks	NaN	NaN	NaN

```
Pad_Type                          NaN          NaN          NaN
Ride_Control                      NaN          NaN          NaN
Stick                             NaN          NaN          NaN
Transmission                      NaN          NaN          NaN
Turbocharged                      NaN          NaN          NaN
Blade_Extension                   NaN          NaN          NaN
Blade_Width                       NaN          NaN          NaN
Enclosure_Type                    NaN          NaN          NaN
Engine_Horsepower                 NaN          NaN          NaN
Hydraulics                        NaN          NaN          NaN
Pushblock                         NaN          NaN          NaN
Ripper                            NaN          NaN          NaN
Scarifier                         NaN          NaN          NaN
Tip_Control                       NaN          NaN          NaN
Tire_Size                         NaN          NaN          NaN
Coupler                           NaN          NaN          NaN
Coupler_System                    NaN          NaN          NaN
Grouser_Tracks                    NaN          NaN          NaN
Hydraulics_Flow                   NaN          NaN          NaN
Track_Type                        NaN          NaN          NaN
Undercarriage_Pad_Width           NaN          NaN          NaN
Stick_Length                      NaN          NaN          NaN
Thumb                             NaN          NaN          NaN
Pattern_Changer                   NaN          NaN          NaN
Grouser_Type                      NaN          NaN          NaN
Backhoe_Mounting                  NaN          NaN          NaN
Blade_Type                        NaN          NaN          NaN
Travel_Controls                   NaN          NaN          NaN
Differential_Type                 NaN          NaN          NaN
Steering_Controls                 NaN          NaN          NaN
```

It's important to note what metric is being used for a project. Generally, selecting the metric(s) is an important part of the project setup. However, in this case Kaggle tells us what metric to use: RMSLE (root mean squared log error) between the actual and predicted auction prices. Therefore we take the log of the prices, so that RMSE will give us what we need.

```
In [73]: df_raw.SalePrice.head()
```

```
Out[73]: 0    66000
         1    57000
         2    10000
         3    38500
         4    11000
         Name: SalePrice, dtype: int64
```

```
In [74]: # since we know they care about RMSLE our dependent variable
         # should be represented as a log
```

```
# numpy can cast every element as a log
df_raw.SalePrice = np.log(df_raw.SalePrice)
```

In [75]:
```
df_raw.SalePrice.head()
```

Out[75]:
```
0    11.097410
1    10.950807
2     9.210340
3    10.558414
4     9.305651
Name: SalePrice, dtype: float64
```

1.0.1 Initial processing

In [76]:
```
# here we create a model that leverages all our cpu/gpu power hence the -1
m = RandomForestRegressor(n_jobs=-1)
# The following code is supposed to fail due to string values in the input data
# here we call fit, fits, general form is the list of independent variables
# and then the independent variables we want to predict
# pandas.drop() will return everything but the dropped column
# then we just pass that column as the dependent variable
# axis = 1 means remove columns
m.fit(df_raw.drop('SalePrice', axis=1), df_raw.SalePrice)
```

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

ValueError                                Traceback (most recent call last)

<ipython-input-76-952edc4fb6f1> in <module>()
      7 # then we just pass that column as the dependent variable
      8 # axis = 1 means remove columns
----> 9 m.fit(df_raw.drop('SalePrice', axis=1), df_raw.SalePrice)


~/anaconda3/envs/fastai/lib/python3.6/site-packages/sklearn/ensemble/forest.py in fit(s
    245         """
    246         # Validate or convert input data
--> 247         X = check_array(X, accept_sparse="csc", dtype=DTYPE)
    248         y = check_array(y, accept_sparse='csc', ensure_2d=False, dtype=None)
    249         if sample_weight is not None:


~/anaconda3/envs/fastai/lib/python3.6/site-packages/sklearn/utils/validation.py in chec
    431                                       force_all_finite)
    432     else:
--> 433         array = np.array(array, dtype=dtype, order=order, copy=copy)
    434 
    435         if ensure_2d:
```

```
ValueError: could not convert string to float: 'Conventional'
```

Machine Learning models need numbers , notice above the fit command didn't work thats because the column conventional is in string format and could not be convreted to a float. ML models convert all numbers to floats, or at least SKLearn and Pythorch do.

This dataset contains a mix of **continuous** and **categorical** variables.

The following method extracts particular date fields from a complete datetime for the purpose of constructing categoricals. You should always consider this feature extraction step when working with date-time. Without expanding your date-time into these additional fields, you can't capture any trend/cyclical behavior as a function of time at any of these granularities.

```
In [77]: # notice how this is of type datetime, we actuallu declared this earlier
         # this is not a number its an object
         # time to do our first piece of feature engineering
         df_raw.saledate.head()

Out[77]: 0   2006-11-16
         1   2004-03-26
         2   2004-02-26
         3   2011-05-19
         4   2009-07-23
         Name: saledate, dtype: datetime64[ns]
```

There is **a lot** going on in a date

```
In [78]: # Here is something very useful from the fastai library
         # for handling dates, notice where its from
         add_datepart

Out[78]: <function fastai.structured.add_datepart(df, fldname, drop=True, time=False)>
```

add_datepart is actually really great, it will go through a declared datetime object and see what kind of date data it contains, then it wil break all of them into seperte columns of number the model can digest, and finally delete the original datetime object from the dataframe.

```
In [79]: df_raw.columns

Out[79]: Index(['SalesID', 'SalePrice', 'MachineID', 'ModelID', 'datasource',
                'auctioneerID', 'YearMade', 'MachineHoursCurrentMeter', 'UsageBand',
                'saledate', 'fiModelDesc', 'fiBaseModel', 'fiSecondaryDesc',
                'fiModelSeries', 'fiModelDescriptor', 'ProductSize',
                'fiProductClassDesc', 'state', 'ProductGroup', 'ProductGroupDesc',
                'Drive_System', 'Enclosure', 'Forks', 'Pad_Type', 'Ride_Control',
                'Stick', 'Transmission', 'Turbocharged', 'Blade_Extension',
                'Blade_Width', 'Enclosure_Type', 'Engine_Horsepower', 'Hydraulics',
                'Pushblock', 'Ripper', 'Scarifier', 'Tip_Control', 'Tire_Size',
```

```
             'Coupler', 'Coupler_System', 'Grouser_Tracks', 'Hydraulics_Flow',
             'Track_Type', 'Undercarriage_Pad_Width', 'Stick_Length', 'Thumb',
             'Pattern_Changer', 'Grouser_Type', 'Backhoe_Mounting', 'Blade_Type',
             'Travel_Controls', 'Differential_Type', 'Steering_Controls'],
            dtype='object')
```

```
In [80]: add_datepart(df_raw, 'saledate')
         df_raw.saleYear.head()
```

```
Out[80]: 0    2006
         1    2004
         2    2004
         3    2011
         4    2009
         Name: saleYear, dtype: int64
```

```
In [81]: # notice sale date is now gone and the parts that were used
         # to make it up now have their own columns
         df_raw.columns
```

```
Out[81]: Index(['SalesID', 'SalePrice', 'MachineID', 'ModelID', 'datasource',
                'auctioneerID', 'YearMade', 'MachineHoursCurrentMeter', 'UsageBand',
                'fiModelDesc', 'fiBaseModel', 'fiSecondaryDesc', 'fiModelSeries',
                'fiModelDescriptor', 'ProductSize', 'fiProductClassDesc', 'state',
                'ProductGroup', 'ProductGroupDesc', 'Drive_System', 'Enclosure',
                'Forks', 'Pad_Type', 'Ride_Control', 'Stick', 'Transmission',
                'Turbocharged', 'Blade_Extension', 'Blade_Width', 'Enclosure_Type',
                'Engine_Horsepower', 'Hydraulics', 'Pushblock', 'Ripper', 'Scarifier',
                'Tip_Control', 'Tire_Size', 'Coupler', 'Coupler_System',
                'Grouser_Tracks', 'Hydraulics_Flow', 'Track_Type',
                'Undercarriage_Pad_Width', 'Stick_Length', 'Thumb', 'Pattern_Changer',
                'Grouser_Type', 'Backhoe_Mounting', 'Blade_Type', 'Travel_Controls',
                'Differential_Type', 'Steering_Controls', 'saleYear', 'saleMonth',
                'saleWeek', 'saleDay', 'saleDayofweek', 'saleDayofyear',
                'saleIs_month_end', 'saleIs_month_start', 'saleIs_quarter_end',
                'saleIs_quarter_start', 'saleIs_year_end', 'saleIs_year_start',
                'saleElapsed'],
               dtype='object')
```

This solves our datetime object problem but not our string problem, things like 'low', 'high', 'medium'

```
In [82]: df_raw.head()
```

```
Out[82]:    SalesID  SalePrice  MachineID  ModelID  datasource  auctioneerID  YearMade  \
         0  1139246  11.097410     999089     3157         121           3.0      2004
         1  1139248  10.950807     117657       77         121           3.0      1996
         2  1139249   9.210340     434808     7009         121           3.0      2001
         3  1139251  10.558414    1026470      332         121           3.0      2001
```

```
4  1139253   9.305651    1057373    17311          121            3.0      2007

   MachineHoursCurrentMeter UsageBand fiModelDesc      ...     saleDay  \
0                      68.0       Low        521D      ...          16
1                    4640.0       Low      950FII      ...          26
2                    2838.0      High         226      ...          26
3                    3486.0      High    PC120-6E      ...          19
4                     722.0    Medium        S175      ...          23

  saleDayofweek saleDayofyear saleIs_month_end saleIs_month_start  \
0             3           320            False              False
1             4            86            False              False
2             3            57            False              False
3             3           139            False              False
4             3           204            False              False

  saleIs_quarter_end saleIs_quarter_start saleIs_year_end saleIs_year_start  \
0              False                False           False             False
1              False                False           False             False
2              False                False           False             False
3              False                False           False             False
4              False                False           False             False

  saleElapsed
0  1163635200
1  1080259200
2  1077753600
3  1305763200
4  1248307200

[5 rows x 65 columns]
```

The categorical variables are currently stored as strings, which is inefficient, and doesn't provide the numeric coding required for a random forest. Therefore we call `train_cats` to convert strings to pandas categories. Pandas has a built in concept for categories, but by default it doesn't turn anything into a category.

```
In [83]: # train_cats is our solution
         # there is some nuance here however
         # when assigning categories it is important to be consistent
         # notice how this is called "train"_cats
         # this is for the training set
         # when you try and train your model you typically have a train and valid set
         # so 'high' might be 0 in train but it could be 2 in valid
         # for future reference this is why fastai has a apply_cats(df, trn)
         # you can pass the data frame of the validation set as well as the
         # applied categories from the train set to achieve aforemention consistency
         train_cats(df_raw)
```

We can specify the order to use for categorical variables if we wish:

```
In [84]: # UsageBand used to be a column of strings where as now
         # it is a category so we can now call .cat
         df_raw.UsageBand.cat.categories

Out[84]: Index(['High', 'Low', 'Medium'], dtype='object')

In [85]: # things may work a little better if you first order your categories
         df_raw.UsageBand.cat.set_categories(['High', 'Medium', 'Low'], ordered=True, inplace=T

In [91]: # notie it is of type categories
         # which pandas is totally cool with
         #df_raw.UsageBand.head()
         os.makedirs('tmp', exist_ok=True)
         df_raw.to_feather('tmp/bulldozers-raw')

In [87]: # here are the category types
         df_raw.UsageBand.cat.categories

Out[87]: Index(['High', 'Medium', 'Low'], dtype='object')

In [88]: # now lets see the codes instead of the strings
         df_raw.UsageBand.cat.codes.head()

Out[88]: 0    2
         1    2
         2    0
         3    0
         4    1
         dtype: int8
```

We're still not quite done - for instance we have lots of missing values, which we can't pass directly to a random forest.

```
In [89]: # isnull will find how many elements are null
         # sum() will add that and give a number back
         # then we can sort them and divide by the size of the dataset
         # this will sjow use the percent of null values in a given column
         display_all(df_raw.isnull().sum().sort_index()/len(df_raw))

Backhoe_Mounting            0.803872
Blade_Extension             0.937129
Blade_Type                  0.800977
Blade_Width                 0.937129
Coupler                     0.466620
Coupler_System              0.891660
Differential_Type           0.826959
Drive_System                0.739829
Enclosure                   0.000810
```

```
Enclosure_Type              0.937129
Engine_Horsepower           0.937129
Forks                       0.521154
Grouser_Tracks              0.891899
Grouser_Type                0.752813
Hydraulics                  0.200823
Hydraulics_Flow             0.891899
MachineHoursCurrentMeter    0.644089
MachineID                   0.000000
ModelID                     0.000000
Pad_Type                    0.802720
Pattern_Changer             0.752651
ProductGroup                0.000000
ProductGroupDesc            0.000000
ProductSize                 0.525460
Pushblock                   0.937129
Ride_Control                0.629527
Ripper                      0.740388
SalePrice                   0.000000
SalesID                     0.000000
Scarifier                   0.937102
Steering_Controls           0.827064
Stick                       0.802720
Stick_Length                0.752651
Thumb                       0.752476
Tip_Control                 0.937129
Tire_Size                   0.763869
Track_Type                  0.752813
Transmission                0.543210
Travel_Controls             0.800975
Turbocharged                0.802720
Undercarriage_Pad_Width     0.751020
UsageBand                   0.826391
YearMade                    0.000000
auctioneerID                0.050199
datasource                  0.000000
fiBaseModel                 0.000000
fiModelDesc                 0.000000
fiModelDescriptor           0.820707
fiModelSeries               0.858129
fiProductClassDesc          0.000000
fiSecondaryDesc             0.342016
saleDay                     0.000000
saleDayofweek               0.000000
saleDayofyear               0.000000
saleElapsed                 0.000000
saleIs_month_end            0.000000
saleIs_month_start          0.000000
```

```
saleIs_quarter_end          0.000000
saleIs_quarter_start        0.000000
saleIs_year_end             0.000000
saleIs_year_start           0.000000
saleMonth                   0.000000
saleWeek                    0.000000
saleYear                    0.000000
state                       0.000000
dtype: float64
```

But let's save this file for now, since it's already in format can we be stored and accessed efficiently.

```
In [90]: # This will save our dataframe for further use and transport
         # also the feather format is really good for saving
         # large amounts of data very fast
         # becoming a standard
         #os.makedirs('tmp', exist_ok=True)
         #df_raw.to_feather('tmp/bulldozers-raw')
```

1.0.2 Pre-processing

In the future we can simply read it from this fast format.

```
In [ ]: #df_raw = pd.read_feather('tmp/bulldozers-raw')
```

We'll replace categories with their numeric codes, handle missing continuous values, and split the dependent variable into a separate variable.

```
In [29]: # now we will be using another tool from the fastai library called proc_df
         # this takes in a dataframe object and the dependent variable
         # copy data frame
         # grab y value
         # drop dependent variable from original
         # then it call fix_missing
         proc_df

Out[29]: <function fastai.structured.proc_df(df, y_fld=None, skip_flds=None, ignore_flds=None,

In [30]: # this is called in the function above
         # it takes in a df, col, and name
         # it checks that a value is actually missing
         # if its numeric it replaces the missing variables with the median
         # value in the column
         # pandas handles categorical variables automatically by setting them to -1

         # if its not numeric and its a categorical type
         # from wht i understand it bumps everyting up by one
         # so -1 will be 0, 0 will become 1, 1 will become 2 ... and so on
         fix_missing
```

Out[30]: <function fastai.structured.fix_missing(df, col, name, na_dict)>

In [31]:
```
df, y, nas = proc_df(df_raw, 'SalePrice')
```

In [32]:
```
# notice now, everything is a number
df.head()
```

Out[32]:
```
   SalesID  MachineID  ModelID  datasource  auctioneerID  YearMade  \
0  1139246     999089     3157         121           3.0      2004
1  1139248     117657       77         121           3.0      1996
2  1139249     434808     7009         121           3.0      2001
3  1139251    1026470      332         121           3.0      2001
4  1139253    1057373    17311         121           3.0      2007

   MachineHoursCurrentMeter  UsageBand  fiModelDesc  fiBaseModel  \
0                      68.0          3          950          296
1                    4640.0          3         1725          527
2                    2838.0          1          331          110
3                    3486.0          1         3674         1375
4                     722.0          2         4208         1529

              ...                saleDayofyear  saleIs_month_end  \
0             ...                          320             False
1             ...                           86             False
2             ...                           57             False
3             ...                          139             False
4             ...                          204             False

   saleIs_month_start  saleIs_quarter_end  saleIs_quarter_start  \
0               False               False                 False
1               False               False                 False
2               False               False                 False
3               False               False                 False
4               False               False                 False

   saleIs_year_end  saleIs_year_start  saleElapsed  auctioneerID_na  \
0            False              False   1163635200            False
1            False              False   1080259200            False
2            False              False   1077753600            False
3            False              False   1305763200            False
4            False              False   1248307200            False

   MachineHoursCurrentMeter_na
0                        False
1                        False
2                        False
3                        False
4                        False
```

```
[5 rows x 66 columns]
```

We now have something we can pass to a random forest!

```
In [33]: m = RandomForestRegressor(n_jobs=-1)
         m.fit(df, y)
         m.score(df,y)

Out[33]: 0.9831480803038706
```

In statistics, the coefficient of determination, denoted R2 or r2 and pronounced "R squared", is the proportion of the variance in the dependent variable that is predictable from the independent variable(s). https://en.wikipedia.org/wiki/Coefficient_of_determination

Wow, an r^2 of 0.98 - that's great, right? Well, perhaps not…

Possibly **the most important idea** in machine learning is that of having separate training & validation data sets. As motivation, suppose you don't divide up your data, but instead use all of it. And suppose you have lots of parameters:

Underfitting and Overfitting

The error for the pictured data points is lowest for the model on the far right (the blue curve passes through the red points almost perfectly), yet it's not the best choice. Why is that? If you were to gather some new data points, they most likely would not be on that curve in the graph on the right, but would be closer to the curve in the middle graph.

This illustrates how using all our data can lead to **overfitting**. A validation set helps diagnose this problem.

```
In [34]: # here is just a quick splitting of the data into a tain and test set
         # I'm pretty sure we don't have a validation
         # set because we don't need one
         def split_vals(a,n):
             return a[:n].copy(), a[n:].copy()

         n_valid = 12000  # same as Kaggle's test set size
         n_trn = len(df)-n_valid
         raw_train, raw_valid = split_vals(df_raw, n_trn)
         X_train, X_valid = split_vals(df, n_trn)
         y_train, y_valid = split_vals(y, n_trn)

         X_train.shape, y_train.shape, X_valid.shape, y_valid.shape

Out[34]: ((389125, 66), (389125,), (12000, 66), (12000,))
```

2 Random Forests

2.1 Base model

Let's try our model again, this time with separate training and validation sets.

```
In [35]: def rmse(x,y): return math.sqrt(((x-y)**2).mean())

         def print_score(m):
             # gives rmse for both train and valid then accuracy (r^2) for each
             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)
```

```
In [36]: m = RandomForestRegressor(n_jobs=-1)
         %time m.fit(X_train, y_train)
         print_score(m)
```

```
CPU times: user 1min 18s, sys: 552 ms, total: 1min 18s
Wall time: 16.3 s
[0.09023701101160281, 0.2533649711716698, 0.9829821681203281, 0.8853586960612733]
```

An r^2 in the high-80's isn't bad at all (and the RMSLE puts us around rank 100 of 470 on the Kaggle leaderboard), but we can see from the validation set score that we're over-fitting badly. To understand this issue, let's simplify things down to a single small tree.

2.2 Speeding things up

```
In [37]: # this will randomly sample 30000 rows
         df_trn, y_trn, nas = proc_df(df_raw, 'SalePrice', subset=30000, na_dict=nas)
         # then we will take new values for the training sets x values
         # and y values, the _ is just a generally accepted practice for throwing
         # a variable away
         X_train, _ = split_vals(df_trn, 20000)
         y_train, _ = split_vals(y_trn, 20000)
```

```
In [38]: X_train.shape, y_train.shape
```

```
Out[38]: ((20000, 66), (20000,))
```

```
In [39]: m = RandomForestRegressor(n_jobs=-1)
         %time m.fit(X_train, y_train)
         print_score(m)
```

```
CPU times: user 3.41 s, sys: 24 ms, total: 3.43 s
Wall time: 810 ms
[0.11155338743152253, 0.37539404082299194, 0.972510943059477, 0.7483350570726175]
```

2.3 Single tree

```
In [40]: # in sklearn trees are referred to as estimators
         # random trees randomize a buch of things
```

```
# bootstrap = False disables that
# this is a poor model btw
m = RandomForestRegressor(n_estimators=1, max_depth=3, bootstrap=False, n_jobs=-1)
m.fit(X_train, y_train)
print_score(m)
```

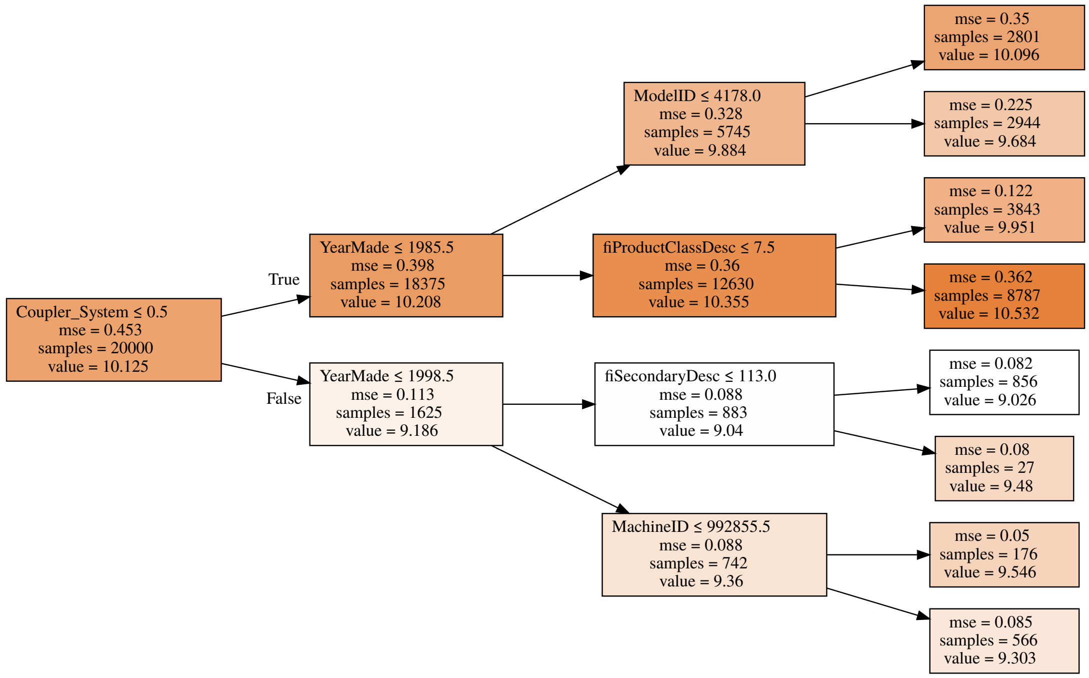
```
[0.5205144895789622, 0.5828702179892538, 0.40150577711834023, 0.3932752591762205]
```

In [41]:
```
# a tree consists of a sequence of binary splits
# notice we start with 20000 rows
# this uses graphiz
# the average of the log is 10.098
# the most basic is a sing leaf which is just predict the average
# we are trying to improve the mse or mean squared error
# so when coupler stems is not less than .5 we improve that
# by alot, however if it is less than the mse doesn't improve much
# we can also see how the samples split  False has a lot less
# how should we make our first binary decsion?
# find a variable that we could split in two such that the two grups
# are as different as possible

# test all possible splits
# for each variable for each possible value of
# that varibale see wether its better

# here is the answer:
# take a weighted average
# take 0.4 * 18317 and add it to .111*1638 for instance
# then compared this weighted average to all other possible splits
# that is how we know where best to make the fist split

# you stop at depth variable or when there are not more splits
draw_tree(m.estimators_[0], df_trn, precision=3)
```




```
In [42]: # better than shallow tree but not good enough
         m = RandomForestRegressor(n_estimators=1, bootstrap=False, n_jobs=-1)
         m.fit(X_train, y_train)
         print_score(m)
```

```
[6.993523870246153e-17, 0.46839068491988145, 1.0, 0.6081999627307028]
```

The training set result looks great! But the validation set is worse than our original model. This is why we need to use *bagging* of multiple trees to get more generalizable results.

2.4 Bagging

2.4.1 Intro to bagging

To learn about bagging in random forests, let's start with our basic model again.

```
In [43]: # by default this will create 10 trees
         # we're looking for a low RMSE for the validation set
         # new hyper parameter: number of trees
         m = RandomForestRegressor(n_jobs=-1)
         m.fit(X_train, y_train)
         print_score(m)
```

```
[0.11096356235686501, 0.3588938445334113, 0.9728008647077702, 0.7699723760625773]
```

```
In [44]: # a random forest is just a way to bag trees
         # averaging different models is the technique for ensembling
         # create big deep massively overfit trees
         # but only use say 1/10 of the data
         # let's say we do it 100 times
         # all ress will be better than nothing
         # but they overfit terribly
         # They all overfit in different ways for different things
         # What is the average of a bunch of random errors? ans: 0
         # so if we take the average the erros will average out to 0
```

We'll grab the predictions for each individual tree, and look at one example.

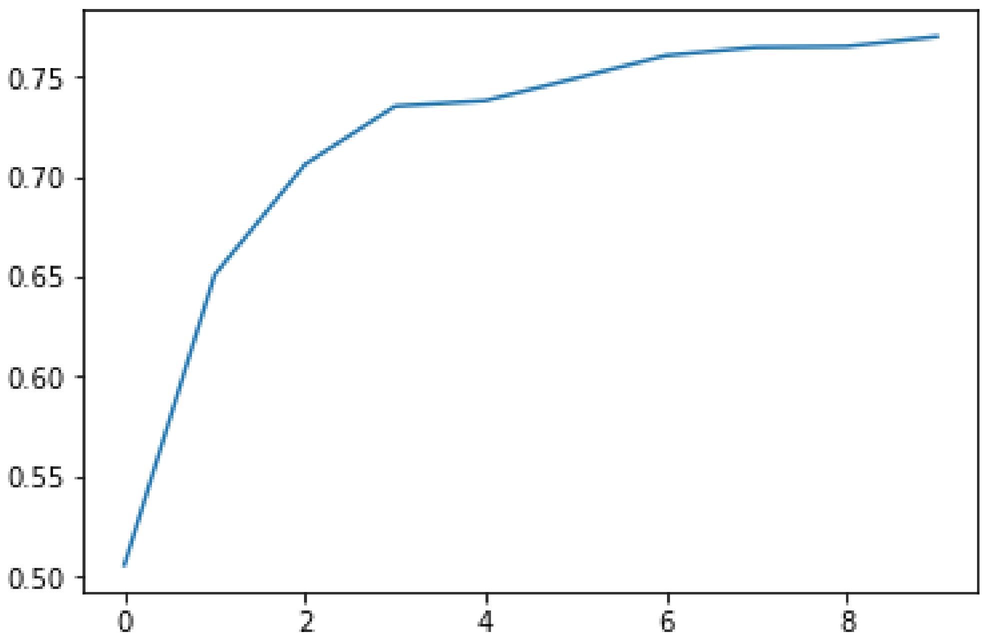
```
In [45]: preds = np.stack([t.predict(X_valid) for t in m.estimators_])
         preds[:,0], np.mean(preds[:,0]), y_valid[0]
```

```
Out[45]: (array([9.18502, 9.15905, 9.15905, 9.13238, 9.15905, 9.71112, 9.04782, 9.18502, 9.1323
          9.185807794479167,
          9.104979856318357)
```

```
In [46]: preds.shape
```

```
Out[46]: (10, 12000)
```

```
In [47]: plt.plot([metrics.r2_score(y_valid, np.mean(preds[:i+1], axis=0)) for i in range(10)]
```



The shape of this curve suggests that adding more trees isn't going to help us much. Let's check. (Compare this to our original model on a sample)

```
In [48]: #m = RandomForestRegressor(n_estimators=20, n_jobs=-1)
         #m.fit(X_train, y_train)
         #print_score(m)
```

```
In [49]: #m = RandomForestRegressor(n_estimators=40, n_jobs=-1)
         #m.fit(X_train, y_train)
         #print_score(m)
```

```
In [50]: #m = RandomForestRegressor(n_estimators=80, n_jobs=-1)
         #m.fit(X_train, y_train)
         #print_score(m)
```

2.4.2 Out-of-bag (OOB) score

Is our validation set worse than our training set because we're over-fitting, or because the validation set is for a different time period, or a bit of both? With the existing information we've shown, we can't tell. However, random forests have a very clever trick called *out-of-bag (OOB) error* which can handle this (and more!)

The idea is to calculate error on the training set, but only include the trees in the calculation of a row's error where that row was *not* included in training that tree. This allows us to see whether the model is over-fitting, without needing a separate validation set.

This also has the benefit of allowing us to see whether our model generalizes, even if we only have a small amount of data so want to avoid separating some out to create a validation set.

This is as simple as adding one more parameter to our model constructor. We print the OOB error last in our `print_score` function below.

```
In [53]: ## OOB score is at the end of this list
         # notice how the r^2 is similar to the validation set (although not as close as jerem
         m = RandomForestRegressor(n_estimators=200, n_jobs=-1, oob_score=True)
         m.fit(X_train, y_train)
         print_score(m)
```

```
[0.0912914003222948, 0.34298413795784344, 0.981589988778005, 0.789914516164208, 0.866072294377
```

This shows that our validation set time difference is making an impact, as is model over-fitting.

2.5 Reducing over-fitting

2.5.1 Subsampling

It turns out that one of the easiest ways to avoid over-fitting is also one of the best ways to speed up analysis: *subsampling*. Let's return to using our full dataset, so that we can demonstrate the impact of this technique.

```
In [54]: df_trn, y_trn, nas = proc_df(df_raw, 'SalePrice')
         X_train, X_valid = split_vals(df_trn, n_trn)
         y_train, y_valid = split_vals(y_trn, n_trn)
```

```
In [55]: len(df_raw)
```

```
Out[55]: 401125

In [56]: len(X_train)

Out[56]: 389125
```

The basic idea is this: rather than limit the total amount of data that our model can access, let's instead limit it to a *different* random subset per tree. That way, given enough trees, the model can still see *all* the data, but for each individual tree it'll be just as fast as if we had cut down our dataset as before.

```
In [57]: # now we can just grab a set of 20000
         set_rf_samples(20000)

In [58]: m = RandomForestRegressor(n_jobs=-1, oob_score=True)
         %time m.fit(X_train, y_train)
         print_score(m)

CPU times: user 21.4 s, sys: 1.73 s, total: 23.1 s
Wall time: 5.5 s
[0.2406191876413026, 0.2779289143235192, 0.878997222690765, 0.8620519912663209, 0.866416202221138
```

Since each additional tree allows the model to see more data, this approach can make additional trees more useful.

```
In [59]: m = RandomForestRegressor(n_estimators=40, n_jobs=-1, oob_score=True)
         m.fit(X_train, y_train)
         print_score(m)

[0.22756925726761643, 0.26527282472021885, 0.8917664233471054, 0.8743294551866888, 0.880152434
```

2.5.2 Tree building parameters

We revert to using a full bootstrap sample in order to show the impact of other over-fitting avoidance methods.

```
In [60]: reset_rf_samples()
```

Let's get a baseline for this full set to compare to.

```
In [61]: m = RandomForestRegressor(n_estimators=40, n_jobs=-1, oob_score=True)
         m.fit(X_train, y_train)
         print_score(m)

[0.07826057371819195, 0.236993210786384, 0.9871996784119512, 0.8996956463677973, 0.90853561181
```

Another way to reduce over-fitting is to grow our trees less deeply. We do this by specifying (with `min_samples_leaf`) that we require some minimum number of rows in every leaf node. This has two benefits: