end-to-end-bulldozer-price-regression

February 22, 2026

1 Predicting the Sale Price of Bulldozers using Machine Learning

In this notebook, we're going to go through an example machine learning project with the goal of predicting the sale price of bulldozers

1.1 1. Problem Definition

> How well can we predict the future sale price of a bulldozer, give it's characteristics and previous examples of how much similar bulldozers have been sold for?

1.2 2. Data

The data is downloaded from the Kaggle Bluebook for bulldozers competition: https://www.kaggle.com/competitions/bluebook-for-bulldozers/data

The data for this competition is split into three parts:

- Train.csv is the training set, which contains data through the end of 2011.
- Valid.csv is the validation set, which contains data from January 1, 2012 - April 30, 2012 You make predictions on this set throughout the majority of the competition. Your score on this set is used to create the public leaderboard.
- Test.csv is the test set, which won't be released until the last week of the competition. It contains data from May 1, 2012 - November 2012. Your score on the test set determines your final rank for the competition.

1.3 3. Evaluation

The evaluation metric for this competition is the RMSLE (root mean squared log error) between the actual and predicted auction prices.

For more on the evaluation of this project check: https://www.kaggle.com/competitions/bluebook-for-bulldozers/overview

Note: The goal for most regression evaluation metrics is to minimize the error. For example, our goal for this project will be to build a machine learning model which minimises RSMLE.

1.4 4.Features

Kaggle provides a data dictionary: https://www.kaggle.com/competitions/bluebook-for-bulldozers/data

[131]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sklearn
```

[132]:
```
# Import the data training & validation

df = pd.read_csv("./data/TrainAndValid.csv", low_memory=False)
```

[133]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 412698 entries, 0 to 412697
Data columns (total 53 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   SalesID                   412698 non-null  int64
 1   SalePrice                 412698 non-null  float64
 2   MachineID                 412698 non-null  int64
 3   ModelID                   412698 non-null  int64
 4   datasource                412698 non-null  int64
 5   auctioneerID              392562 non-null  float64
 6   YearMade                  412698 non-null  int64
 7   MachineHoursCurrentMeter  147504 non-null  float64
 8   UsageBand                 73670 non-null   object
 9   saledate                  412698 non-null  object
 10  fiModelDesc               412698 non-null  object
 11  fiBaseModel               412698 non-null  object
 12  fiSecondaryDesc           271971 non-null  object
 13  fiModelSeries             58667 non-null   object
 14  fiModelDescriptor         74816 non-null   object
 15  ProductSize               196093 non-null  object
 16  fiProductClassDesc        412698 non-null  object
 17  state                     412698 non-null  object
 18  ProductGroup              412698 non-null  object
 19  ProductGroupDesc          412698 non-null  object
 20  Drive_System              107087 non-null  object
 21  Enclosure                 412364 non-null  object
 22  Forks                     197715 non-null  object
 23  Pad_Type                  81096 non-null   object
 24  Ride_Control              152728 non-null  object
 25  Stick                     81096 non-null   object
 26  Transmission              188007 non-null  object
 27  Turbocharged              81096 non-null   object
 28  Blade_Extension           25983 non-null   object
 29  Blade_Width               25983 non-null   object
 30  Enclosure_Type            25983 non-null   object
 31  Engine_Horsepower         25983 non-null   object
```

```
 32  Hydraulics                330133 non-null  object
 33  Pushblock                 25983 non-null   object
 34  Ripper                    106945 non-null  object
 35  Scarifier                 25994 non-null   object
 36  Tip_Control               25983 non-null   object
 37  Tire_Size                 97638 non-null   object
 38  Coupler                   220679 non-null  object
 39  Coupler_System            44974 non-null   object
 40  Grouser_Tracks            44875 non-null   object
 41  Hydraulics_Flow           44875 non-null   object
 42  Track_Type                102193 non-null  object
 43  Undercarriage_Pad_Width   102916 non-null  object
 44  Stick_Length              102261 non-null  object
 45  Thumb                     102332 non-null  object
 46  Pattern_Changer           102261 non-null  object
 47  Grouser_Type              102193 non-null  object
 48  Backhoe_Mounting          80712 non-null   object
 49  Blade_Type                81875 non-null   object
 50  Travel_Controls           81877 non-null   object
 51  Differential_Type         71564 non-null   object
 52  Steering_Controls         71522 non-null   object
dtypes: float64(3), int64(5), object(45)
memory usage: 166.9+ MB
```

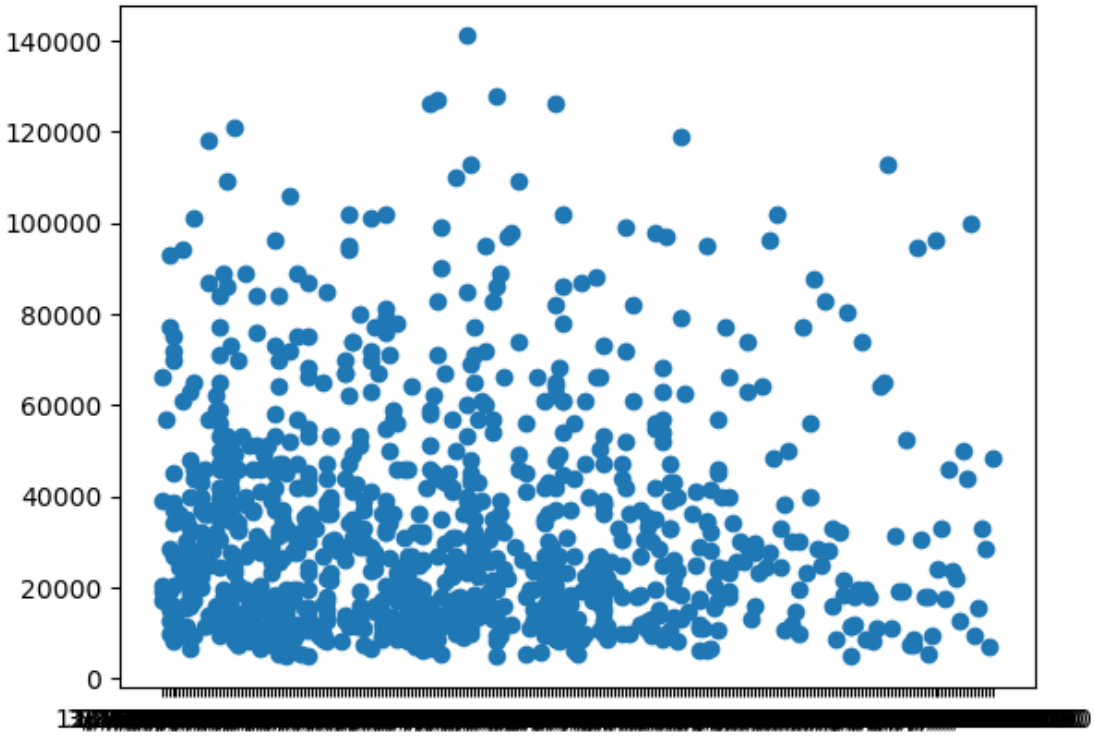
[134]:
```
df.isna().sum()
```

[134]:
```
SalesID                         0
SalePrice                       0
MachineID                       0
ModelID                         0
datasource                      0
auctioneerID                20136
YearMade                        0
MachineHoursCurrentMeter    265194
UsageBand                   339028
saledate                        0
fiModelDesc                     0
fiBaseModel                     0
fiSecondaryDesc            140727
fiModelSeries              354031
fiModelDescriptor          337882
ProductSize                216605
fiProductClassDesc              0
state                           0
ProductGroup                    0
ProductGroupDesc                0
Drive_System               305611
```

```
Enclosure                   334
Forks                    214983
Pad_Type                 331602
Ride_Control             259970
Stick                    331602
Transmission             224691
Turbocharged             331602
Blade_Extension          386715
Blade_Width              386715
Enclosure_Type           386715
Engine_Horsepower        386715
Hydraulics                82565
Pushblock                386715
Ripper                   305753
Scarifier                386704
Tip_Control              386715
Tire_Size                315060
Coupler                  192019
Coupler_System           367724
Grouser_Tracks           367823
Hydraulics_Flow          367823
Track_Type               310505
Undercarriage_Pad_Width  309782
Stick_Length             310437
Thumb                    310366
Pattern_Changer          310437
Grouser_Type             310505
Backhoe_Mounting         331986
Blade_Type               330823
Travel_Controls          330821
Differential_Type        341134
Steering_Controls        341176
dtype: int64
```

[135]:
```
fig, ax = plt.subplots()

ax.scatter(df["saledate"][:1000], df["SalePrice"][:1000])
```

[135]:
```
<matplotlib.collections.PathCollection at 0x235bf3c9950>
```



[136]:
```
df.saledate[:1000]
```

[136]:
```
0       11/16/2006 0:00
1        3/26/2004 0:00
2        2/26/2004 0:00
3        5/19/2011 0:00
4        7/23/2009 0:00
             ...
995      7/16/2009 0:00
996      6/14/2007 0:00
997      9/22/2005 0:00
998      7/28/2005 0:00
999      6/16/2011 0:00
Name: saledate, Length: 1000, dtype: object
```

[137]:
```
#df["SalePrice"].hist()  # taking too long
```

1.5 Parsing Dates

When we work with time series data, we want to enrich the time & date component as much as possible.

We can do that by telling pandas which of our columns has dates in it using the `parse_dates` parameter

[138]:
```
# Import data again but this time parse dates
df = pd.read_csv("./data/TrainAndValid.csv", low_memory=False, parse_dates␣
↪=["saledate"])
```

[139]:
```
df.saledate.dtype
```

[139]:
```
dtype('<M8[ns]')
```

[140]:
```
df.saledate[:1000]
```

[140]:
```
0      2006-11-16
1      2004-03-26
2      2004-02-26
3      2011-05-19
4      2009-07-23
          ...
995    2009-07-16
996    2007-06-14
997    2005-09-22
998    2005-07-28
999    2011-06-16
Name: saledate, Length: 1000, dtype: datetime64[ns]
```

[141]:
```
fig, ax = plt.subplots()

ax.scatter(df["saledate"][:1000], df["SalePrice"][:1000])
```

[141]:
```
<matplotlib.collections.PathCollection at 0x235c53879d0>
```

[142]:
```
df.head()
```

[142]:

	SalesID	SalePrice	MachineID	ModelID	datasource	auctioneerID	YearMade	\
0	1139246	66000.0	999089	3157	121	3.0	2004	
1	1139248	57000.0	117657	77	121	3.0	1996	
2	1139249	10000.0	434808	7009	121	3.0	2001	
3	1139251	38500.0	1026470	332	121	3.0	2001	
4	1139253	11000.0	1057373	17311	121	3.0	2007	

	MachineHoursCurrentMeter	UsageBand	saledate	...	Undercarriage_Pad_Width	\
0	68.0	Low	2006-11-16	...	NaN	
1	4640.0	Low	2004-03-26	...	NaN	
2	2838.0	High	2004-02-26	...	NaN	
3	3486.0	High	2011-05-19	...	NaN	
4	722.0	Medium	2009-07-23	...	NaN	

	Stick_Length	Thumb	Pattern_Changer	Grouser_Type	Backhoe_Mounting	Blade_Type	\
0	NaN	NaN	NaN	NaN	NaN	NaN	
1	NaN	NaN	NaN	NaN	NaN	NaN	
2	NaN	NaN	NaN	NaN	NaN	NaN	
3	NaN	NaN	NaN	NaN	NaN	NaN	
4	NaN	NaN	NaN	NaN	NaN	NaN	

```
  Travel_Controls Differential_Type Steering_Controls
0             NaN          Standard      Conventional
1             NaN          Standard      Conventional
2             NaN               NaN               NaN
3             NaN               NaN               NaN
4             NaN               NaN               NaN

[5 rows x 53 columns]
```

[143]:
```
df.head().T
```

[143]:
```
                                                          0  \
SalesID                                             1139246
SalePrice                                           66000.0
MachineID                                            999089
ModelID                                                3157
datasource                                              121
auctioneerID                                            3.0
YearMade                                               2004
MachineHoursCurrentMeter                               68.0
UsageBand                                               Low
saledate                                2006-11-16 00:00:00
fiModelDesc                                            521D
fiBaseModel                                             521
fiSecondaryDesc                                           D
fiModelSeries                                           NaN
fiModelDescriptor                                       NaN
ProductSize                                             NaN
fiProductClassDesc        Wheel Loader - 110.0 to 120.0 Horsepower
state                                                Alabama
ProductGroup                                              WL
ProductGroupDesc                                Wheel Loader
Drive_System                                            NaN
Enclosure                                         EROPS w AC
Forks                                   None or Unspecified
Pad_Type                                                NaN
Ride_Control                            None or Unspecified
Stick                                                   NaN
Transmission                                            NaN
Turbocharged                                            NaN
Blade_Extension                                         NaN
Blade_Width                                             NaN
Enclosure_Type                                          NaN
Engine_Horsepower                                       NaN
Hydraulics                                        2 Valve
Pushblock                                               NaN
```

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

                                                           1  \
SalesID                                                1139248
SalePrice                                              57000.0
MachineID                                               117657
ModelID                                                     77
datasource                                                 121
auctioneerID                                               3.0
YearMade                                                  1996
MachineHoursCurrentMeter                                4640.0
UsageBand                                                  Low
saledate                                   2004-03-26 00:00:00
fiModelDesc                                              950FII
fiBaseModel                                                950
fiSecondaryDesc                                              F
fiModelSeries                                               II
fiModelDescriptor                                         NaN
ProductSize                                            Medium
fiProductClassDesc        Wheel Loader - 150.0 to 175.0 Horsepower
state                                            North Carolina
ProductGroup                                                WL
ProductGroupDesc                                   Wheel Loader
Drive_System                                              NaN
Enclosure                                          EROPS w AC
Forks                                      None or Unspecified
Pad_Type                                                  NaN
Ride_Control                               None or Unspecified
Stick                                                     NaN
```

Transmission	NaN
Turbocharged	NaN
Blade_Extension	NaN
Blade_Width	NaN
Enclosure_Type	NaN
Engine_Horsepower	NaN
Hydraulics	2 Valve
Pushblock	NaN
Ripper	NaN
Scarifier	NaN
Tip_Control	NaN
Tire_Size	23.5
Coupler	None or Unspecified
Coupler_System	NaN
Grouser_Tracks	NaN
Hydraulics_Flow	NaN
Track_Type	NaN
Undercarriage_Pad_Width	NaN
Stick_Length	NaN
Thumb	NaN
Pattern_Changer	NaN
Grouser_Type	NaN
Backhoe_Mounting	NaN
Blade_Type	NaN
Travel_Controls	NaN
Differential_Type	Standard
Steering_Controls	Conventional

	2 \
SalesID	1139249
SalePrice	10000.0
MachineID	434808
ModelID	7009
datasource	121
auctioneerID	3.0
YearMade	2001
MachineHoursCurrentMeter	2838.0
UsageBand	High
saledate	2004-02-26 00:00:00
fiModelDesc	226
fiBaseModel	226
fiSecondaryDesc	NaN
fiModelSeries	NaN
fiModelDescriptor	NaN
ProductSize	NaN
fiProductClassDesc	Skid Steer Loader - 1351.0 to 1601.0 Lb Operat...
state	New York

ProductGroup	SSL
ProductGroupDesc	Skid Steer Loaders
Drive_System	NaN
Enclosure	OROPS
Forks	None or Unspecified
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	None or Unspecified
Grouser_Tracks	None or Unspecified
Hydraulics_Flow	Standard
Track_Type	NaN
Undercarriage_Pad_Width	NaN
Stick_Length	NaN
Thumb	NaN
Pattern_Changer	NaN
Grouser_Type	NaN
Backhoe_Mounting	NaN
Blade_Type	NaN
Travel_Controls	NaN
Differential_Type	NaN
Steering_Controls	NaN

	3 \
SalesID	1139251
SalePrice	38500.0
MachineID	1026470
ModelID	332
datasource	121
auctioneerID	3.0
YearMade	2001
MachineHoursCurrentMeter	3486.0
UsageBand	High
saledate	2011-05-19 00:00:00

fiModelDesc	PC120-6E
fiBaseModel	PC120
fiSecondaryDesc	NaN
fiModelSeries	-6E
fiModelDescriptor	NaN
ProductSize	Small
fiProductClassDesc	Hydraulic Excavator, Track - 12.0 to 14.0 Metr...
state	Texas
ProductGroup	TEX
ProductGroupDesc	Track Excavators
Drive_System	NaN
Enclosure	EROPS w AC
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	2 Valve
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	NaN
Undercarriage_Pad_Width	NaN
Stick_Length	NaN
Thumb	NaN
Pattern_Changer	NaN
Grouser_Type	NaN
Backhoe_Mounting	NaN
Blade_Type	NaN
Travel_Controls	NaN
Differential_Type	NaN
Steering_Controls	NaN
	4
SalesID	1139253
SalePrice	11000.0

MachineID	1057373
ModelID	17311
datasource	121
auctioneerID	3.0
YearMade	2007
MachineHoursCurrentMeter	722.0
UsageBand	Medium
saledate	2009-07-23 00:00:00
fiModelDesc	S175
fiBaseModel	S175
fiSecondaryDesc	NaN
fiModelSeries	NaN
fiModelDescriptor	NaN
ProductSize	NaN
fiProductClassDesc	Skid Steer Loader - 1601.0 to 1751.0 Lb Operat...
state	New York
ProductGroup	SSL
ProductGroupDesc	Skid Steer Loaders
Drive_System	NaN
Enclosure	EROPS
Forks	None or Unspecified
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	None or Unspecified
Grouser_Tracks	None or Unspecified
Hydraulics_Flow	Standard
Track_Type	NaN
Undercarriage_Pad_Width	NaN
Stick_Length	NaN
Thumb	NaN
Pattern_Changer	NaN
Grouser_Type	NaN
Backhoe_Mounting	NaN

```
Blade_Type                                                    NaN
Travel_Controls                                               NaN
Differential_Type                                             NaN
Steering_Controls                                             NaN
```

[144]:
```
df.saledate.head(20)
```

[144]:
```
0     2006-11-16
1     2004-03-26
2     2004-02-26
3     2011-05-19
4     2009-07-23
5     2008-12-18
6     2004-08-26
7     2005-11-17
8     2009-08-27
9     2007-08-09
10    2008-08-21
11    2006-08-24
12    2005-10-20
13    2006-01-26
14    2006-01-03
15    2006-11-16
16    2007-06-14
17    2010-01-28
18    2006-03-09
19    2005-11-17
Name: saledate, dtype: datetime64[ns]
```

1.5.1 Sort DataFrame by saledate

When working with Time Series Data it is a good idea to sort it by date.

[145]:
```
df.sort_values(by=["saledate"], inplace = True, ascending=True)
```

[146]:
```
df.saledate.head(20)
```

[146]:
```
205615   1989-01-17
274835   1989-01-31
141296   1989-01-31
212552   1989-01-31
62755    1989-01-31
54653    1989-01-31
81383    1989-01-31
204924   1989-01-31
135376   1989-01-31
113390   1989-01-31
113394   1989-01-31
```

```
116419   1989-01-31
32138    1989-01-31
127610   1989-01-31
76171    1989-01-31
127000   1989-01-31
128130   1989-01-31
127626   1989-01-31
55455    1989-01-31
55454    1989-01-31
Name: saledate, dtype: datetime64[ns]
```

[147]:
```
df.head()
```

[147]:

	SalesID	SalePrice	MachineID	ModelID	datasource	auctioneerID	\
205615	1646770	9500.0	1126363	8434	132	18.0	
274835	1821514	14000.0	1194089	10150	132	99.0	
141296	1505138	50000.0	1473654	4139	132	99.0	
212552	1671174	16000.0	1327630	8591	132	99.0	
62755	1329056	22000.0	1336053	4089	132	99.0	

	YearMade	MachineHoursCurrentMeter	UsageBand	saledate	...	\
205615	1974	NaN	NaN	1989-01-17	...	
274835	1980	NaN	NaN	1989-01-31	...	
141296	1978	NaN	NaN	1989-01-31	...	
212552	1980	NaN	NaN	1989-01-31	...	
62755	1984	NaN	NaN	1989-01-31	...	

	Undercarriage_Pad_Width	Stick_Length	Thumb	Pattern_Changer	\
205615	NaN	NaN	NaN	NaN	
274835	NaN	NaN	NaN	NaN	
141296	NaN	NaN	NaN	NaN	
212552	NaN	NaN	NaN	NaN	
62755	NaN	NaN	NaN	NaN	

	Grouser_Type	Backhoe_Mounting	Blade_Type	Travel_Controls	\
205615	NaN	None or Unspecified	Straight	None or Unspecified	
274835	NaN	NaN	NaN	NaN	
141296	NaN	None or Unspecified	Straight	None or Unspecified	
212552	NaN	NaN	NaN	NaN	
62755	NaN	None or Unspecified	PAT	Lever	

	Differential_Type	Steering_Controls
205615	NaN	NaN
274835	Standard	Conventional
141296	NaN	NaN
212552	Standard	Conventional
62755	NaN	NaN

```
[5 rows x 53 columns]
```

1.5.2 Make Copy of the original DataFrame

We make a copy of the original dataframe so when we manipulate the copy, we have still got our original data

[148]:
```
# Make a copy
df_temp = df.copy()
```

[149]:
```
df_temp.saledate.head(20)
```

[149]:
```
205615   1989-01-17
274835   1989-01-31
141296   1989-01-31
212552   1989-01-31
62755    1989-01-31
54653    1989-01-31
81383    1989-01-31
204924   1989-01-31
135376   1989-01-31
113390   1989-01-31
113394   1989-01-31
116419   1989-01-31
32138    1989-01-31
127610   1989-01-31
76171    1989-01-31
127000   1989-01-31
128130   1989-01-31
127626   1989-01-31
55455    1989-01-31
55454    1989-01-31
Name: saledate, dtype: datetime64[ns]
```

1.5.3 Add datetime parameters for `saledate` column

[150]:
```
df_temp["saleYear"] = df_temp.saledate.dt.year
df_temp["saleMonth"] = df_temp.saledate.dt.month
df_temp["saleDay"] = df_temp.saledate.dt.day
df_temp["saleDayOfWeek"] = df_temp.saledate.dt.dayofweek
df_temp["saleDayOfYear"] = df_temp.saledate.dt.dayofyear
```

[151]:
```
df_temp.head().T
```

[151]:
```
                                                205615  \
SalesID                                        1646770
SalePrice                                       9500.0
```

MachineID	1126363
ModelID	8434
datasource	132
auctioneerID	18.0
YearMade	1974
MachineHoursCurrentMeter	NaN
UsageBand	NaN
saledate	1989-01-17 00:00:00
fiModelDesc	TD20
fiBaseModel	TD20
fiSecondaryDesc	NaN
fiModelSeries	NaN
fiModelDescriptor	NaN
ProductSize	Medium
fiProductClassDesc	Track Type Tractor, Dozer - 105.0 to 130.0 Hor...
state	Texas
ProductGroup	TTT
ProductGroupDesc	Track Type Tractors
Drive_System	NaN
Enclosure	OROPS
Forks	NaN
Pad_Type	NaN
Ride_Control	NaN
Stick	NaN
Transmission	Direct Drive
Turbocharged	NaN
Blade_Extension	NaN
Blade_Width	NaN
Enclosure_Type	NaN
Engine_Horsepower	NaN
Hydraulics	2 Valve
Pushblock	NaN
Ripper	None or Unspecified
Scarifier	NaN
Tip_Control	NaN
Tire_Size	NaN
Coupler	NaN
Coupler_System	NaN
Grouser_Tracks	NaN
Hydraulics_Flow	NaN
Track_Type	NaN
Undercarriage_Pad_Width	NaN
Stick_Length	NaN
Thumb	NaN
Pattern_Changer	NaN
Grouser_Type	NaN
Backhoe_Mounting	None or Unspecified

Blade_Type	Straight
Travel_Controls	None or Unspecified
Differential_Type	NaN
Steering_Controls	NaN
saleYear	1989
saleMonth	1
saleDay	17
saleDayOfWeek	1
saleDayOfYear	17

	274835 \
SalesID	1821514
SalePrice	14000.0
MachineID	1194089
ModelID	10150
datasource	132
auctioneerID	99.0
YearMade	1980
MachineHoursCurrentMeter	NaN
UsageBand	NaN
saledate	1989-01-31 00:00:00
fiModelDesc	A66
fiBaseModel	A66
fiSecondaryDesc	NaN
fiModelSeries	NaN
fiModelDescriptor	NaN
ProductSize	NaN
fiProductClassDesc	Wheel Loader - 120.0 to 135.0 Horsepower
state	Florida
ProductGroup	WL
ProductGroupDesc	Wheel Loader
Drive_System	NaN
Enclosure	OROPS
Forks	None or Unspecified
Pad_Type	NaN
Ride_Control	None or Unspecified
Stick	NaN
Transmission	NaN
Turbocharged	NaN
Blade_Extension	NaN
Blade_Width	NaN
Enclosure_Type	NaN
Engine_Horsepower	NaN
Hydraulics	2 Valve
Pushblock	NaN
Ripper	NaN
Scarifier	NaN

Tip_Control	NaN	
Tire_Size	None or Unspecified	
Coupler	None or Unspecified	
Coupler_System	NaN	
Grouser_Tracks	NaN	
Hydraulics_Flow	NaN	
Track_Type	NaN	
Undercarriage_Pad_Width	NaN	
Stick_Length	NaN	
Thumb	NaN	
Pattern_Changer	NaN	
Grouser_Type	NaN	
Backhoe_Mounting	NaN	
Blade_Type	NaN	
Travel_Controls	NaN	
Differential_Type	Standard	
Steering_Controls	Conventional	
saleYear	1989	
saleMonth	1	
saleDay	31	
saleDayOfWeek	1	
saleDayOfYear	31	

	141296	\
SalesID	1505138	
SalePrice	50000.0	
MachineID	1473654	
ModelID	4139	
datasource	132	
auctioneerID	99.0	
YearMade	1978	
MachineHoursCurrentMeter	NaN	
UsageBand	NaN	
saledate	1989-01-31 00:00:00	
fiModelDesc	D7G	
fiBaseModel	D7	
fiSecondaryDesc	G	
fiModelSeries	NaN	
fiModelDescriptor	NaN	
ProductSize	Large	
fiProductClassDesc	Track Type Tractor, Dozer - 190.0 to 260.0 Hor...	
state	Florida	
ProductGroup	TTT	
ProductGroupDesc	Track Type Tractors	
Drive_System	NaN	
Enclosure	OROPS	
Forks	NaN	

```
Pad_Type                                                        NaN
Ride_Control                                                    NaN
Stick                                                           NaN
Transmission                                               Standard
Turbocharged                                                    NaN
Blade_Extension                                                 NaN
Blade_Width                                                     NaN
Enclosure_Type                                                  NaN
Engine_Horsepower                                               NaN
Hydraulics                                                  2 Valve
Pushblock                                                       NaN
Ripper                                         None or Unspecified
Scarifier                                                       NaN
Tip_Control                                                     NaN
Tire_Size                                                       NaN
Coupler                                                         NaN
Coupler_System                                                  NaN
Grouser_Tracks                                                  NaN
Hydraulics_Flow                                                 NaN
Track_Type                                                      NaN
Undercarriage_Pad_Width                                         NaN
Stick_Length                                                    NaN
Thumb                                                           NaN
Pattern_Changer                                                 NaN
Grouser_Type                                                    NaN
Backhoe_Mounting                               None or Unspecified
Blade_Type                                                 Straight
Travel_Controls                                None or Unspecified
Differential_Type                                               NaN
Steering_Controls                                               NaN
saleYear                                                       1989
saleMonth                                                         1
saleDay                                                          31
saleDayOfWeek                                                     1
saleDayOfYear                                                    31

                                          212552  \
SalesID                                  1671174
SalePrice                                16000.0
MachineID                                1327630
ModelID                                     8591
datasource                                   132
auctioneerID                                99.0
YearMade                                    1980
MachineHoursCurrentMeter                     NaN
UsageBand                                    NaN
saledate                     1989-01-31 00:00:00
```

fiModelDesc	A62
fiBaseModel	A62
fiSecondaryDesc	NaN
fiModelSeries	NaN
fiModelDescriptor	NaN
ProductSize	NaN
fiProductClassDesc	Wheel Loader - Unidentified
state	Florida
ProductGroup	WL
ProductGroupDesc	Wheel Loader
Drive_System	NaN
Enclosure	EROPS
Forks	None or Unspecified
Pad_Type	NaN
Ride_Control	None or Unspecified
Stick	NaN
Transmission	NaN
Turbocharged	NaN
Blade_Extension	NaN
Blade_Width	NaN
Enclosure_Type	NaN
Engine_Horsepower	NaN
Hydraulics	2 Valve
Pushblock	NaN
Ripper	NaN
Scarifier	NaN
Tip_Control	NaN
Tire_Size	None or Unspecified
Coupler	None or Unspecified
Coupler_System	NaN
Grouser_Tracks	NaN
Hydraulics_Flow	NaN
Track_Type	NaN
Undercarriage_Pad_Width	NaN
Stick_Length	NaN
Thumb	NaN
Pattern_Changer	NaN
Grouser_Type	NaN
Backhoe_Mounting	NaN
Blade_Type	NaN
Travel_Controls	NaN
Differential_Type	Standard
Steering_Controls	Conventional
saleYear	1989
saleMonth	1
saleDay	31
saleDayOfWeek	1

saleDayOfYear	31
	62755
SalesID	1329056
SalePrice	22000.0
MachineID	1336053
ModelID	4089
datasource	132
auctioneerID	99.0
YearMade	1984
MachineHoursCurrentMeter	NaN
UsageBand	NaN
saledate	1989-01-31 00:00:00
fiModelDesc	D3B
fiBaseModel	D3
fiSecondaryDesc	B
fiModelSeries	NaN
fiModelDescriptor	NaN
ProductSize	NaN
fiProductClassDesc	Track Type Tractor, Dozer - 20.0 to 75.0 Horse...
state	Florida
ProductGroup	TTT
ProductGroupDesc	Track Type Tractors
Drive_System	NaN
Enclosure	OROPS
Forks	NaN
Pad_Type	NaN
Ride_Control	NaN
Stick	NaN
Transmission	Standard
Turbocharged	NaN
Blade_Extension	NaN
Blade_Width	NaN
Enclosure_Type	NaN
Engine_Horsepower	NaN
Hydraulics	2 Valve
Pushblock	NaN
Ripper	None or Unspecified
Scarifier	NaN
Tip_Control	NaN
Tire_Size	NaN
Coupler	NaN
Coupler_System	NaN
Grouser_Tracks	NaN
Hydraulics_Flow	NaN
Track_Type	NaN
Undercarriage_Pad_Width	NaN

```
Stick_Length                                                    NaN
Thumb                                                           NaN
Pattern_Changer                                                 NaN
Grouser_Type                                                    NaN
Backhoe_Mounting                                None or Unspecified
Blade_Type                                                      PAT
Travel_Controls                                               Lever
Differential_Type                                               NaN
Steering_Controls                                               NaN
saleYear                                                       1989
saleMonth                                                         1
saleDay                                                          31
saleDayOfWeek                                                     1
saleDayOfYear                                                    31
```

[152]:
```
# Now we have enriched our DataFrame with date time features we can remove␣
↪saledate
df_temp.drop("saledate",axis=1,inplace=True)
```

[153]:
```
# Check the values of diffrent columns
df_temp.state.value_counts()
```

[153]:
```
state
Florida           67320
Texas             53110
California        29761
Washington        16222
Georgia           14633
Maryland          13322
Mississippi       13240
Ohio              12369
Illinois          11540
Colorado          11529
New Jersey        11156
North Carolina    10636
Tennessee         10298
Alabama           10292
Pennsylvania      10234
South Carolina     9951
Arizona            9364
New York           8639
Connecticut        8276
Minnesota          7885
Missouri           7178
Nevada             6932
Louisiana          6627
Kentucky           5351
```

```
Maine              5096
Indiana            4124
Arkansas           3933
New Mexico         3631
Utah               3046
Unspecified        2801
Wisconsin          2745
New Hampshire      2738
Virginia           2353
Idaho              2025
Oregon             1911
Michigan           1831
Wyoming            1672
Montana            1336
Iowa               1336
Oklahoma           1326
Nebraska            866
West Virginia       840
Kansas              667
Delaware            510
North Dakota        480
Alaska              430
Massachusetts       347
Vermont             300
South Dakota        244
Hawaii              118
Rhode Island         83
Puerto Rico          42
Washington DC         2
Name: count, dtype: int64
```

1.6 5. Modelling

We've done enough EDA (we could always do more) but let's start to do some model driven EDA.

[154]:
```
# Let's build a machine learning model
from sklearn.ensemble import RandomForestRegressor

model = RandomForestRegressor(n_jobs=-1,
                              random_state = 42 #same like random_seed
                              )

#model.fit(df_temp.drop("SalePrice",axis = 1), df_temp.SalePrice) # is not␣
 ↪working because we have some features which datatypes are object
```

1.6.1 Convert string to categories

One way we can turn all of our data into numbers is by converting them into panda categories

We can check the opportunities here: https://pandas.pydata.org/docs/reference/api/pandas.Categorical.dtype.html

[155]:
```
pd.api.types.is_string_dtype(df_temp["UsageBand"])
```

[155]:
```
False
```

[156]:
```
# Find the columns which contain strings
for label, content in df_temp.items():
    if pd.api.types.is_string_dtype(content):
        print(label)
```

```
fiModelDesc
fiBaseModel
fiProductClassDesc
state
ProductGroup
ProductGroupDesc
```

[157]:
```
# if you are wondering what df.items does here is a example

random_dict = {"key1":"Hello",
               "key2":"World", }
```

[158]:
```
for label, content in random_dict.items():
        print(label)
```

```
key1
key2
```

[159]:
```
# This will turn all of the string values into category values

for label, content in df_temp.items():
    if pd.api.types.is_string_dtype(content):
        df_temp[label] = content.astype("category").cat.as_ordered()
```

[160]:
```
# Wir wandeln diese Spalte(n) in den pandas-Datentyp "category" um.
# Dabei speichert pandas nicht jeden String-Wert pro Zeile, sondern:
#   1) eine feste Liste aller möglichen Kategorien (Labels) und
#   2) pro Zeile nur einen integer Code, der auf die Kategorie zeigt.
# Vorteil: weniger Speicherverbrauch und oft schnellere Operationen bei wenigen,␣
 ↪häufig wiederholten Werten.
# Optional (falls gesetzt): Mit ordered=True bekommt die Kategorie eine feste␣
 ↪Reihenfolge, die Sortierung/Vergleiche beeinflusst.
```

[161]:
```
df_temp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 412698 entries, 205615 to 409203
Data columns (total 57 columns):
```

```
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   SalesID                   412698 non-null  int64
 1   SalePrice                 412698 non-null  float64
 2   MachineID                 412698 non-null  int64
 3   ModelID                   412698 non-null  int64
 4   datasource                412698 non-null  int64
 5   auctioneerID              392562 non-null  float64
 6   YearMade                  412698 non-null  int64
 7   MachineHoursCurrentMeter  147504 non-null  float64
 8   UsageBand                 73670 non-null   object
 9   fiModelDesc               412698 non-null  category
 10  fiBaseModel               412698 non-null  category
 11  fiSecondaryDesc           271971 non-null  object
 12  fiModelSeries             58667 non-null   object
 13  fiModelDescriptor         74816 non-null   object
 14  ProductSize               196093 non-null  object
 15  fiProductClassDesc        412698 non-null  category
 16  state                     412698 non-null  category
 17  ProductGroup              412698 non-null  category
 18  ProductGroupDesc          412698 non-null  category
 19  Drive_System              107087 non-null  object
 20  Enclosure                 412364 non-null  object
 21  Forks                     197715 non-null  object
 22  Pad_Type                  81096 non-null   object
 23  Ride_Control              152728 non-null  object
 24  Stick                     81096 non-null   object
 25  Transmission              188007 non-null  object
 26  Turbocharged              81096 non-null   object
 27  Blade_Extension           25983 non-null   object
 28  Blade_Width               25983 non-null   object
 29  Enclosure_Type            25983 non-null   object
 30  Engine_Horsepower         25983 non-null   object
 31  Hydraulics                330133 non-null  object
 32  Pushblock                 25983 non-null   object
 33  Ripper                    106945 non-null  object
 34  Scarifier                 25994 non-null   object
 35  Tip_Control               25983 non-null   object
 36  Tire_Size                 97638 non-null   object
 37  Coupler                   220679 non-null  object
 38  Coupler_System            44974 non-null   object
 39  Grouser_Tracks            44875 non-null   object
 40  Hydraulics_Flow           44875 non-null   object
 41  Track_Type                102193 non-null  object
 42  Undercarriage_Pad_Width   102916 non-null  object
 43  Stick_Length              102261 non-null  object
 44  Thumb                     102332 non-null  object
 45  Pattern_Changer           102261 non-null  object
```

```
 46  Grouser_Type              102193 non-null  object
 47  Backhoe_Mounting          80712 non-null   object
 48  Blade_Type                81875 non-null   object
 49  Travel_Controls           81877 non-null   object
 50  Differential_Type         71564 non-null   object
 51  Steering_Controls         71522 non-null   object
 52  saleYear                  412698 non-null  int32
 53  saleMonth                 412698 non-null  int32
 54  saleDay                   412698 non-null  int32
 55  saleDayOfWeek             412698 non-null  int32
 56  saleDayOfYear             412698 non-null  int32
dtypes: category(6), float64(3), int32(5), int64(5), object(38)
memory usage: 159.3+ MB
```

[162]:
```
df_temp.state.cat.categories
```

[162]:
```
Index(['Alabama', 'Alaska', 'Arizona', 'Arkansas', 'California', 'Colorado',
       'Connecticut', 'Delaware', 'Florida', 'Georgia', 'Hawaii', 'Idaho',
       'Illinois', 'Indiana', 'Iowa', 'Kansas', 'Kentucky', 'Louisiana',
       'Maine', 'Maryland', 'Massachusetts', 'Michigan', 'Minnesota',
       'Mississippi', 'Missouri', 'Montana', 'Nebraska', 'Nevada',
       'New Hampshire', 'New Jersey', 'New Mexico', 'New York',
       'North Carolina', 'North Dakota', 'Ohio', 'Oklahoma', 'Oregon',
       'Pennsylvania', 'Puerto Rico', 'Rhode Island', 'South Carolina',
       'South Dakota', 'Tennessee', 'Texas', 'Unspecified', 'Utah', 'Vermont',
       'Virginia', 'Washington', 'Washington DC', 'West Virginia', 'Wisconsin',
       'Wyoming'],
      dtype='object')
```

[163]:
```
df_temp.state.value_counts()
```

[163]:
```
state
Florida           67320
Texas             53110
California        29761
Washington        16222
Georgia           14633
Maryland          13322
Mississippi       13240
Ohio              12369
Illinois          11540
Colorado          11529
New Jersey        11156
North Carolina    10636
Tennessee         10298
Alabama           10292
Pennsylvania      10234
South Carolina     9951
```

```
Arizona            9364
New York           8639
Connecticut        8276
Minnesota          7885
Missouri           7178
Nevada             6932
Louisiana          6627
Kentucky           5351
Maine              5096
Indiana            4124
Arkansas           3933
New Mexico         3631
Utah               3046
Unspecified        2801
Wisconsin          2745
New Hampshire      2738
Virginia           2353
Idaho              2025
Oregon             1911
Michigan           1831
Wyoming            1672
Montana            1336
Iowa               1336
Oklahoma           1326
Nebraska            866
West Virginia       840
Kansas              667
Delaware            510
North Dakota        480
Alaska              430
Massachusetts       347
Vermont             300
South Dakota        244
Hawaii              118
Rhode Island         83
Puerto Rico          42
Washington DC         2
Name: count, dtype: int64
```

[164]:
```python
# This will turn all of the object values into category values

for label, content in df_temp.items():
    if pd.api.types.is_object_dtype(content):
        df_temp[label] = content.astype("category").cat.as_ordered()
```

[165]:
```python
# Wir wandeln diese Spalte(n) in den pandas-Datentyp "category" um.
# Dabei speichert pandas nicht jeden String-Wert pro Zeile, sondern:
```

```
#   1) eine feste Liste aller möglichen Kategorien (Labels) und
#   2) pro Zeile nur einen integer Code, der auf die Kategorie zeigt.
# Vorteil: weniger Speicherverbrauch und oft schnellere Operationen bei wenigen,␣
 ↪häufig wiederholten Werten.
# Optional (falls gesetzt): Mit ordered=True bekommt die Kategorie eine feste␣
 ↪Reihenfolge, die Sortierung/Vergleiche beeinflusst.
```

[166]:
```
df_temp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 412698 entries, 205615 to 409203
Data columns (total 57 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   SalesID                   412698 non-null  int64
 1   SalePrice                 412698 non-null  float64
 2   MachineID                 412698 non-null  int64
 3   ModelID                   412698 non-null  int64
 4   datasource                412698 non-null  int64
 5   auctioneerID              392562 non-null  float64
 6   YearMade                  412698 non-null  int64
 7   MachineHoursCurrentMeter  147504 non-null  float64
 8   UsageBand                 73670 non-null   category
 9   fiModelDesc               412698 non-null  category
 10  fiBaseModel               412698 non-null  category
 11  fiSecondaryDesc           271971 non-null  category
 12  fiModelSeries             58667 non-null   category
 13  fiModelDescriptor         74816 non-null   category
 14  ProductSize               196093 non-null  category
 15  fiProductClassDesc        412698 non-null  category
 16  state                     412698 non-null  category
 17  ProductGroup              412698 non-null  category
 18  ProductGroupDesc          412698 non-null  category
 19  Drive_System              107087 non-null  category
 20  Enclosure                 412364 non-null  category
 21  Forks                     197715 non-null  category
 22  Pad_Type                  81096 non-null   category
 23  Ride_Control              152728 non-null  category
 24  Stick                     81096 non-null   category
 25  Transmission              188007 non-null  category
 26  Turbocharged              81096 non-null   category
 27  Blade_Extension           25983 non-null   category
 28  Blade_Width               25983 non-null   category
 29  Enclosure_Type            25983 non-null   category
 30  Engine_Horsepower         25983 non-null   category
 31  Hydraulics                330133 non-null  category
 32  Pushblock                 25983 non-null   category
 33  Ripper                    106945 non-null  category
```

```
 34  Scarifier                 25994 non-null   category
 35  Tip_Control               25983 non-null   category
 36  Tire_Size                 97638 non-null   category
 37  Coupler                   220679 non-null  category
 38  Coupler_System            44974 non-null   category
 39  Grouser_Tracks            44875 non-null   category
 40  Hydraulics_Flow           44875 non-null   category
 41  Track_Type                102193 non-null  category
 42  Undercarriage_Pad_Width   102916 non-null  category
 43  Stick_Length              102261 non-null  category
 44  Thumb                     102332 non-null  category
 45  Pattern_Changer           102261 non-null  category
 46  Grouser_Type              102193 non-null  category
 47  Backhoe_Mounting          80712 non-null   category
 48  Blade_Type                81875 non-null   category
 49  Travel_Controls           81877 non-null   category
 50  Differential_Type         71564 non-null   category
 51  Steering_Controls         71522 non-null   category
 52  saleYear                  412698 non-null  int32
 53  saleMonth                 412698 non-null  int32
 54  saleDay                   412698 non-null  int32
 55  saleDayOfWeek             412698 non-null  int32
 56  saleDayOfYear             412698 non-null  int32
dtypes: category(44), float64(3), int32(5), int64(5)
memory usage: 55.4 MB
```

[167]:
```
df_temp.state.cat.codes
```

[167]:
```
205615    43
274835     8
141296     8
212552     8
62755      8
          ..
410879     4
412476     4
411927     4
407124     4
409203     4
Length: 412698, dtype: int8
```

Thanks to pandas Categories we now have a way to access all of our data in the form of numbers.

But we still have a bunch of missing data

[168]:
```
df_temp.isnull().sum()/len(df_temp)
```

[168]:

SalesID	0.000000
SalePrice	0.000000
MachineID	0.000000
ModelID	0.000000
datasource	0.000000
auctioneerID	0.048791
YearMade	0.000000
MachineHoursCurrentMeter	0.642586
UsageBand	0.821492
fiModelDesc	0.000000
fiBaseModel	0.000000
fiSecondaryDesc	0.340993
fiModelSeries	0.857845
fiModelDescriptor	0.818715
ProductSize	0.524851
fiProductClassDesc	0.000000
state	0.000000
ProductGroup	0.000000
ProductGroupDesc	0.000000
Drive_System	0.740520
Enclosure	0.000809
Forks	0.520921
Pad_Type	0.803498
Ride_Control	0.629928
Stick	0.803498
Transmission	0.544444
Turbocharged	0.803498
Blade_Extension	0.937041
Blade_Width	0.937041
Enclosure_Type	0.937041
Engine_Horsepower	0.937041
Hydraulics	0.200062
Pushblock	0.937041
Ripper	0.740864
Scarifier	0.937014
Tip_Control	0.937041
Tire_Size	0.763415
Coupler	0.465277
Coupler_System	0.891024
Grouser_Tracks	0.891264
Hydraulics_Flow	0.891264
Track_Type	0.752378
Undercarriage_Pad_Width	0.750626
Stick_Length	0.752213
Thumb	0.752041
Pattern_Changer	0.752213
Grouser_Type	0.752378

```
Backhoe_Mounting          0.804428
Blade_Type                0.801610
Travel_Controls           0.801606
Differential_Type         0.826595
Steering_Controls         0.826697
saleYear                  0.000000
saleMonth                 0.000000
saleDay                   0.000000
saleDayOfWeek             0.000000
saleDayOfYear             0.000000
dtype: float64
```

1.6.2 Save preprocessed data

[169]:
```
#Export current temp df

df_temp.to_csv("data/train_temp.csv", index = False)
```

[170]:
```
# Import preprocessed data

df_temp=pd.read_csv("data/train_temp.csv", low_memory= False)

df_temp.head().T
```

[170]:
```
                                                                0  \
SalesID                                                   1646770
SalePrice                                                  9500.0
MachineID                                                 1126363
ModelID                                                      8434
datasource                                                    132
auctioneerID                                                 18.0
YearMade                                                     1974
MachineHoursCurrentMeter                                      NaN
UsageBand                                                     NaN
fiModelDesc                                                  TD20
fiBaseModel                                                  TD20
fiSecondaryDesc                                               NaN
fiModelSeries                                                 NaN
fiModelDescriptor                                             NaN
ProductSize                                                Medium
fiProductClassDesc        Track Type Tractor, Dozer - 105.0 to 130.0 Hor...
state                                                      Texas
ProductGroup                                                 TTT
ProductGroupDesc                                Track Type Tractors
Drive_System                                                  NaN
Enclosure                                                   OROPS
Forks                                                         NaN
```

Pad_Type	NaN
Ride_Control	NaN
Stick	NaN
Transmission	Direct Drive
Turbocharged	NaN
Blade_Extension	NaN
Blade_Width	NaN
Enclosure_Type	NaN
Engine_Horsepower	NaN
Hydraulics	2 Valve
Pushblock	NaN
Ripper	None or Unspecified
Scarifier	NaN
Tip_Control	NaN
Tire_Size	NaN
Coupler	NaN
Coupler_System	NaN
Grouser_Tracks	NaN
Hydraulics_Flow	NaN
Track_Type	NaN
Undercarriage_Pad_Width	NaN
Stick_Length	NaN
Thumb	NaN
Pattern_Changer	NaN
Grouser_Type	NaN
Backhoe_Mounting	None or Unspecified
Blade_Type	Straight
Travel_Controls	None or Unspecified
Differential_Type	NaN
Steering_Controls	NaN
saleYear	1989
saleMonth	1
saleDay	17
saleDayOfWeek	1
saleDayOfYear	17

	1 \
SalesID	1821514
SalePrice	14000.0
MachineID	1194089
ModelID	10150
datasource	132
auctioneerID	99.0
YearMade	1980
MachineHoursCurrentMeter	NaN
UsageBand	NaN
fiModelDesc	A66

fiBaseModel	A66
fiSecondaryDesc	NaN
fiModelSeries	NaN
fiModelDescriptor	NaN
ProductSize	NaN
fiProductClassDesc	Wheel Loader - 120.0 to 135.0 Horsepower
state	Florida
ProductGroup	WL
ProductGroupDesc	Wheel Loader
Drive_System	NaN
Enclosure	OROPS
Forks	None or Unspecified
Pad_Type	NaN
Ride_Control	None or Unspecified
Stick	NaN
Transmission	NaN
Turbocharged	NaN
Blade_Extension	NaN
Blade_Width	NaN
Enclosure_Type	NaN
Engine_Horsepower	NaN
Hydraulics	2 Valve
Pushblock	NaN
Ripper	NaN
Scarifier	NaN
Tip_Control	NaN
Tire_Size	None or Unspecified
Coupler	None or Unspecified
Coupler_System	NaN
Grouser_Tracks	NaN
Hydraulics_Flow	NaN
Track_Type	NaN
Undercarriage_Pad_Width	NaN
Stick_Length	NaN
Thumb	NaN
Pattern_Changer	NaN
Grouser_Type	NaN
Backhoe_Mounting	NaN
Blade_Type	NaN
Travel_Controls	NaN
Differential_Type	Standard
Steering_Controls	Conventional
saleYear	1989
saleMonth	1
saleDay	31
saleDayOfWeek	1
saleDayOfYear	31

	2 \
SalesID	1505138
SalePrice	50000.0
MachineID	1473654
ModelID	4139
datasource	132
auctioneerID	99.0
YearMade	1978
MachineHoursCurrentMeter	NaN
UsageBand	NaN
fiModelDesc	D7G
fiBaseModel	D7
fiSecondaryDesc	G
fiModelSeries	NaN
fiModelDescriptor	NaN
ProductSize	Large
fiProductClassDesc	Track Type Tractor, Dozer - 190.0 to 260.0 Hor...
state	Florida
ProductGroup	TTT
ProductGroupDesc	Track Type Tractors
Drive_System	NaN
Enclosure	OROPS
Forks	NaN
Pad_Type	NaN
Ride_Control	NaN
Stick	NaN
Transmission	Standard
Turbocharged	NaN
Blade_Extension	NaN
Blade_Width	NaN
Enclosure_Type	NaN
Engine_Horsepower	NaN
Hydraulics	2 Valve
Pushblock	NaN
Ripper	None or Unspecified
Scarifier	NaN
Tip_Control	NaN
Tire_Size	NaN
Coupler	NaN
Coupler_System	NaN
Grouser_Tracks	NaN
Hydraulics_Flow	NaN
Track_Type	NaN
Undercarriage_Pad_Width	NaN
Stick_Length	NaN
Thumb	NaN

Pattern_Changer	NaN
Grouser_Type	NaN
Backhoe_Mounting	None or Unspecified
Blade_Type	Straight
Travel_Controls	None or Unspecified
Differential_Type	NaN
Steering_Controls	NaN
saleYear	1989
saleMonth	1
saleDay	31
saleDayOfWeek	1
saleDayOfYear	31

	3 \
SalesID	1671174
SalePrice	16000.0
MachineID	1327630
ModelID	8591
datasource	132
auctioneerID	99.0
YearMade	1980
MachineHoursCurrentMeter	NaN
UsageBand	NaN
fiModelDesc	A62
fiBaseModel	A62
fiSecondaryDesc	NaN
fiModelSeries	NaN
fiModelDescriptor	NaN
ProductSize	NaN
fiProductClassDesc	Wheel Loader - Unidentified
state	Florida
ProductGroup	WL
ProductGroupDesc	Wheel Loader
Drive_System	NaN
Enclosure	EROPS
Forks	None or Unspecified
Pad_Type	NaN
Ride_Control	None or Unspecified
Stick	NaN
Transmission	NaN
Turbocharged	NaN
Blade_Extension	NaN
Blade_Width	NaN
Enclosure_Type	NaN
Engine_Horsepower	NaN
Hydraulics	2 Valve
Pushblock	NaN

Ripper	NaN
Scarifier	NaN
Tip_Control	NaN
Tire_Size	None or Unspecified
Coupler	None or Unspecified
Coupler_System	NaN
Grouser_Tracks	NaN
Hydraulics_Flow	NaN
Track_Type	NaN
Undercarriage_Pad_Width	NaN
Stick_Length	NaN
Thumb	NaN
Pattern_Changer	NaN
Grouser_Type	NaN
Backhoe_Mounting	NaN
Blade_Type	NaN
Travel_Controls	NaN
Differential_Type	Standard
Steering_Controls	Conventional
saleYear	1989
saleMonth	1
saleDay	31
saleDayOfWeek	1
saleDayOfYear	31

	4
SalesID	1329056
SalePrice	22000.0
MachineID	1336053
ModelID	4089
datasource	132
auctioneerID	99.0
YearMade	1984
MachineHoursCurrentMeter	NaN
UsageBand	NaN
fiModelDesc	D3B
fiBaseModel	D3
fiSecondaryDesc	B
fiModelSeries	NaN
fiModelDescriptor	NaN
ProductSize	NaN
fiProductClassDesc	Track Type Tractor, Dozer - 20.0 to 75.0 Horse...
state	Florida
ProductGroup	TTT
ProductGroupDesc	Track Type Tractors
Drive_System	NaN
Enclosure	OROPS

```
Forks                                                    NaN
Pad_Type                                                 NaN
Ride_Control                                             NaN
Stick                                                    NaN
Transmission                                        Standard
Turbocharged                                             NaN
Blade_Extension                                          NaN
Blade_Width                                              NaN
Enclosure_Type                                           NaN
Engine_Horsepower                                        NaN
Hydraulics                                         2 Valve
Pushblock                                                NaN
Ripper                                  None or Unspecified
Scarifier                                                NaN
Tip_Control                                              NaN
Tire_Size                                                NaN
Coupler                                                  NaN
Coupler_System                                           NaN
Grouser_Tracks                                           NaN
Hydraulics_Flow                                          NaN
Track_Type                                               NaN
Undercarriage_Pad_Width                                  NaN
Stick_Length                                             NaN
Thumb                                                    NaN
Pattern_Changer                                          NaN
Grouser_Type                                             NaN
Backhoe_Mounting                        None or Unspecified
Blade_Type                                             PAT
Travel_Controls                                      Lever
Differential_Type                                        NaN
Steering_Controls                                        NaN
saleYear                                            1989
saleMonth                                              1
saleDay                                               31
saleDayOfWeek                                          1
saleDayOfYear                                         31
```

[171]:
```
df_temp.isnull().sum()/len(df_temp)
```

[171]:
```
SalesID                   0.000000
SalePrice                 0.000000
MachineID                 0.000000
ModelID                   0.000000
datasource                0.000000
auctioneerID              0.048791
YearMade                  0.000000
MachineHoursCurrentMeter  0.642586
```

UsageBand	0.821492
fiModelDesc	0.000000
fiBaseModel	0.000000
fiSecondaryDesc	0.340993
fiModelSeries	0.857845
fiModelDescriptor	0.818715
ProductSize	0.524851
fiProductClassDesc	0.000000
state	0.000000
ProductGroup	0.000000
ProductGroupDesc	0.000000
Drive_System	0.740520
Enclosure	0.000809
Forks	0.520921
Pad_Type	0.803498
Ride_Control	0.629928
Stick	0.803498
Transmission	0.544444
Turbocharged	0.803498
Blade_Extension	0.937041
Blade_Width	0.937041
Enclosure_Type	0.937041
Engine_Horsepower	0.937041
Hydraulics	0.200062
Pushblock	0.937041
Ripper	0.740864
Scarifier	0.937014
Tip_Control	0.937041
Tire_Size	0.763415
Coupler	0.465277
Coupler_System	0.891024
Grouser_Tracks	0.891264
Hydraulics_Flow	0.891264
Track_Type	0.752378
Undercarriage_Pad_Width	0.750626
Stick_Length	0.752213
Thumb	0.752041
Pattern_Changer	0.752213
Grouser_Type	0.752378
Backhoe_Mounting	0.804428
Blade_Type	0.801610
Travel_Controls	0.801606
Differential_Type	0.826595
Steering_Controls	0.826697
saleYear	0.000000
saleMonth	0.000000
saleDay	0.000000

```
saleDayOfWeek               0.000000
saleDayOfYear               0.000000
dtype: float64
```

[172]:
```
df_temp.isna().sum()
```

[172]:
```
SalesID                         0
SalePrice                       0
MachineID                       0
ModelID                         0
datasource                      0
auctioneerID                20136
YearMade                        0
MachineHoursCurrentMeter    265194
UsageBand                   339028
fiModelDesc                     0
fiBaseModel                     0
fiSecondaryDesc             140727
fiModelSeries               354031
fiModelDescriptor           337882
ProductSize                 216605
fiProductClassDesc              0
state                           0
ProductGroup                    0
ProductGroupDesc                0
Drive_System                305611
Enclosure                      334
Forks                       214983
Pad_Type                    331602
Ride_Control                259970
Stick                       331602
Transmission                224691
Turbocharged                331602
Blade_Extension             386715
Blade_Width                 386715
Enclosure_Type              386715
Engine_Horsepower           386715
Hydraulics                   82565
Pushblock                   386715
Ripper                      305753
Scarifier                   386704
Tip_Control                 386715
Tire_Size                   315060
Coupler                     192019
Coupler_System              367724
Grouser_Tracks              367823
Hydraulics_Flow             367823
```

```
Track_Type                  310505
Undercarriage_Pad_Width     309782
Stick_Length                310437
Thumb                       310366
Pattern_Changer             310437
Grouser_Type                310505
Backhoe_Mounting            331986
Blade_Type                  330823
Travel_Controls             330821
Differential_Type           341134
Steering_Controls           341176
saleYear                         0
saleMonth                        0
saleDay                          0
saleDayOfWeek                    0
saleDayOfYear                    0
dtype: int64
```

1.7 Fill missing values

1.7.1 Fill numerical missing values first

[173]:
```
df_temp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 412698 entries, 0 to 412697
Data columns (total 57 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   SalesID                   412698 non-null  int64
 1   SalePrice                 412698 non-null  float64
 2   MachineID                 412698 non-null  int64
 3   ModelID                   412698 non-null  int64
 4   datasource                412698 non-null  int64
 5   auctioneerID              392562 non-null  float64
 6   YearMade                  412698 non-null  int64
 7   MachineHoursCurrentMeter  147504 non-null  float64
 8   UsageBand                 73670 non-null   object
 9   fiModelDesc               412698 non-null  object
 10  fiBaseModel               412698 non-null  object
 11  fiSecondaryDesc           271971 non-null  object
 12  fiModelSeries             58667 non-null   object
 13  fiModelDescriptor         74816 non-null   object
 14  ProductSize               196093 non-null  object
 15  fiProductClassDesc        412698 non-null  object
 16  state                     412698 non-null  object
 17  ProductGroup              412698 non-null  object
 18  ProductGroupDesc          412698 non-null  object
```

```
 19  Drive_System              107087 non-null  object
 20  Enclosure                 412364 non-null  object
 21  Forks                     197715 non-null  object
 22  Pad_Type                  81096 non-null   object
 23  Ride_Control              152728 non-null  object
 24  Stick                     81096 non-null   object
 25  Transmission              188007 non-null  object
 26  Turbocharged              81096 non-null   object
 27  Blade_Extension           25983 non-null   object
 28  Blade_Width               25983 non-null   object
 29  Enclosure_Type            25983 non-null   object
 30  Engine_Horsepower         25983 non-null   object
 31  Hydraulics                330133 non-null  object
 32  Pushblock                 25983 non-null   object
 33  Ripper                    106945 non-null  object
 34  Scarifier                 25994 non-null   object
 35  Tip_Control               25983 non-null   object
 36  Tire_Size                 97638 non-null   object
 37  Coupler                   220679 non-null  object
 38  Coupler_System            44974 non-null   object
 39  Grouser_Tracks            44875 non-null   object
 40  Hydraulics_Flow           44875 non-null   object
 41  Track_Type                102193 non-null  object
 42  Undercarriage_Pad_Width   102916 non-null  object
 43  Stick_Length              102261 non-null  object
 44  Thumb                     102332 non-null  object
 45  Pattern_Changer           102261 non-null  object
 46  Grouser_Type              102193 non-null  object
 47  Backhoe_Mounting          80712 non-null   object
 48  Blade_Type                81875 non-null   object
 49  Travel_Controls           81877 non-null   object
 50  Differential_Type         71564 non-null   object
 51  Steering_Controls         71522 non-null   object
 52  saleYear                  412698 non-null  int64
 53  saleMonth                 412698 non-null  int64
 54  saleDay                   412698 non-null  int64
 55  saleDayOfWeek             412698 non-null  int64
 56  saleDayOfYear             412698 non-null  int64
dtypes: float64(3), int64(10), object(44)
memory usage: 179.5+ MB
```

[174]:
```
for label,content in df_temp.items():
    if pd.api.types.is_object_dtype(content): # if read_csv we need to change␣
 ↪the dtype again to categorical
        df_temp[label] = content.astype("category").cat.as_ordered()
    if pd.api.types.is_string_dtype(content):
        df_temp[label] = content.astype("category").cat.as_ordered()
```

```
    if pd.api.types.is_numeric_dtype(content):
        print(label)

# Hinweis: CSV-Dateien speichern keine Datentyp-Informationen (nur Werte als␣
 ↪Text).
# Beim pd.read_csv() werden die Spaltentypen deshalb von pandas neu "erraten"␣
 ↪(Type Inference),
# wodurch z.B. category, datetime oder Strings mit führenden Nullen als andere␣
 ↪dtypes eingelesen werden können.
# Lösung: dtypes/parse_dates beim Import explizit setzen oder ein Format wie␣
 ↪Parquet/Feather nutzen, das dtypes mit speichert.
```

```
SalesID
SalePrice
MachineID
ModelID
datasource
auctioneerID
YearMade
MachineHoursCurrentMeter
saleYear
saleMonth
saleDay
saleDayOfWeek
saleDayOfYear
```

[175]:
```
df_temp.ModelID
```

[175]:
```
0          8434
1         10150
2          4139
3          8591
4          4089
          ...
412693     5266
412694    19330
412695    17244
412696     3357
412697     4701
Name: ModelID, Length: 412698, dtype: int64
```

[176]:
```
# Check for which numeric columns have null values

for label,content in df_temp.items():
    if pd.api.types.is_numeric_dtype(content):
        if pd.isnull(content).sum():
            print(label)
```

```
auctioneerID
MachineHoursCurrentMeter
```

[177]:
```
# Fill numeric rows with the median

for label,content in df_temp.items():
    if pd.api.types.is_numeric_dtype(content):
        if pd.isnull(content).sum():
            #Add a binary column which tells us if the data was missing
            df_temp[label+"_is_missing"] = pd.isnull(content)
            #Fill missing numeric values with median
            df_temp[label]= content.fillna(content.median())

# Mean (arithm. Mittelwert) = Summe aller Werte / Anzahl: nutzt jede Beobachtung␣
 ↪direkt und wird durch Ausreißer stark beeinflusst.
# Median = der mittlere Wert der sortierten Daten (bei gerader Anzahl: Mittel␣
 ↪der zwei mittleren): robust gegenüber Ausreißern
# und oft besser für schiefe Verteilungen (z.B. Einkommen, Preise, Wartezeiten).
```

[178]:
```
# Check for which numeric columns have null values

for label,content in df_temp.items():
    if pd.api.types.is_numeric_dtype(content):
        if pd.isnull(content).sum():
            print(label)
```

[179]:
```
df_temp.auctioneerID_is_missing.value_counts()
```

[179]:
```
auctioneerID_is_missing
False    392562
True      20136
Name: count, dtype: int64
```

[180]:
```
df_temp.isna().sum()
```

[180]:
```
SalesID                         0
SalePrice                       0
MachineID                       0
ModelID                         0
datasource                      0
auctioneerID                    0
YearMade                        0
MachineHoursCurrentMeter        0
UsageBand                  339028
fiModelDesc                     0
fiBaseModel                     0
```

fiSecondaryDesc	140727
fiModelSeries	354031
fiModelDescriptor	337882
ProductSize	216605
fiProductClassDesc	0
state	0
ProductGroup	0
ProductGroupDesc	0
Drive_System	305611
Enclosure	334
Forks	214983
Pad_Type	331602
Ride_Control	259970
Stick	331602
Transmission	224691
Turbocharged	331602
Blade_Extension	386715
Blade_Width	386715
Enclosure_Type	386715
Engine_Horsepower	386715
Hydraulics	82565
Pushblock	386715
Ripper	305753
Scarifier	386704
Tip_Control	386715
Tire_Size	315060
Coupler	192019
Coupler_System	367724
Grouser_Tracks	367823
Hydraulics_Flow	367823
Track_Type	310505
Undercarriage_Pad_Width	309782
Stick_Length	310437
Thumb	310366
Pattern_Changer	310437
Grouser_Type	310505
Backhoe_Mounting	331986
Blade_Type	330823
Travel_Controls	330821
Differential_Type	341134
Steering_Controls	341176
saleYear	0
saleMonth	0
saleDay	0
saleDayOfWeek	0
saleDayOfYear	0
auctioneerID_is_missing	0

```
MachineHoursCurrentMeter_is_missing         0
dtype: int64
```

[181]:
```
### Filling and turning categorical variables into numbers


#Check for columns which arent't numeric

for label,content in df_temp.items():
    if not pd.api.types.is_numeric_dtype(content):
        print(label)
```

```
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
```

[182]:
```
pd.Categorical(df_temp["state"]).dtype
```

[182]:
```
CategoricalDtype(categories=['Alabama', 'Alaska', 'Arizona', 'Arkansas',
'California',
                  'Colorado', 'Connecticut', 'Delaware', 'Florida', 'Georgia',
                  'Hawaii', 'Idaho', 'Illinois', 'Indiana', 'Iowa', 'Kansas',
                  'Kentucky', 'Louisiana', 'Maine', 'Maryland',
                  'Massachusetts', 'Michigan', 'Minnesota', 'Mississippi',
                  'Missouri', 'Montana', 'Nebraska', 'Nevada', 'New Hampshire',
                  'New Jersey', 'New Mexico', 'New York', 'North Carolina',
                  'North Dakota', 'Ohio', 'Oklahoma', 'Oregon', 'Pennsylvania',
                  'Puerto Rico', 'Rhode Island', 'South Carolina',
                  'South Dakota', 'Tennessee', 'Texas', 'Unspecified', 'Utah',
                  'Vermont', 'Virginia', 'Washington', 'Washington DC',
                  'West Virginia', 'Wisconsin', 'Wyoming'],
, ordered=True, categories_dtype=object)
```

[183]:
```
pd.Categorical(df_temp["state"]).codes # if missing value then -1
```

[183]:
```
array([43,  8,  8, ...,  4,  4,  4], shape=(412698,), dtype=int8)
```

[184]:
```
# Turn categorical variables into number and fill missing

for label,content in df_temp.items():
    if not pd.api.types.is_numeric_dtype(content):
        # add binary columns to indicate whether sample had missing value
        df_temp[label+"_is_missing"] = pd.isnull(content)
        #Turn categories into numbers and add +1
        df_temp[label] = pd.Categorical(content).codes+1
```

[185]:
```
df_temp.head()
```

[185]:

	SalesID	SalePrice	MachineID	ModelID	datasource	auctioneerID	YearMade	\
0	1646770	9500.0	1126363	8434	132	18.0	1974	
1	1821514	14000.0	1194089	10150	132	99.0	1980	
2	1505138	50000.0	1473654	4139	132	99.0	1978	
3	1671174	16000.0	1327630	8591	132	99.0	1980	
4	1329056	22000.0	1336053	4089	132	99.0	1984	

```
   MachineHoursCurrentMeter  UsageBand  fiModelDesc  ...  \
0                       0.0          0         4593  ...
1                       0.0          0         1820  ...
2                       0.0          0         2348  ...
3                       0.0          0         1819  ...
4                       0.0          0         2119  ...

   Undercarriage_Pad_Width_is_missing  Stick_Length_is_missing  \
0                                True                     True
1                                True                     True
2                                True                     True
3                                True                     True
4                                True                     True

   Thumb_is_missing  Pattern_Changer_is_missing  Grouser_Type_is_missing  \
0              True                        True                     True
1              True                        True                     True
2              True                        True                     True
3              True                        True                     True
4              True                        True                     True

   Backhoe_Mounting_is_missing  Blade_Type_is_missing  \
0                        False                  False
1                         True                   True
2                        False                  False
3                         True                   True
4                        False                  False

   Travel_Controls_is_missing  Differential_Type_is_missing  \
0                       False                          True
1                        True                         False
2                       False                          True
3                        True                         False
4                       False                          True

   Steering_Controls_is_missing
0                          True
1                         False
2                          True
3                         False
4                          True

[5 rows x 103 columns]
```

[186]:
```
df_temp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 412698 entries, 0 to 412697
```

```
Columns: 103 entries, SalesID to Steering_Controls_is_missing
dtypes: bool(46), float64(3), int16(4), int64(10), int8(40)
memory usage: 77.9 MB
```

[187]:
```
df_temp.head().T
```

[187]:

	0	1	2	3	4
SalesID	1646770	1821514	1505138	1671174	1329056
SalePrice	9500.0	14000.0	50000.0	16000.0	22000.0
MachineID	1126363	1194089	1473654	1327630	1336053
ModelID	8434	10150	4139	8591	4089
datasource	132	132	132	132	132
...	...	...	...	...	...
Backhoe_Mounting_is_missing	False	True	False	True	False
Blade_Type_is_missing	False	True	False	True	False
Travel_Controls_is_missing	False	True	False	True	False
Differential_Type_is_missing	True	False	True	False	True
Steering_Controls_is_missing	True	False	True	False	True

[103 rows x 5 columns]

[188]:
```
df_temp.isna().sum()[:70]
```

[188]:
```
SalesID                         0
SalePrice                       0
MachineID                       0
ModelID                         0
datasource                      0
                               ..
ProductSize_is_missing          0
fiProductClassDesc_is_missing   0
state_is_missing                0
ProductGroup_is_missing         0
ProductGroupDesc_is_missing     0
Length: 70, dtype: int64
```

Now that all of our data is numeric as well as our df has no missing values, we should be able to build a machine learning model

[189]:
```
df_temp.head()
```

[189]:

	SalesID	SalePrice	MachineID	ModelID	datasource	auctioneerID	YearMade	\
0	1646770	9500.0	1126363	8434	132	18.0	1974	
1	1821514	14000.0	1194089	10150	132	99.0	1980	
2	1505138	50000.0	1473654	4139	132	99.0	1978	
3	1671174	16000.0	1327630	8591	132	99.0	1980	
4	1329056	22000.0	1336053	4089	132	99.0	1984	

```
   MachineHoursCurrentMeter  UsageBand  fiModelDesc  ...  \
0                       0.0          0         4593  ...
1                       0.0          0         1820  ...
2                       0.0          0         2348  ...
3                       0.0          0         1819  ...
4                       0.0          0         2119  ...

   Undercarriage_Pad_Width_is_missing  Stick_Length_is_missing  \
0                                True                     True
1                                True                     True
2                                True                     True
3                                True                     True
4                                True                     True

   Thumb_is_missing  Pattern_Changer_is_missing  Grouser_Type_is_missing  \
0              True                        True                     True
1              True                        True                     True
2              True                        True                     True
3              True                        True                     True
4              True                        True                     True

   Backhoe_Mounting_is_missing  Blade_Type_is_missing  \
0                        False                  False
1                         True                   True
2                        False                  False
3                         True                   True
4                        False                  False

   Travel_Controls_is_missing  Differential_Type_is_missing  \
0                       False                          True
1                        True                         False
2                       False                          True
3                        True                         False
4                       False                          True

   Steering_Controls_is_missing
0                          True
1                         False
2                          True
3                         False
4                          True

[5 rows x 103 columns]
```

[190]:
```
%%time

#JN Function to calculate how much time the cell take
```

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

# Fit the model

model.fit(df_temp.drop("SalePrice", axis = 1), df_temp["SalePrice"])
```

```
CPU times: total: 32min 12s
Wall time: 5min 6s
```

[190]:
```
RandomForestRegressor(n_jobs=-1)
```

[191]:
```
# Score the model with train data wrong
model.score(df_temp.drop("SalePrice", axis = 1), df_temp["SalePrice"])
```

[191]:
```
0.9875665110400845
```

Question: Why doesn't the above metric reliable?

1.7.2 Splitting data into train/validation sets

[192]:
```
df_temp.saleYear
```

[192]:
```
0         1989
1         1989
2         1989
3         1989
4         1989
          ...
412693    2012
412694    2012
412695    2012
412696    2012
412697    2012
Name: saleYear, Length: 412698, dtype: int64
```

[193]:
```
df_temp.saleYear.value_counts()
```

[193]:
```
saleYear
2009    43849
2008    39767
2011    35197
2010    33390
2007    32208
2006    21685
2005    20463
2004    19879
2001    17594
```

```
2000    17415
2002    17246
2003    15254
1998    13046
1999    12793
2012    11573
1997     9785
1996     8829
1995     8530
1994     7929
1993     6303
1992     5519
1991     5109
1989     4806
1990     4529
Name: count, dtype: int64
```

[194]:
```python
# 1) Split We have data leakage because we calculate the median with past and␣
 ↪future datasets
df_val   = df_temp[df_temp.saleYear == 2012].copy()
df_train = df_temp[df_temp.saleYear != 2012].copy()

# 2) Imputation-Statistiken nur aus Training
num_cols = df_train.select_dtypes(include="number").columns
medians = df_train[num_cols].median()

df_train[num_cols] = df_train[num_cols].fillna(medians)

# 2) Imputation-Statistiken nur aus Validation
num_cols = df_val.select_dtypes(include="number").columns
medians = df_val[num_cols].median()

df_val[num_cols] = df_val[num_cols].fillna(medians)
```

[195]:
```python
# Split data into X and y

X_train,y_train = df_train.drop("SalePrice",axis = 1), df_train.SalePrice

# Split data into X and y

X_valid,y_valid = df_val.drop("SalePrice",axis = 1), df_val.SalePrice
```

1.7.3 Build an evaluation fuction

[196]:
```
# Create evaluation function (the competition uses RMSLE)
from sklearn.metrics import mean_squared_log_error, mean_absolute_error,r2_score

def rmsle (y_test,y_preds):
    """
    Calculates root mean squared log error between predictions and true labels.
    """
    return np.sqrt(mean_squared_log_error(y_test,y_preds))

# Create a function to evaluate model on a few diffrent levels
def show_scores(model):
    train_preds = model.predict(X_train)
    val_preds = model.predict(X_valid)
    scores = {"Training MAE":mean_absolute_error(y_train,train_preds),
    "Valid MAE": mean_absolute_error(y_valid,val_preds),
    "Training RMSLE":rmsle(y_train,train_preds),
    "Valid RMSLE":rmsle(y_valid,val_preds),
    "Training r^2":r2_score(y_train,train_preds),
    "Valid r^2":r2_score(y_valid,val_preds)}
    return scores
```

1.8 Testing our model on a subset (to tune the hyperparameters)

[197]:
```
model = RandomForestRegressor(n_jobs = -1,
                              random_state = 42)

#This takes far too long .... for experimenting
#model.fit(X_train,y_train)
```

[198]:
```
len(X_train)
```

[198]: 401125

[199]:
```
#Change max_samples value

model = RandomForestRegressor(n_jobs = -1, random_state = 42,max_samples=10000)
```

[200]:
```
%%time
##cutting down on the max number of samples each estimators can see improves␣
 ↪training time
model.fit(X_train,y_train)
```

```
CPU times: total: 1min 5s
Wall time: 10.8 s
```

[200]: RandomForestRegressor(max_samples=10000, n_jobs=-1, random_state=42)

[201]:
```
show_scores(model)
```

[201]:
```
{'Training MAE': 5561.298809224058,
 'Valid MAE': 7177.26365505919,
 'Training RMSLE': np.float64(0.25774537825697696),
 'Valid RMSLE': np.float64(0.29362638671089003),
 'Training r^2': 0.8606658995199188,
 'Valid r^2': 0.8320374995090507}
```

1.8.1 Hyperparameter tuning with RandomizedSearchCV

[202]:
```
%%time
from sklearn.model_selection import RandomizedSearchCV

#Diffrent RandomForestRegressor HP
rf_grid ={
    "n_estimators":np.arange(10,100,10),
    "max_depth":[None,3,5,10],
    "min_samples_split": np.arange(2,20,2),
    "min_samples_leaf":np.arange(1,20,2),
    "max_samples":[10000],
    "max_features":[0.5,1,"sqrt","auto"]
}

# Instantiate RandomizesSearchCV
rs_model = RandomizedSearchCV(RandomForestRegressor(n_jobs = -1, random_state =␣
 ↪42), param_distributions=rf_grid,
                              n_iter= 5, # very low because it take too long!
                              cv=5,
                              verbose = True)

#Fit the RSCV model
rs_model.fit(X_train,y_train)
```

```
Fitting 5 folds for each of 5 candidates, totalling 25 fits
CPU times: total: 1min 33s
Wall time: 36.5 s
```

[202]:
```
RandomizedSearchCV(cv=5,
                   estimator=RandomForestRegressor(n_jobs=-1, random_state=42),
                   n_iter=5,
                   param_distributions={'max_depth': [None, 3, 5, 10],
                                        'max_features': [0.5, 1, 'sqrt',
                                                         'auto'],
                                        'max_samples': [10000],
                                        'min_samples_leaf': array([ 1,  3,  5,
7,  9, 11, 13, 15, 17, 19]),
                                        'min_samples_split': array([ 2,  4,  6,
```

```
8, 10, 12, 14, 16, 18]),
                                                'n_estimators': array([10, 20, 30, 40,
50, 60, 70, 80, 90])},
                   verbose=True)
```

[203]:
```
#find the best HP for the model

rs_model.best_params_
```

[203]:
```
{'n_estimators': np.int64(30),
 'min_samples_split': np.int64(10),
 'min_samples_leaf': np.int64(5),
 'max_samples': 10000,
 'max_features': 'sqrt',
 'max_depth': 10}
```

[204]:
```
# Evaluate the randomized Search Model
show_scores(rs_model)
```

[204]:
```
{'Training MAE': 9024.775697210287,
 'Valid MAE': 10547.658931378968,
 'Training RMSLE': np.float64(0.3964490510077746),
 'Valid RMSLE': np.float64(0.41947442230936866),
 'Training r^2': 0.6780160387865647,
 'Valid r^2': 0.6808627887416161}
```

1.8.2 Train the model with the best hyperparameters

Note These were found after 100 iterations of RandomizedSearchCV.

[206]:
```
%%time

# Most ideal Hyperparameters

ideal_model = RandomForestRegressor(n_estimators = 40,
                                    min_samples_leaf=1,
                                    min_samples_split=14,
                                    max_features=0.5,
                                    n_jobs = -1,
                                    max_samples = None,
                                   random_state = 42 # so our results are␣
 ↪reproducable
)

#Fit the ideal model
ideal_model.fit(X_train,y_train)
```

```
CPU times: total: 5min 16s
Wall time: 53.6 s
```

[206]:
```
RandomForestRegressor(max_features=0.5, min_samples_split=14, n_estimators=40,
                      n_jobs=-1, random_state=42)
```

[207]:
```
show_scores(ideal_model)
```

[207]:
```
{'Training MAE': 2953.816113716348,
 'Valid MAE': 5951.247761444453,
 'Training RMSLE': np.float64(0.14469006962371855),
 'Valid RMSLE': np.float64(0.2452416398953833),
 'Training r^2': 0.9588145522577225,
 'Valid r^2': 0.8818019502450093}
```

1.9 Make Predictions on test data

[208]:
```
#Import the test data

df_test = pd.read_csv("data/Test.csv", low_memory = False,␣
↪parse_dates=["saledate"])

df_test.head()
```

[208]:

	SalesID	MachineID	ModelID	datasource	auctioneerID	YearMade	\
0	1227829	1006309	3168	121	3	1999	
1	1227844	1022817	7271	121	3	1000	
2	1227847	1031560	22805	121	3	2004	
3	1227848	56204	1269	121	3	2006	
4	1227863	1053887	22312	121	3	2005	

	MachineHoursCurrentMeter	UsageBand	saledate	fiModelDesc	...	\
0	3688.0	Low	2012-05-03	580G	...	
1	28555.0	High	2012-05-10	936	...	
2	6038.0	Medium	2012-05-10	EC210BLC	...	
3	8940.0	High	2012-05-10	330CL	...	
4	2286.0	Low	2012-05-10	650K	...	

	Undercarriage_Pad_Width	Stick_Length	Thumb	Pattern_Changer	\
0	NaN	NaN	NaN	NaN	
1	NaN	NaN	NaN	NaN	
2	None or Unspecified	9' 6"	Manual	None or Unspecified	
3	None or Unspecified	None or Unspecified	Manual	Yes	
4	NaN	NaN	NaN	NaN	

	Grouser_Type	Backhoe_Mounting	Blade_Type	Travel_Controls	\
0	NaN	NaN	NaN	NaN	
1	NaN	NaN	NaN	NaN	

```
2       Double                   NaN        NaN                   NaN
3       Triple                   NaN        NaN                   NaN
4          NaN  None or Unspecified        PAT  None or Unspecified

  Differential_Type Steering_Controls
0               NaN               NaN
1          Standard      Conventional
2               NaN               NaN
3               NaN               NaN
4               NaN               NaN

[5 rows x 52 columns]
```

[209]:
```
df_test.isna().sum()
```

[209]:
```
SalesID                       0
MachineID                     0
ModelID                       0
datasource                    0
auctioneerID                  0
YearMade                      0
MachineHoursCurrentMeter    10328
UsageBand                   10623
saledate                      0
fiModelDesc                   0
fiBaseModel                   0
fiSecondaryDesc            3975
fiModelSeries             10451
fiModelDescriptor          9433
ProductSize                6409
fiProductClassDesc            0
state                         0
ProductGroup                  0
ProductGroupDesc              0
Drive_System               9698
Enclosure                     2
Forks                      6149
Pad_Type                  10349
Ride_Control               8216
Stick                     10349
Transmission               7639
Turbocharged              10349
Blade_Extension           11806
Blade_Width               11806
Enclosure_Type            11806
Engine_Horsepower         11806
Hydraulics                 2142
```

```
Pushblock                 11806
Ripper                     9753
Scarifier                 11806
Tip_Control               11806
Tire_Size                  9679
Coupler                    4856
Coupler_System            10391
Grouser_Tracks            10391
Hydraulics_Flow           10391
Track_Type                 9063
Undercarriage_Pad_Width    9059
Stick_Length               9063
Thumb                      9062
Pattern_Changer            9063
Grouser_Type               9063
Backhoe_Mounting          10406
Blade_Type                10399
Travel_Controls           10399
Differential_Type         10328
Steering_Controls         10328
dtype: int64
```

[210]:
```
df_test.dtypes
```

[210]:
```
SalesID                          int64
MachineID                        int64
ModelID                          int64
datasource                       int64
auctioneerID                     int64
YearMade                         int64
MachineHoursCurrentMeter       float64
UsageBand                       object
saledate                datetime64[ns]
fiModelDesc                     object
fiBaseModel                     object
fiSecondaryDesc                 object
fiModelSeries                   object
fiModelDescriptor               object
ProductSize                     object
fiProductClassDesc              object
state                           object
ProductGroup                    object
ProductGroupDesc                object
Drive_System                    object
Enclosure                       object
Forks                           object
Pad_Type                        object
```

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

[211]:
```
df_test.columns
```

[211]:
```
Index(['SalesID', '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')
```

1.10 Preprocessing the data to the same format like X_Train

[212]:
```
def preprocessing_data(df):
    """
    Performs transformation on df and returns transformed df.
    """
    df["saleYear"] = df.saledate.dt.year
    df["saleMonth"] = df.saledate.dt.month
    df["saleDay"] = df.saledate.dt.day
    df["saleDayOfWeek"] = df.saledate.dt.dayofweek
    df["saleDayOfYear"] = df.saledate.dt.dayofyear

    df.drop("saledate", axis = 1, inplace = True)

    # Fill the numeric rows with median
    for label,content in df.items():
        if pd.api.types.is_numeric_dtype(content):
            if pd.isnull(content).sum():
            #Add a binary column which tells us if the data was missing
                df[label+"_is_missing"] = pd.isnull(content)
            #Fill missing numeric values with median
                df[label]= content.fillna(content.median())

    #Filled categorical missing datat and turned categories into numbers

        if not pd.api.types.is_numeric_dtype(content):
            df[label+"_is_missing"] = pd.isnull(content)
        #We add +1 to the category code because pandas encodes missing␣
↪categories with -1
            df[label]=pd.Categorical(content).codes+1

    return df
```

[213]:
```
df_test = preprocessing_data(df_test)

df_test.columns, df_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12457 entries, 0 to 12456
Columns: 101 entries, SalesID to Steering_Controls_is_missing
dtypes: bool(45), float64(1), int16(2), int32(5), int64(6), int8(42)
memory usage: 2.0 MB
```

[213]:
```
(Index(['SalesID', 'MachineID', 'ModelID', 'datasource', 'auctioneerID',
        'YearMade', 'MachineHoursCurrentMeter', 'UsageBand', 'fiModelDesc',
        'fiBaseModel',
        ...
        'Undercarriage_Pad_Width_is_missing', 'Stick_Length_is_missing',
```

```
        'Thumb_is_missing', 'Pattern_Changer_is_missing',
        'Grouser_Type_is_missing', 'Backhoe_Mounting_is_missing',
        'Blade_Type_is_missing', 'Travel_Controls_is_missing',
        'Differential_Type_is_missing', 'Steering_Controls_is_missing'],
       dtype='object', length=101),
 None)
```

[214]:
```
# Make predictions on updated Test data dont work because a feature is missing
#test_preds = ideal_model.predict(df_test)
```

[215]:
```
# We can find how the columns differ using sets
set(X_train.columns)-set(df_test.columns)
```

[215]:
```
{'auctioneerID_is_missing'}
```

[216]:
```
# Manually fit the column
df_test["auctioneerID_is_missing"] = False

df_test.head()
```

[216]:

	SalesID	MachineID	ModelID	datasource	auctioneerID	YearMade	\
0	1227829	1006309	3168	121	3	1999	
1	1227844	1022817	7271	121	3	1000	
2	1227847	1031560	22805	121	3	2004	
3	1227848	56204	1269	121	3	2006	
4	1227863	1053887	22312	121	3	2005	

	MachineHoursCurrentMeter	UsageBand	fiModelDesc	fiBaseModel	...	\
0	3688.0	2	499	180	...	
1	28555.0	1	831	292	...	
2	6038.0	3	1177	404	...	
3	8940.0	1	287	113	...	
4	2286.0	2	566	196	...	

	Stick_Length_is_missing	Thumb_is_missing	Pattern_Changer_is_missing	\
0	True	True	True	
1	True	True	True	
2	False	False	False	
3	False	False	False	
4	True	True	True	

	Grouser_Type_is_missing	Backhoe_Mounting_is_missing	\
0	True	True	
1	True	True	
2	False	True	
3	False	True	
4	True	False	

```
   Blade_Type_is_missing  Travel_Controls_is_missing  \
0                   True                        True
1                   True                        True
2                   True                        True
3                   True                        True
4                  False                       False

   Differential_Type_is_missing  Steering_Controls_is_missing  \
0                          True                          True
1                         False                         False
2                          True                          True
3                          True                          True
4                          True                          True

   auctioneerID_is_missing
0                    False
1                    False
2                    False
3                    False
4                    False

[5 rows x 102 columns]
```

[223]:
```
# nach model.fit(X_train, y_train)
fit_cols = ideal_model.feature_names_in_

# X_test / X_new exakt wie beim Fit ausrichten
X_new_aligned = df_test.reindex(columns=fit_cols, fill_value=0)



test_preds = ideal_model.predict(X_new_aligned)
```

[222]:
```
# We can find how the columns differ using sets
set(X_train.columns)-set(df_test.columns)
```

[222]:
```
set()
```

[225]:
```
len(test_preds)
```

[225]:
```
12457
```

[226]:
```
test_preds
```

[226]:
```
array([17030.00927386, 14355.53565165, 46623.08774286, ...,
       11964.85073347, 16496.71079281, 27119.99044029], shape=(12457,))
```

[227]:
```
# Format  predictions into the same format Kaggle is after

df_preds = pd.DataFrame()
df_preds ["SalesID"] = df_test["SalesID"]
df_preds ["SalesPrice"] = test_preds

df_preds
```

[227]:

	SalesID	SalesPrice
0	1227829	17030.009274
1	1227844	14355.535652
2	1227847	46623.087743
3	1227848	71680.261335
4	1227863	61762.999424
...	...	...
12452	6643171	39966.363007
12453	6643173	12049.704433
12454	6643184	11964.850733
12455	6643186	16496.710793
12456	6643196	27119.990440

[12457 rows x 2 columns]

[228]:
```
df_preds.to_csv("data/test_predictions.csv", index = False)
```

1.10.1 Feature Importance

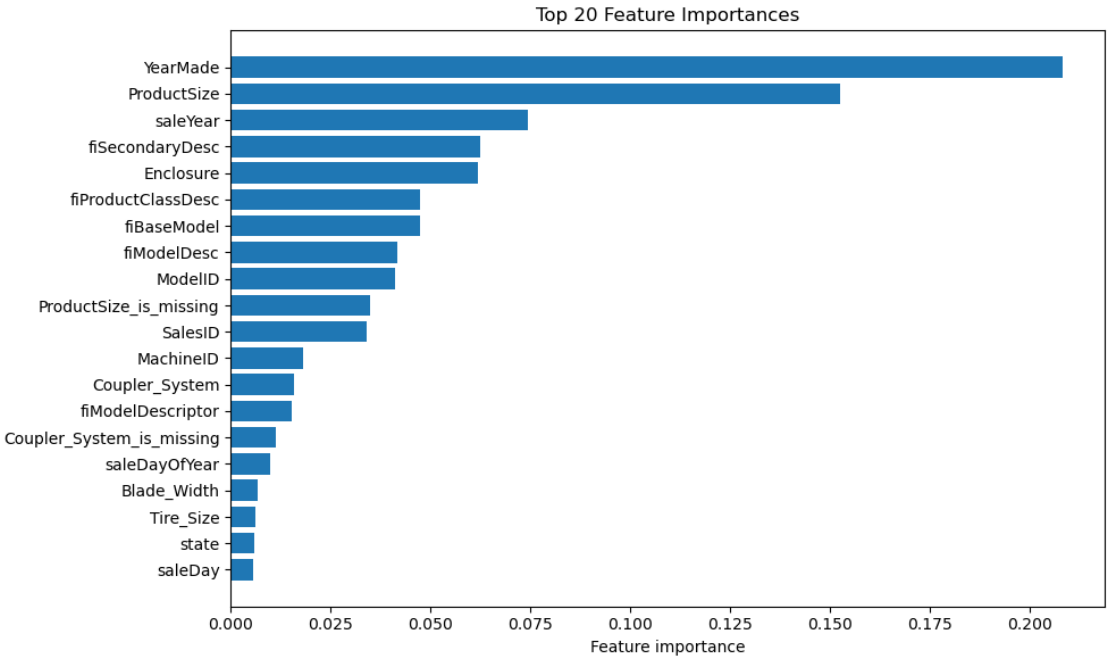
seeks to figure out which diffrent attributes of the data were most importance when it comes to predicting the target variable

[231]:
```
# FInd feature importance of our best models

len(ideal_model.feature_importances_)
```

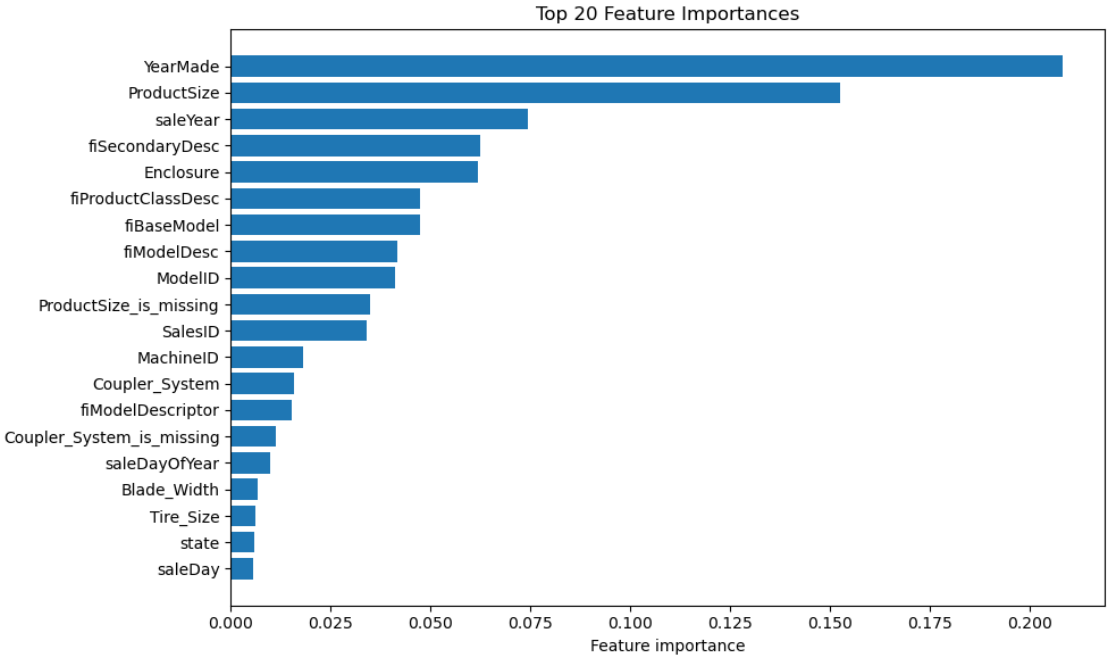
[231]: 102

[264]:
```
# Helper function for plotting feature importance
def plot_features(columns, importanceFactor, top_n=20):
    columns = np.array(columns)
    importanceFactor = np.array(importanceFactor)

    # sortiere nach Wichtigkeit (absteigend) und nimm Top-N
    idx = np.argsort(importanceFactor)[::-1][:top_n]
    cols = columns[idx]
    imps = importanceFactor[idx]

    fig, ax = plt.subplots(figsize=(10, 6))
    ax.barh(cols[::-1], imps[::-1])  # barh + umdrehen => wichtigste oben
```

```
    ax.set_xlabel("Feature importance")
    ax.set_title(f"Top {top_n} Feature Importances")
    plt.tight_layout()
    plt.show()

plot_features(X_new_aligned.columns, ideal_model.feature_importances_, top_n=20)
```

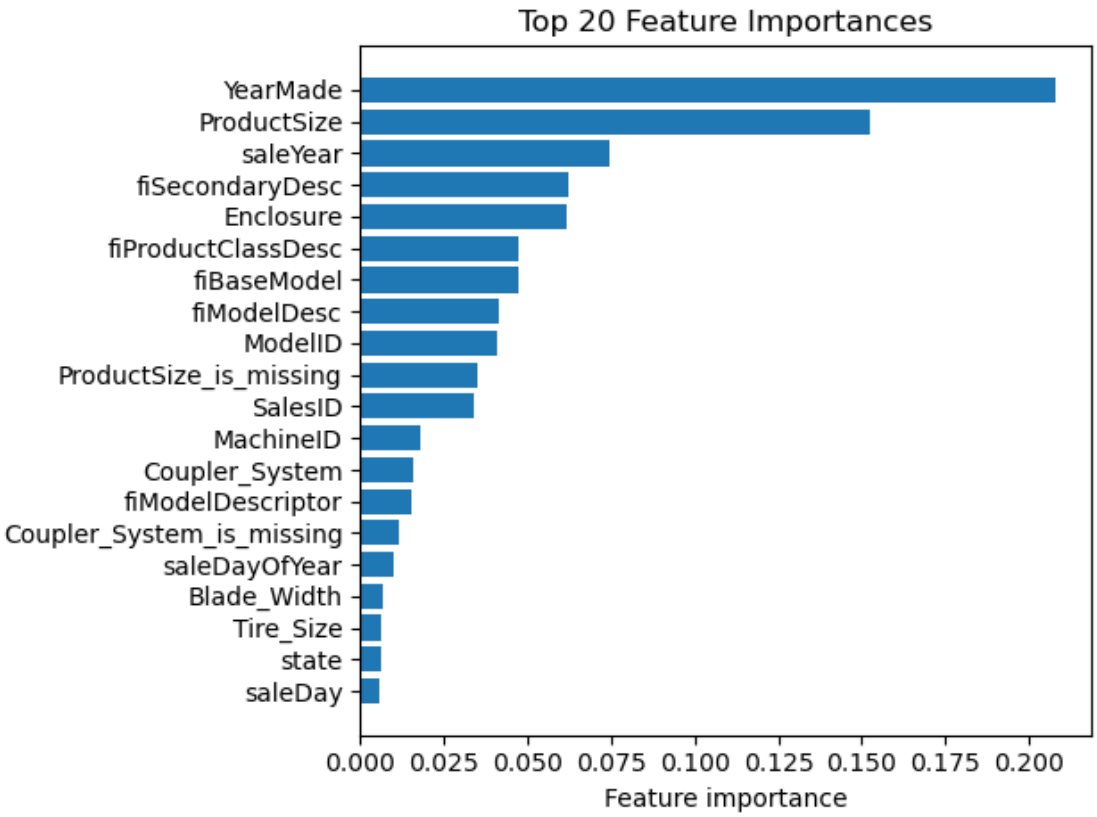


[265]:
```
plot_features(X_new_aligned.columns,ideal_model.feature_importances_)
```



[277]:
```
def plot_feature_other_way (columns, importances, n = 20):
    df = (pd.DataFrame({"features":columns,
                        "features_importances": importances})
          .sort_values("features_importances",ascending = False)
          .reset_index(drop=True))

    # PLot the dataframe
    fig, ax = plt.subplots()
    #ax.barh(np.flip(np.array(df["features"][:n:])), np.flip(np.
↪array(df["features_importances"][:n:])))
    ax.barh(df["features"][:n:][::-1], df["features_importances"][:n:][::-1])
    ax.set_xlabel("Feature importance")
    ax.set_title(f"Top {n} Feature Importances")
    plt.tight_layout()
    plt.show()
```

[278]:
```
plot_feature_other_way(X_new_aligned.columns,ideal_model.feature_importances_)
```



Question to finish: Why might knowing the feature importances of a trained machine learning model be helpful?

Final challenge: What other machine learning models could you try on our dataset? Hint checkout the regression section of scikit learn map or try to look at CatBoost.ai or XGBoost.ai

[ ]: