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. Problem defination

> **How well we can predict the future sale price of a bulldozer, given its characteristics and previous examples of how much similar bulldozer have been sold for?**

2. Data

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

There are 3 main datasets:

- **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.**

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/evaluation

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 minimize RMSLE (root mean squared log error).

4. Features

Kaggle provides a data dictionary detailing all of the features of the data. View this data here: D:\F Drive\Complete Machine Learning and Data Science Zero to Mastery\12. Milestone Project 2 Supervised Learning (Time Series Data)\bluebook-for-bulldozers

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sklearn
```

In [2]:

```
# import training and validation sets
df = pd.read_csv("data/bluebook-for-bulldozers/TrainAndValid.csv",
                 low_memory=False)
```

In [3]:

```
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
```

In [4]:

```
df.isna().sum()
```

Out[4]:

```
SalesID                         0
SalePrice                       0
MachineID                       0
ModelID                         0
datasource                      0
```

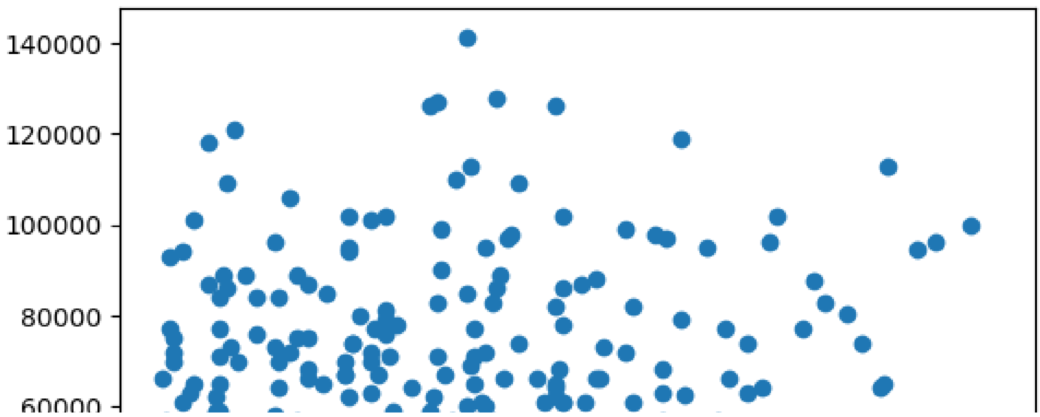
```
auctioneerID                 20136
YearMade                         0
MachineHoursCurrentMeter    265194
UsageBand                   339028
saledate                         0
fiModelDesc                      0
fiBaseModel                      0
fiSecondaryDesc             140727
fiModelSeries               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
```

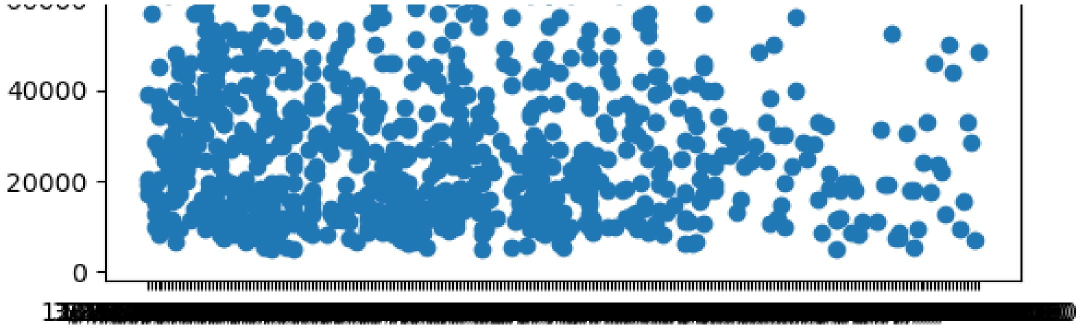
In [5]:

```python
fig, ax = plt.subplots()
ax.scatter(df["saledate"][:1000], df["SalePrice"][:1000])
```

Out[5]:

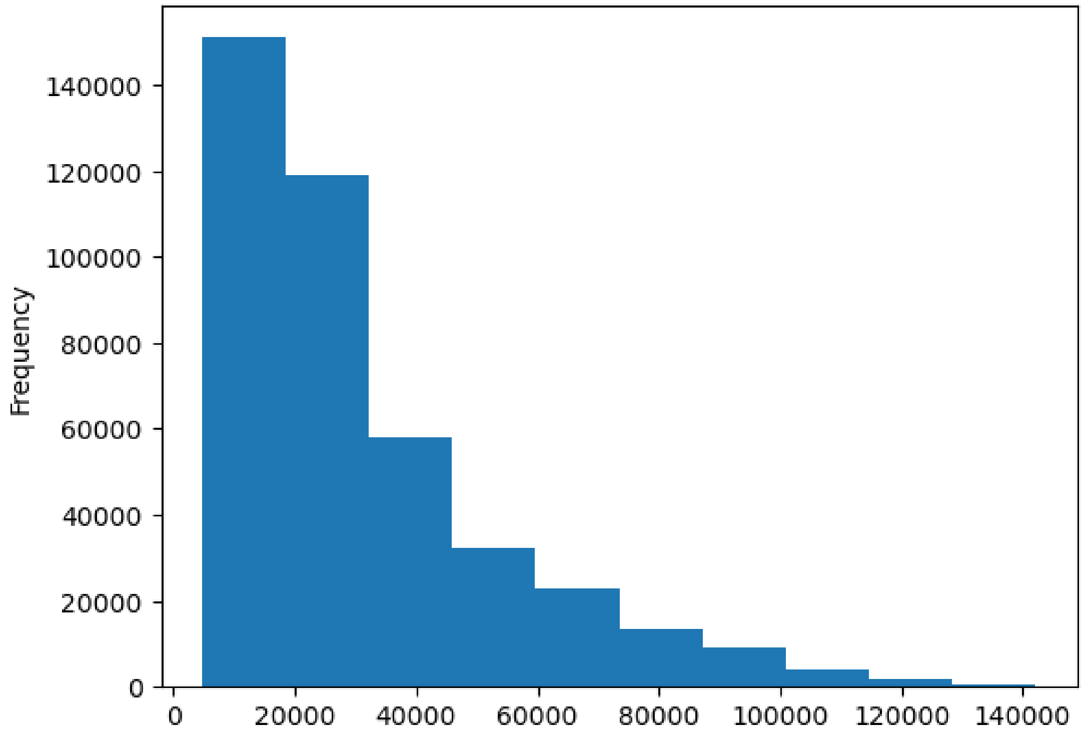
```
<matplotlib.collections.PathCollection at 0x15587646f70>
```





In [6]:

```
df.SalePrice.plot.hist();
```



Parsing date

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.**

In [7]:

```
# Import data again but this time parse dates
df = pd.read_csv("data/bluebook-for-bulldozers/TrainAndValid.csv",
                 low_memory=False,
                 parse_dates=["saledate"])
```

In [8]:

```
df.saledate.dtype
```

Out[8]:

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

In [9]:

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

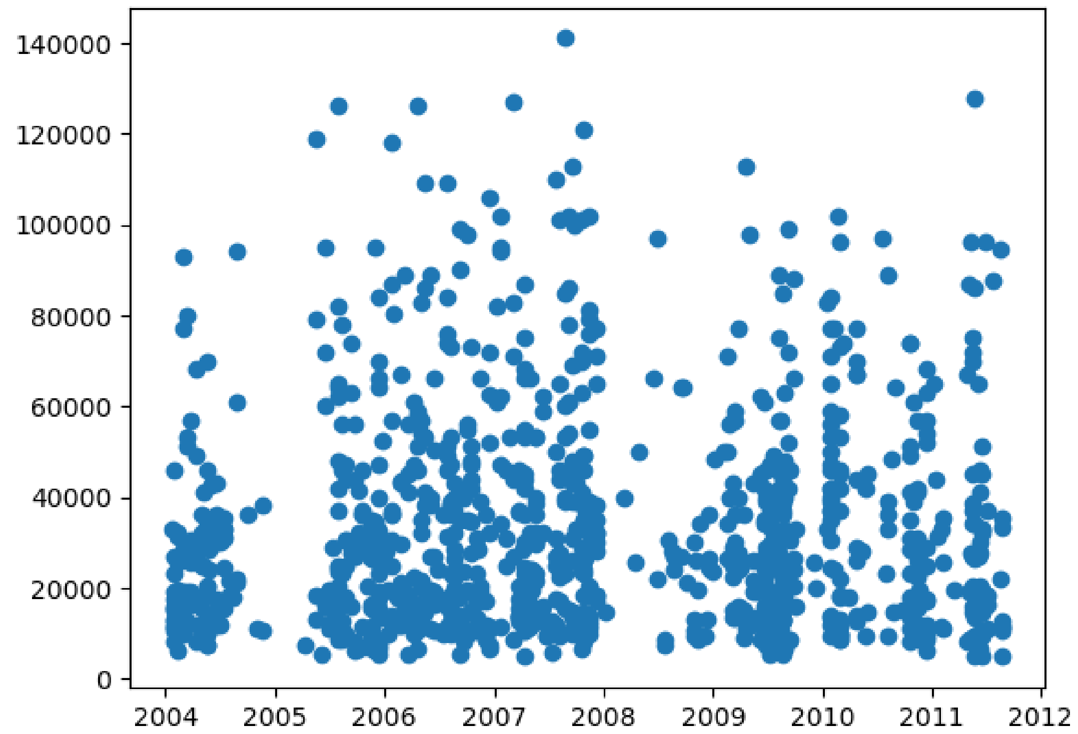
Out[9]:

```
0      2006-11-16
1      2004-03-26
2      2004-02-26
```

```
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]
```

In [10]:

```
fig, ax = plt.subplots()
ax.scatter(df["saledate"][:1000], df["SalePrice"][:1000]);
```



In [11]:

```
df.head()
```

Out[11]:

	SalesID	SalePrice	MachineID	ModelID	datasource	auctioneerID	YearMade	MachineHoursCurrentMeter	UsageBand	sa
0	1139246	66000.0	999089	3157	121	3.0	2004	68.0	Low	
1	1139248	57000.0	117657	77	121	3.0	1996	4640.0	Low	
2	1139249	10000.0	434808	7009	121	3.0	2001	2838.0	High	
3	1139251	38500.0	1026470	332	121	3.0	2001	3486.0	High	
4	1139253	11000.0	1057373	17311	121	3.0	2007	722.0	Medium	

5 rows × 53 columns

In [12]:

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

Out[12]:

	0	1	2	3	4
SalesID	1139246	1139248	1139249	1139251	1139253
SalePrice	66000.0	57000.0	10000.0	38500.0	11000.0
MachineID	999089	117657	434808	1026470	1057373
ModelID	3157	77	7009	332	17311
datasource	121	121	121	121	121
auctioneerID	3.0	3.0	3.0	3.0	3.0
YearMade	2004	1996	2001	2001	2007
MachineHoursCurrentMeter	68.0	4640.0	2838.0	3486.0	722.0
UsageBand	Low	Low	High	High	Medium
saledate	2006-11-16 00:00:00	2004-03-26 00:00:00	2004-02-26 00:00:00	2011-05-19 00:00:00	2009-07-23 00:00:00
fiModelDesc	521D	950FII	226	PC120-6E	S175
fiBaseModel	521	950	226	PC120	S175
fiSecondaryDesc	D	F	NaN	NaN	NaN
fiModelSeries	NaN	II	NaN	-6E	NaN
fiModelDescriptor	NaN	NaN	NaN	NaN	NaN
ProductSize	NaN	Medium	NaN	Small	NaN
fiProductClassDesc	Wheel Loader - 110.0 to 120.0 Horsepower	Wheel Loader - 150.0 to 175.0 Horsepower	Skid Steer Loader - 1351.0 to 1601.0 Lb Operat...	Hydraulic Excavator, Track - 12.0 to 14.0 Metr...	Skid Steer Loader - 1601.0 to 1751.0 Lb Operat...
state	Alabama	North Carolina	New York	Texas	New York
ProductGroup	WL	WL	SSL	TEX	SSL
ProductGroupDesc	Wheel Loader	Wheel Loader	Skid Steer Loaders	Track Excavators	Skid Steer Loaders
Drive_System	NaN	NaN	NaN	NaN	NaN
Enclosure	EROPS w AC	EROPS w AC	OROPS	EROPS w AC	EROPS
Forks	None or Unspecified	None or Unspecified	None or Unspecified	NaN	None or Unspecified
Pad_Type	NaN	NaN	NaN	NaN	NaN
Ride_Control	None or Unspecified	None or Unspecified	NaN	NaN	NaN
Stick	NaN	NaN	NaN	NaN	NaN
Transmission	NaN	NaN	NaN	NaN	NaN
Turbocharged	NaN	NaN	NaN	NaN	NaN
Blade_Extension	NaN	NaN	NaN	NaN	NaN
Blade_Width	NaN	NaN	NaN	NaN	NaN
Enclosure_Type	NaN	NaN	NaN	NaN	NaN
Engine_Horsepower	NaN	NaN	NaN	NaN	NaN
Hydraulics	2 Valve	2 Valve	Auxiliary	2 Valve	Auxiliary
Pushblock	NaN	NaN	NaN	NaN	NaN
Ripper	NaN	NaN	NaN	NaN	NaN
Scarifier	NaN	NaN	NaN	NaN	NaN
Tip_Control	NaN	NaN	NaN	NaN	NaN
Tire_Size	None or Unspecified	23.5	NaN	NaN	NaN
Coupler	None or Unspecified	None or Unspecified	None or Unspecified	None or Unspecified	None or Unspecified
Coupler_System	NaN	NaN	None or	NaN	None or

	0	1	2	3	4
Coupler_System	NaN	NaN	Unspecified	NaN	Unspecified
Grouser_Tracks	NaN	NaN	None or Unspecified	NaN	None or Unspecified
Hydraulics_Flow	NaN	NaN	Standard	NaN	Standard
Track_Type	NaN	NaN	NaN	NaN	NaN
Undercarriage_Pad_Width	NaN	NaN	NaN	NaN	NaN
Stick_Length	NaN	NaN	NaN	NaN	NaN
Thumb	NaN	NaN	NaN	NaN	NaN
Pattern_Changer	NaN	NaN	NaN	NaN	NaN
Grouser_Type	NaN	NaN	NaN	NaN	NaN
Backhoe_Mounting	NaN	NaN	NaN	NaN	NaN
Blade_Type	NaN	NaN	NaN	NaN	NaN
Travel_Controls	NaN	NaN	NaN	NaN	NaN
Differential_Type	Standard	Standard	NaN	NaN	NaN
Steering_Controls	Conventional	Conventional	NaN	NaN	NaN

In [13]:

```
df.saledate.head(20)
```

Out[13]:

```
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]
```

Sort DataFrame by saledate

When working with time series data, it's a good idea to sort it by date.

In [14]:

```
# Sort DataFrame in date order
df.sort_values(by=["saledate"], inplace=True, ascending=True)
df.saledate.head(20)
```

Out[14]:

```
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
```

```
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]
```

Make a copy of original DataFrame

I make a copy of the original DataFrame so when I manipulate the copy, I'll still got the original data.

In [15]:

```
# Make a copy
df_tmp = df.copy()
```

In [16]:

```
df_tmp.saledate.head(20)
```

Out[16]:

```
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]
```

Add datetime parameters for `saledate` column

In [17]:

```
df_tmp[:1].saledate.dt.year
```

Out[17]:

```
205615    1989
Name: saledate, dtype: int64
```

In [18]:

```
df_tmp[:1].saledate.dt.day
```

Out[18]:

```
205615    17
Name: saledate, dtype: int64
```

```
Name: saledate, dtype: int64
```

In [19]:

```
df_tmp[:1].saledate
```

Out[19]:

```
205615   1989-01-17
Name: saledate, dtype: datetime64[ns]
```

In [20]:

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

In [21]:

```
df_tmp.head().T
```

Out[21]:

	205615	274835	141296	212552	62755
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
auctioneerID	18.0	99.0	99.0	99.0	99.0
YearMade	1974	1980	1978	1980	1984
MachineHoursCurrentMeter	NaN	NaN	NaN	NaN	NaN
UsageBand	NaN	NaN	NaN	NaN	NaN
saledate	1989-01-17 00:00:00	1989-01-31 00:00:00	1989-01-31 00:00:00	1989-01-31 00:00:00	1989-01-31 00:00:00
fiModelDesc	TD20	A66	D7G	A62	D3B
fiBaseModel	TD20	A66	D7	A62	D3
fiSecondaryDesc	NaN	NaN	G	NaN	B
fiModelSeries	NaN	NaN	NaN	NaN	NaN
fiModelDescriptor	NaN	NaN	NaN	NaN	NaN
ProductSize	Medium	NaN	Large	NaN	NaN
fiProductClassDesc	Track Type Tractor, Dozer - 105.0 to 130.0 Hor...	Wheel Loader - 120.0 to 135.0 Horsepower	Track Type Tractor, Dozer - 190.0 to 260.0 Hor...	Wheel Loader - Unidentified	Track Type Tractor, Dozer - 20.0 to 75.0 Horse...
state	Texas	Florida	Florida	Florida	Florida
ProductGroup	TTT	WL	TTT	WL	TTT
ProductGroupDesc	Track Type Tractors	Wheel Loader	Track Type Tractors	Wheel Loader	Track Type Tractors
Drive_System	NaN	NaN	NaN	NaN	NaN
Enclosure	OROPS	OROPS	OROPS	EROPS	OROPS
Forks	NaN	None or Unspecified	NaN	None or Unspecified	NaN
Pad_Type	NaN	NaN	NaN	NaN	NaN
Ride_Control	NaN	None or Unspecified	NaN	None or Unspecified	NaN

	205615	274835	141296	212552	62755
Stick	NaN	NaN	NaN	NaN	NaN
Transmission	Direct Drive	NaN	Standard	NaN	Standard
Turbocharged	NaN	NaN	NaN	NaN	NaN
Blade_Extension	NaN	NaN	NaN	NaN	NaN
Blade_Width	NaN	NaN	NaN	NaN	NaN
Enclosure_Type	NaN	NaN	NaN	NaN	NaN
Engine_Horsepower	NaN	NaN	NaN	NaN	NaN
Hydraulics	2 Valve	2 Valve	2 Valve	2 Valve	2 Valve
Pushblock	NaN	NaN	NaN	NaN	NaN
Ripper	None or Unspecified	NaN	None or Unspecified	NaN	None or Unspecified
Scarifier	NaN	NaN	NaN	NaN	NaN
Tip_Control	NaN	NaN	NaN	NaN	NaN
Tire_Size	NaN	None or Unspecified	NaN	None or Unspecified	NaN
Coupler	NaN	None or Unspecified	NaN	None or Unspecified	NaN
Coupler_System	NaN	NaN	NaN	NaN	NaN
Grouser_Tracks	NaN	NaN	NaN	NaN	NaN
Hydraulics_Flow	NaN	NaN	NaN	NaN	NaN
Track_Type	NaN	NaN	NaN	NaN	NaN
Undercarriage_Pad_Width	NaN	NaN	NaN	NaN	NaN
Stick_Length	NaN	NaN	NaN	NaN	NaN
Thumb	NaN	NaN	NaN	NaN	NaN
Pattern_Changer	NaN	NaN	NaN	NaN	NaN
Grouser_Type	NaN	NaN	NaN	NaN	NaN
Backhoe_Mounting	None or Unspecified	NaN	None or Unspecified	NaN	None or Unspecified
Blade_Type	Straight	NaN	Straight	NaN	PAT
Travel_Controls	None or Unspecified	NaN	None or Unspecified	NaN	Lever
Differential_Type	NaN	Standard	NaN	Standard	NaN
Steering_Controls	NaN	Conventional	NaN	Conventional	NaN
saleYear	1989	1989	1989	1989	1989
saleMonth	1	1	1	1	1
saleDay	17	31	31	31	31
saleDayOfWeek	1	1	1	1	1
saleDayOfYear	17	31	31	31	31

In [22]:

```
# Now I've enriched our DataFrame with the date time features, now I can remove 'saledate
'
df_tmp.drop("saledate", axis=1, inplace=True)
```

In [23]:

```
# Check the values of different columns
df_tmp.state.value_counts()
```

Out[23]:

```
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: state, dtype: int64
```

5. Modelling

I've done enough EDA (exploratory data analysis). I will start to do some model-driven EDA.

In [24]:

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

model = RandomForestRegressor(n_jobs=-1,
                              random_state=42)

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

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [24], in <cell line: 7>()
```

```
      2 from sklearn.ensemble import RandomForestRegressor
      4 model = RandomForestRegressor(n_jobs=-1,
      5                                random_state=42)
----> 7 model.fit(df_tmp.drop("SalePrice", axis=1), df_tmp["SalePrice"])

File ~\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:331, in BaseForest.fit(sel
f, X, y, sample_weight)
    329 if issparse(y):
    330     raise ValueError("sparse multilabel-indicator for y is not supported.")
--> 331 X, y = self._validate_data(
    332     X, y, multi_output=True, accept_sparse="csc", dtype=DTYPE
    333 )
    334 if sample_weight is not None:
    335     sample_weight = _check_sample_weight(sample_weight, X)

File ~\anaconda3\lib\site-packages\sklearn\base.py:596, in BaseEstimator._validate_data(s
elf, X, y, reset, validate_separately, **check_params)
    594         y = check_array(y, input_name="y", **check_y_params)
    595     else:
--> 596         X, y = check_X_y(X, y, **check_params)
    597     out = X, y
    599 if not no_val_X and check_params.get("ensure_2d", True):

File ~\anaconda3\lib\site-packages\sklearn\utils\validation.py:1074, in check_X_y(X, y, a
ccept_sparse, accept_large_sparse, dtype, order, copy, force_all_finite, ensure_2d, allow
_nd, multi_output, ensure_min_samples, ensure_min_features, y_numeric, estimator)
   1069         estimator_name = _check_estimator_name(estimator)
   1070     raise ValueError(
   1071         f"{estimator_name} requires y to be passed, but the target y is None"
   1072     )
-> 1074 X = check_array(
   1075     X,
   1076     accept_sparse=accept_sparse,
   1077     accept_large_sparse=accept_large_sparse,
   1078     dtype=dtype,
   1079     order=order,
   1080     copy=copy,
   1081     force_all_finite=force_all_finite,
   1082     ensure_2d=ensure_2d,
   1083     allow_nd=allow_nd,
   1084     ensure_min_samples=ensure_min_samples,
   1085     ensure_min_features=ensure_min_features,
   1086     estimator=estimator,
   1087     input_name="X",
   1088 )
   1090 y = _check_y(y, multi_output=multi_output, y_numeric=y_numeric, estimator=estimat
or)
   1092 check_consistent_length(X, y)

File ~\anaconda3\lib\site-packages\sklearn\utils\validation.py:856, in check_array(array,
accept_sparse, accept_large_sparse, dtype, order, copy, force_all_finite, ensure_2d, allo
w_nd, ensure_min_samples, ensure_min_features, estimator, input_name)
    854         array = array.astype(dtype, casting="unsafe", copy=False)
    855     else:
--> 856         array = np.asarray(array, order=order, dtype=dtype)
    857 except ComplexWarning as complex_warning:
    858     raise ValueError(
    859         "Complex data not supported\n{}\n".format(array)
    860     ) from complex_warning

File ~\anaconda3\lib\site-packages\pandas\core\generic.py:2064, in NDFrame.__array__(self
, dtype)
   2063 def __array__(self, dtype: npt.DTypeLike | None = None) -> np.ndarray:
-> 2064     return np.asarray(self._values, dtype=dtype)

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

In [25]:

```
df_tmp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
Int64Index: 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  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: 182.6+ MB
```

In [26]:

```
df_tmp["UsageBand"].dtype
```

Out[26]:

```
dtype('O')
```

In [27]:

```
df_tmp.isna().sum()
```

Out[27]:

```
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
```

Converting string to categories

One way I can turn all our data into numbers is by converting them into pandas categories.

I can check different datatypes compatible with pandas here: https://pandas.pydata.org/pandas-docs/version/1.4/reference/general_utility_functions.html

In [28]:

```
pd.api.types.is_string_dtype(df_tmp["UsageBand"])
```

Out[28]:

```
True
```

In [29]:

```
# Find the columns which contains string

for label, content in df_tmp.items(): # here label means columns and content refers colum
n names
    if pd.api.types.is_string_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
```

In [30]:

```
# Creating a dictionary
random_dict = {"key1": "hello",
               "key2": "world"}

for key, value in random_dict.items():
    print(f"this is a key: {key}",
          f"this has a value: {value}")
```

```
this is a key: key1 this has a value: hello
this is a key: key2 this has a value: world
```

In [31]:

```
# This will turn all of the string values into category values

for label, content in df_tmp.items():
    if pd.api.types.is_string_dtype(content):
        df_tmp[label] = content.astype("category").cat.as_ordered() # as_ordered() funct
ion is for arrange data alphabetically
```

In [32]:

```
df_tmp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 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
```

```
 50  Differential_Type         71564 non-null   category
 51  Steering_Controls         71522 non-null   category
 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: category(44), float64(3), int64(10)
memory usage: 63.2 MB
```

In [33]:

```
df_tmp.state.cat.categories
```

Out[33]:

```
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')
```

In [34]:

```
df_tmp.state.cat.codes # Though the states showing as string/object but behind the scence state names datatyppe changed to int.
```

Out[34]:

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

In [35]:

```
df_tmp.state.value_counts()
```

Out[35]:

```
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
```

```
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: state, dtype: int64
```

Still we've a bunch of missing data...

In [36]:

```
# Check the missing data
df_tmp.isnull().sum()/len(df_tmp) * 100
```

Out[36]:

```
SalesID                      0.000000
SalePrice                    0.000000
MachineID                    0.000000
ModelID                      0.000000
datasource                   0.000000
auctioneerID                 4.879113
YearMade                     0.000000
MachineHoursCurrentMeter    64.258610
UsageBand                   82.149174
fiModelDesc                  0.000000
fiBaseModel                  0.000000
fiSecondaryDesc             34.099269
fiModelSeries               85.784520
fiModelDescriptor           81.871490
ProductSize                 52.485110
fiProductClassDesc           0.000000
state                        0.000000
ProductGroup                 0.000000
ProductGroupDesc             0.000000
Drive_System                74.051970
Enclosure                    0.080931
Forks                       52.092087
Pad_Type                    80.349796
Ride_Control                62.992794
Stick                       80.349796
Transmission                54.444412
```

```
Turbocharged               80.349796
Blade_Extension            93.704113
Blade_Width                93.704113
Enclosure_Type             93.704113
Engine_Horsepower          93.704113
Hydraulics                 20.006155
Pushblock                  93.704113
Ripper                     74.086378
Scarifier                  93.701448
Tip_Control                93.704113
Tire_Size                  76.341538
Coupler                    46.527727
Coupler_System             89.102443
Grouser_Tracks             89.126431
Hydraulics_Flow            89.126431
Track_Type                 75.237825
Undercarriage_Pad_Width    75.062637
Stick_Length               75.221348
Thumb                      75.204144
Pattern_Changer            75.221348
Grouser_Type               75.237825
Backhoe_Mounting           80.442842
Blade_Type                 80.161038
Travel_Controls            80.160553
Differential_Type          82.659475
Steering_Controls          82.669652
saleYear                    0.000000
saleMonth                   0.000000
saleDay                     0.000000
saleDayOfWeek               0.000000
saleDayOfYear               0.000000
dtype: float64
```

Save preprocessed data

In [37]:

```python
# Export current tmp dataframe
df_tmp.to_csv("data/bluebook-for-bulldozers/train_tmp.csv",
             index=False)
```

In [38]:

```python
# Import preprocessed data
df_tmp = pd.read_csv("data/bluebook-for-bulldozers/train_tmp.csv",
                     low_memory=False)
df_tmp.head().T
```

Out[38]:

	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
auctioneerID	18.0	99.0	99.0	99.0	99.0
YearMade	1974	1980	1978	1980	1984
MachineHoursCurrentMeter	NaN	NaN	NaN	NaN	NaN
UsageBand	NaN	NaN	NaN	NaN	NaN
fiModelDesc	TD20	A66	D7G	A62	D3B
fiBaseModel	TD20	A66	D7	A62	D3
fiSecondaryDesc	NaN	NaN	G	NaN	B

	0	1	2	3	4
fiModelSeries	NaN	NaN	NaN	NaN	NaN
fiModelDescriptor	NaN	NaN	NaN	NaN	NaN
ProductSize	Medium	NaN	Large	NaN	NaN
fiProductClassDesc	Track Type Tractor, Dozer - 105.0 to 130.0 Hor...	Wheel Loader - 120.0 to 135.0 Horsepower	Track Type Tractor, Dozer - 190.0 to 260.0 Hor...	Wheel Loader - Unidentified	Track Type Tractor, Dozer - 20.0 to 75.0 Horse...
state	Texas	Florida	Florida	Florida	Florida
ProductGroup	TTT	WL	TTT	WL	TTT
ProductGroupDesc	Track Type Tractors	Wheel Loader	Track Type Tractors	Wheel Loader	Track Type Tractors
Drive_System	NaN	NaN	NaN	NaN	NaN
Enclosure	OROPS	OROPS	OROPS	EROPS	OROPS
Forks	NaN	None or Unspecified	NaN	None or Unspecified	NaN
Pad_Type	NaN	NaN	NaN	NaN	NaN
Ride_Control	NaN	None or Unspecified	NaN	None or Unspecified	NaN
Stick	NaN	NaN	NaN	NaN	NaN
Transmission	Direct Drive	NaN	Standard	NaN	Standard
Turbocharged	NaN	NaN	NaN	NaN	NaN
Blade_Extension	NaN	NaN	NaN	NaN	NaN
Blade_Width	NaN	NaN	NaN	NaN	NaN
Enclosure_Type	NaN	NaN	NaN	NaN	NaN
Engine_Horsepower	NaN	NaN	NaN	NaN	NaN
Hydraulics	2 Valve	2 Valve	2 Valve	2 Valve	2 Valve
Pushblock	NaN	NaN	NaN	NaN	NaN
Ripper	None or Unspecified	NaN	None or Unspecified	NaN	None or Unspecified
Scarifier	NaN	NaN	NaN	NaN	NaN
Tip_Control	NaN	NaN	NaN	NaN	NaN
Tire_Size	NaN	None or Unspecified	NaN	None or Unspecified	NaN
Coupler	NaN	None or Unspecified	NaN	None or Unspecified	NaN
Coupler_System	NaN	NaN	NaN	NaN	NaN
Grouser_Tracks	NaN	NaN	NaN	NaN	NaN
Hydraulics_Flow	NaN	NaN	NaN	NaN	NaN
Track_Type	NaN	NaN	NaN	NaN	NaN
Undercarriage_Pad_Width	NaN	NaN	NaN	NaN	NaN
Stick_Length	NaN	NaN	NaN	NaN	NaN
Thumb	NaN	NaN	NaN	NaN	NaN
Pattern_Changer	NaN	NaN	NaN	NaN	NaN
Grouser_Type	NaN	NaN	NaN	NaN	NaN
Backhoe_Mounting	None or Unspecified	NaN	None or Unspecified	NaN	None or Unspecified
Blade_Type	Straight	NaN	Straight	NaN	PAT
Travel_Controls	None or Unspecified	NaN	None or Unspecified	NaN	Lever
Differential_Type	NaN	Standard	NaN	Standard	NaN
Steering_Controls	NaN	Conventional	NaN	Conventional	NaN

	0	1	2	3	4
Steering_Controls	NaN	Conventional	NaN	Conventional	NaN
saleYear	1989	1989	1989	1989	1989
saleMonth	1	1	1	1	1
saleDay	17	31	31	31	31
saleDayOfWeek	1	1	1	1	1
saleDayOfYear	17	31	31	31	31

In [39]:

```
df_tmp.isna().sum()
```

Out[39]:

```
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
```

```
saleDayOfYear                    0
dtype: int64
```

Fill missing values

1. **Fill numerical missing values**

In [40]:

```
# Finding which columns are numeric
for label, content in df_tmp.items():
    if pd.api.types.is_numeric_dtype(content):
        print(label)
```

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

In [41]:

```
# Checking numeric or non-numeric
df_tmp.ModelID
```

Out[41]:

```
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
```

In [42]:

```
# Check for which numeric columns have null values
for label, content in df_tmp.items():
    if pd.api.types.is_numeric_dtype(content):
        if pd.isnull(content).sum():
            print(label)
```

```
auctioneerID
MachineHoursCurrentMeter
```

In [43]:

```
# Fill empty/null numeric rows with the median
for label, content in df_tmp.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 or not
            df_tmp[label+"_is_missing"] = pd.isnull(content)
            # Fill missing numeric values with median (We are using median rather than m
ean, because If data contains outliers such as the 1000 in our example, then you would ty
pically rather use the median because otherwise the value of the mean would be dominated
```

```
by the outliers rather than the typical values)
            df_tmp[label] = content.fillna(content.median())
```

In [45]:

```python
# Check if there's any null numeric values
for label, content in df_tmp.items():
    if pd.api.types.is_numeric_dtype(content):
        if pd.isnull(content).sum():
            print(label)
```

In [46]:

```python
# Check to see how many examples were missing
df_tmp.auctioneerID_is_missing.value_counts()
```

Out[46]:

```
False    392562
True      20136
Name: auctioneerID_is_missing, dtype: int64
```

In [47]:

```python
df_tmp.isna().sum()
```

Out[47]:

```
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
```

```
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
```

Filling and turning categorical variables into numbers

In [48]:

```
# Check for coluumns which aren't numeric
for label, content in df_tmp.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
```

In [49]:

```
# Turn categorical variables into numbers and fill missing
for label, content in df_tmp.items():
    if not pd.api.types.is_numeric_dtype(content):
        # Add binary column to indicate whether samples had missing values
        df_tmp[label+"_is_missing"] = pd.isnull(content)
        # Turn categories into numbers and add +1
        df_tmp[label] = pd.Categorical(content).codes+1
```

In [50]:

```
pd.Categorical(df_tmp["state"]).codes+1
```

Out[50]:

```
array([44,  9,  9, ...,  5,  5,  5], dtype=int8)
```

In [51]:

```
pd.Categorical(df_tmp["UsageBand"]).codes # NOTE: for missing values -1 is showing. So,
to fill those missing values we need to add +1
```

Out[51]:

```
array([0, 0, 0, ..., 0, 0, 0], dtype=int8)
```

In [52]:

```
df_tmp.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
```

In [53]:

```
df_tmp.head().T
```

Out[53]:

	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 × 5 columns

In [54]:

```
df_tmp.isna().sum()
```

Out[54]:

```
SalesID                        0
SalePrice                      0
```

```
SalePrice                        0
MachineID                        0
ModelID                          0
datasource                       0
                                ..
Backhoe_Mounting_is_missing      0
Blade_Type_is_missing            0
Travel_Controls_is_missing       0
Differential_Type_is_missing     0
Steering_Controls_is_missing     0
Length: 103, dtype: int64
```

Now that all of data is numeric as well as my dataframe has no missing values, I should be able to build a machine learninng model.

In [55]:

```
len(df_tmp)
```

Out[55]:

```
412698
```

In [56]:

```
%%time
#Instantiated model
model = RandomForestRegressor(n_jobs=-1,
                              random_state=42)

# Fit the model
model.fit(df_tmp.drop("SalePrice", axis=1), df_tmp["SalePrice"])
```

```
CPU times: total: 28min 40s
Wall time: 1min 54s
```

Out[56]:

```
▾            RandomForestRegressor
RandomForestRegressor(n_jobs=-1, random_state=42)
```

In [57]:

```
# Score the model
model.score(df_tmp.drop("SalePrice", axis=1), df_tmp["SalePrice"])
```

Out[57]:

```
0.9875468079970562
```

Spliting data into train/validation sets

In [58]:

```
df_tmp.saleYear
```

Out[58]:

```
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
```

In [59]:

```
df_tmp.saleYear.value_counts()
```

Out[59]:

```
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: saleYear, dtype: int64
```

In [60]:

```
# Spliting data into training and validation
df_val = df_tmp[df_tmp.saleYear == 2012]
df_train = df_tmp[df_tmp.saleYear != 2012]

len(df_train), len(df_val)
```

Out[60]:

```
(401125, 11573)
```

In [61]:

```
# Split data into X and y
X_train, y_train = df_train.drop("SalePrice", axis=1), df_train.SalePrice
X_valid, y_valid = df_val.drop("SalePrice", axis=1), df_val.SalePrice
X_train.shape, y_train.shape, X_valid.shape, y_valid.shape
```

Out[61]:

```
((401125, 102), (401125,), (11573, 102), (11573,))
```

In [62]:

```
y_train
```

Out[62]:

```
0          9500.0
1         14000.0
2         50000.0
3         16000.0
4         22000.0
           ...
401120    29000.0
401121    11000.0
401122    11000.0
401123    18000.0
401124    13500.0
Name: SalePrice, Length: 401125, dtype: float64
```

Building an evaluation function

In [64]:

```
# Create evaluation function (the competation uses RMSLE)
from sklearn.metrics import mean_squared_log_error, mean_absolute_error, r2_score

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

# Create a function to evaluate model on a few different values
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
```

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

In [65]:

```
# # This takes far too long (time)... for experimenting

# %%time
# model = RandomForestRegressor(n_jobs=-1,
#                               random_state=42)

# model.fit(X_train, y_train)
```

In [66]:

```
len(X_train)
```

Out[66]:

```
401125
```

In [67]:

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

In [68]:

```
%%time
# Cutting down the max number of samples each estimator can see imporves training time.
model.fit(X_train, y_train)
```

```
CPU times: total: 50.2 s
Wall time: 3.6 s
```

Out[68]:

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

In [69]:

```
(X_train.shape[0] * 100) / 1000000
```

Out[69]:

```
40.1125
```

In [70]:

```
10000 * 100
```

Out[70]:

```
1000000
```

In [71]:

```
show_scores(model)
```

Out[71]:

```
{'Training MAE': 5561.2988092240585,
 'Valid MAE': 7177.26365505919,
 'Training RMSLE': 0.257745378256977,
 'Valid RMSLE': 0.29362638671089003,
 'Training R^2': 0.8606658995199189,
 'Valid R^2': 0.8320374995090507}
```

Hyperparameter tunning with RandomizedSearchCV

In [72]:

```
%%time
from sklearn.model_selection import RandomizedSearchCV

# Different RandomForestRegressor hyperparameters
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_features": [0.5, 1, "sqrt", "auto"],
           "max_samples": [10000]}

# Instantitate RandomizedSearchCV model
rs_model = RandomizedSearchCV(RandomForestRegressor(n_jobs=-1,
                                                    random_state=42),
                              param_distributions=rf_grid,
                              n_iter=100,
                              cv=5,
                              verbose=True)
# Fit the RandomizedSearchCV model
rs_model.fit(X_train, y_train)
```

```
Fitting 5 folds for each of 100 candidates, totalling 500 fits
```

```
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
```

```
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
```

```
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
```

```
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
```

```
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
```

```
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
```

```
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
```

```
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
```

```
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:416: Future
Warning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it
is also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
```

```
CPU times: total: 6min 52s
Wall time: 11min 10s
```

Out[72]:

```
▸ RandomizedSearchCV
▸ estimator: RandomForestRegressor
  ▸ RandomForestRegressor
```

In [73]:

```python
# Find the best model hyperparameters
rs_model.best_params_
```

Out[73]:

```
{'n_estimators': 70,
 'min_samples_split': 18,
 'min_samples_leaf': 1,
 'max_samples': 10000,
 'max_features': 'auto',
 'max_depth': None}
```

In [74]:

```python
# Evaluate the RandomizedSearch model
show_scores(rs_model)
```

Out[74]:

```
{'Training MAE': 5868.130564628453,
 'Valid MAE': 7402.897892083917,
 'Training RMSLE': 0.26820268248315415,
 'Valid RMSLE': 0.299808571068006,
 'Training R^2': 0.8447963614628694,
 'Valid R^2': 0.8190401132153241}
```

Training a model with best hyperparameter

Note: These were found after 100 iterations of `RandomizedSearchCV`

In [75]:

```python
%%time

# Most ideal hyperparameters
ideal_model = RandomForestRegressor(n_estimators=70,
                                    min_samples_split=2,
                                    min_samples_leaf=3,
                                    max_features=0.5,
                                    n_jobs=-1,
                                    max_samples=None,
                                    max_depth=None,
                                    random_state=42)  # Random state so our results are r
eproducable

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

```
CPU times: total: 9min 25s
Wall time: 38.9 s
```

Out[75]:

```
RandomForestRegressor
RandomForestRegressor(max_features=0.5, min_samples_leaf=3, n_estimators=70,
                      n_jobs=-1, random_state=42)
```

In [76]:

```
# Scores for ideal model (trained on all data)
show_scores(ideal_model)
```

Out[76]:

```
{'Training MAE': 2517.532396966851,
 'Valid MAE': 5919.7841407788055,
 'Training RMSLE': 0.12836315365062645,
 'Valid RMSLE': 0.24361757440413018,
 'Training R^2': 0.9677638308179852,
 'Valid R^2': 0.8818719316864689}
```

In [77]:

```
# Scores for rs_model (only on 10000 data)
show_scores(rs_model)
```

Out[77]:

```
{'Training MAE': 5868.130564628453,
 'Valid MAE': 7402.897892083917,
 'Training RMSLE': 0.26820268248315415,
 'Valid RMSLE': 0.299808571068006,
 'Training R^2': 0.8447963614628694,
 'Valid R^2': 0.8190401132153241}
```

Make predictions on test data

In [78]:

```
# Import the test data
df_test = pd.read_csv("data/bluebook-for-bulldozers/Test.csv",
                      low_memory=False,
                      parse_dates=["saledate"])
df_test.head()
```

Out[78]:

	SalesID	MachineID	ModelID	datasource	auctioneerID	YearMade	MachineHoursCurrentMeter	UsageBand	saledate	fiM
0	1227829	1006309	3168	121	3	1999	3688.0	Low	2012-05-03	
1	1227844	1022817	7271	121	3	1000	28555.0	High	2012-05-10	
2	1227847	1031560	22805	121	3	2004	6038.0	Medium	2012-05-10	E
3	1227848	56204	1269	121	3	2006	8940.0	High	2012-05-10	
4	1227863	1053887	22312	121	3	2005	2286.0	Low	2012-05-10	

5 rows × 52 columns

In [79]:

```
# Make predictions on test dataset
test_preds = ideal_model.predict(df_test)
```

```
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\base.py:493: FutureWarning: The
feature names should match those that were passed during fit. Starting version 1.2, an er
ror will be raised.
Feature names unseen at fit time:
- saledate
Feature names seen at fit time, yet now missing:
- Backhoe_Mounting_is_missing
- Blade_Extension_is_missing
- Blade_Type_is_missing
- Blade_Width_is_missing
- Coupler_System_is_missing
- ...

  warnings.warn(message, FutureWarning)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [79], in <cell line: 2>()
      1 # Make predictions on test dataset
----> 2 test_preds = ideal_model.predict(df_test)

File ~\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:991, in ForestRegressor.pr
edict(self, X)
    989 check_is_fitted(self)
    990 # Check data
--> 991 X = self._validate_X_predict(X)
    993 # Assign chunk of trees to jobs
    994 n_jobs, _, _ = _partition_estimators(self.n_estimators, self.n_jobs)

File ~\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:605, in BaseForest._valida
te_X_predict(self, X)
    602 """
    603 Validate X whenever one tries to predict, apply, predict_proba."""
    604 check_is_fitted(self)
--> 605 X = self._validate_data(X, dtype=DTYPE, accept_sparse="csr", reset=False)
    606 if issparse(X) and (X.indices.dtype != np.intc or X.indptr.dtype != np.intc):
    607     raise ValueError("No support for np.int64 index based sparse matrices")

File ~\anaconda3\lib\site-packages\sklearn\base.py:577, in BaseEstimator._validate_data(s
elf, X, y, reset, validate_separately, **check_params)
    575     raise ValueError("Validation should be done on X, y or both.")
    576 elif not no_val_X and no_val_y:
--> 577     X = check_array(X, input_name="X", **check_params)
    578     out = X
    579 elif no_val_X and not no_val_y:

File ~\anaconda3\lib\site-packages\sklearn\utils\validation.py:856, in check_array(array,
accept_sparse, accept_large_sparse, dtype, order, copy, force_all_finite, ensure_2d, allo
w_nd, ensure_min_samples, ensure_min_features, estimator, input_name)
    854         array = array.astype(dtype, casting="unsafe", copy=False)
    855     else:
--> 856         array = np.asarray(array, order=order, dtype=dtype)
    857 except ComplexWarning as complex_warning:
    858     raise ValueError(
    859         "Complex data not supported\n{}\n".format(array)
    860     ) from complex_warning

File ~\anaconda3\lib\site-packages\pandas\core\generic.py:2064, in NDFrame.__array__(self
, dtype)
   2063 def __array__(self, dtype: npt.DTypeLike | None = None) -> np.ndarray:
-> 2064     return np.asarray(self._values, dtype=dtype)

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

Preprocessing the test dataset (getting the test dataset in the same format as our training dataset)

In [80]:

```
def preprocess_data(df):
    """
    Performes transformations 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 or not
                df[label+"_is_missing"] = pd.isnull(content)
                # Fill missing numeric values with median (We are using median rather th
an mean, because If data contains outliers such as the 1000 in our example, then you woul
d typically rather use the median because otherwise the value of the mean would be domina
ted by the outliers rather than the typical values)
                df[label] = content.fillna(content.median())
        # Fill categorical missing data and turn categories into numbers
    for label, content in df.items():
        if not pd.api.types.is_numeric_dtype(content):
            # Add binary column to indicate whether samples had missing values
            df[label+"_is_missing"] = pd.isnull(content)
            # We add +1 to the category code because pandas encodes missing categories as
-1
            df[label] = pd.Categorical(content).codes+1

    return df
```

In [81]:

```
# Process the test data
df_test = preprocess_data(df_test)
df_test.head()
```

Out[81]:

	SalesID	MachineID	ModelID	datasource	auctioneerID	YearMade	MachineHoursCurrentMeter	UsageBand	fiModelDesc
0	1227829	1006309	3168	121	3	1999	3688.0	2	499
1	1227844	1022817	7271	121	3	1000	28555.0	1	831
2	1227847	1031560	22805	121	3	2004	6038.0	3	1177
3	1227848	56204	1269	121	3	2006	8940.0	1	287
4	1227863	1053887	22312	121	3	2005	2286.0	2	566

5 rows × 101 columns

In [82]:

```
# Make predictions on updated test data
test_preds = ideal_model.predict(df_test)
```

```
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\base.py:493: FutureWarning: The
feature names should match those that were passed during fit. Starting version 1.2, an er
ror will be raised.
Feature names seen at fit time, yet now missing:
- auctioneerID_is_missing

  warnings.warn(message, FutureWarning)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
```

```
ValueError                                Traceback (most recent call last)
Input In [82], in <cell line: 2>()
      1 # Make predictions on updated test data
----> 2 test_preds = ideal_model.predict(df_test)

File ~\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:991, in ForestRegressor.predict(self, X)
    989 check_is_fitted(self)
    990 # Check data
--> 991 X = self._validate_X_predict(X)
    993 # Assign chunk of trees to jobs
    994 n_jobs, _, _ = _partition_estimators(self.n_estimators, self.n_jobs)

File ~\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:605, in BaseForest._validate_X_predict(self, X)
    602 """
    603 Validate X whenever one tries to predict, apply, predict_proba."""
    604 check_is_fitted(self)
--> 605 X = self._validate_data(X, dtype=DTYPE, accept_sparse="csr", reset=False)
    606 if issparse(X) and (X.indices.dtype != np.intc or X.indptr.dtype != np.intc):
    607     raise ValueError("No support for np.int64 index based sparse matrices")

File ~\anaconda3\lib\site-packages\sklearn\base.py:600, in BaseEstimator._validate_data(self, X, y, reset, validate_separately, **check_params)
    597     out = X, y
    599 if not no_val_X and check_params.get("ensure_2d", True):
--> 600     self._check_n_features(X, reset=reset)
    602 return out

File ~\anaconda3\lib\site-packages\sklearn\base.py:400, in BaseEstimator._check_n_features(self, X, reset)
    397     return
    399 if n_features != self.n_features_in_:
--> 400     raise ValueError(
    401         f"X has {n_features} features, but {self.__class__.__name__} "
    402         f"is expecting {self.n_features_in_} features as input."
    403     )

ValueError: X has 101 features, but RandomForestRegressor is expecting 102 features as input.
```

In [83]:

```
X_train.head()
```

Out[83]:

	SalesID	MachineID	ModelID	datasource	auctioneerID	YearMade	MachineHoursCurrentMeter	UsageBand	fiModelDesc
0	1646770	1126363	8434	132	18.0	1974	0.0	0	4593
1	1821514	1194089	10150	132	99.0	1980	0.0	0	1820
2	1505138	1473654	4139	132	99.0	1978	0.0	0	2348
3	1671174	1327630	8591	132	99.0	1980	0.0	0	1819
4	1329056	1336053	4089	132	99.0	1984	0.0	0	2119

5 rows × 102 columns

In [84]:

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

Out[84]:

```
{'auctioneerID_is_missing'}
```

In [85]:

```
# Manually adjust df_test to have auctioneerID_is_missing column
```

```python
df_test["auctioneerID_is_missing"] = False
df_test.head()
```

Out[85]:

	SalesID	MachineID	ModelID	datasource	auctioneerID	YearMade	MachineHoursCurrentMeter	UsageBand	fiModelDesc
0	1227829	1006309	3168	121	3	1999	3688.0	2	499
1	1227844	1022817	7271	121	3	1000	28555.0	1	831
2	1227847	1031560	22805	121	3	2004	6038.0	3	1177
3	1227848	56204	1269	121	3	2006	8940.0	1	287
4	1227863	1053887	22312	121	3	2005	2286.0	2	566

5 rows × 102 columns

Finally test dataframe has the same features as the training dataframe and now I can make predictions!

In [86]:

```python
# Make predictions on the test data
test_preds = ideal_model.predict(df_test)
```

```
C:\Users\Hero Clament\anaconda3\lib\site-packages\sklearn\base.py:493: FutureWarning: The feature names should match those that were passed during fit. Starting version 1.2, an er
ror will be raised.
Feature names must be in the same order as they were in fit.

  warnings.warn(message, FutureWarning)
```

In [87]:

```python
test_preds
```

Out[87]:

```
array([19817.21371882, 20122.20068027, 51645.47488226, ...,
       13902.27581942, 20413.00896722, 30952.35752343])
```

In [88]:

```python
len(test_preds)
```

Out[88]:

```
12457
```

I've made some predictions but they are not in the same format Kaggle is asking for:
https://www.kaggle.com/competitions/bluebook-for-bulldozers/overview/evaluation

In [89]:

```python
# 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
```

Out[89]:

	SalesID	SalesPrice
0	1227829	19817.213719
1	1227844	20122.200680
2	1227847	51645.474882
3	1227848	61693.202948

	SalesID	SalesPrice
4	1227863	394[illegible]
...	...	...
12452	6643171	47432.284580
12453	6643173	15891.095522
12454	6643184	13902.275819
12455	6643186	20413.008967
12456	6643196	30952.357523

12457 rows × 2 columns

In [90]:

```
# Export prediction data to csv
df_preds.to_csv("data/bluebook-for-bulldozers/test_predictions.csv", index=False)
```

Feature importance

Feature impotance seeks to fugure out which different attributes or features of the dataset were most important when need to predict the target value (SalePrice)

In [91]:

```
# Find feature importance of our ideal model
ideal_model.feature_importances_, len(ideal_model.feature_importances_)
```

Out[91]:

```
(array([3.60960567e-02, 2.02177121e-02, 3.86157178e-02, 1.94105532e-03,
        3.78908183e-03, 2.04181240e-01, 3.26628759e-03, 1.11768940e-03,
        4.59202806e-02, 4.10168720e-02, 6.54292748e-02, 4.01984150e-03,
        1.52024105e-02, 1.50716489e-01, 4.72622672e-02, 7.12267097e-03,
        2.67254321e-03, 1.98210252e-03, 3.62633952e-03, 6.02980419e-02,
        4.92716722e-04, 6.37713685e-05, 1.17656302e-03, 2.86932591e-04,
        9.49393707e-04, 3.01793841e-05, 1.73117640e-03, 7.40277708e-03,
        6.89207470e-04, 1.29833277e-03, 3.83225459e-03, 2.87894201e-03,
        3.82986895e-03, 1.79518107e-03, 4.95762192e-04, 6.27456507e-03,
        8.88684450e-04, 1.41856592e-02, 4.01781038e-04, 3.86466881e-03,
        1.12954041e-03, 9.88219772e-04, 2.15663678e-03, 6.92267712e-04,
        4.87480250e-04, 3.75179793e-04, 2.74313846e-04, 1.74030601e-03,
        1.14402467e-03, 1.75339734e-04, 4.97037057e-04, 7.28930089e-02,
        4.92682705e-03, 7.27169970e-03, 3.68050204e-03, 1.12618568e-02,
        1.96856176e-04, 1.64530102e-03, 4.32044558e-04, 0.00000000e+00,
        0.00000000e+00, 3.00276175e-03, 1.52018155e-03, 6.87192417e-03,
        3.09070677e-02, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
        0.00000000e+00, 6.35293804e-04, 1.53418365e-06, 1.11549648e-04,
        7.43348598e-06, 1.65009692e-04, 2.68785369e-06, 3.62241039e-04,
        1.37263502e-05, 2.68817995e-03, 2.26370317e-03, 5.63107573e-03,
        1.93814933e-04, 1.76818016e-03, 4.19436426e-05, 7.65459051e-04,
        2.20660259e-03, 1.30943132e-03, 2.61594775e-03, 1.51268708e-04,
        1.35002335e-02, 1.32066854e-03, 1.36272524e-03, 5.02479576e-05,
        1.18068999e-04, 4.51654367e-05, 1.23834936e-04, 9.40691761e-05,
        3.50127461e-05, 3.41651301e-04, 1.67564278e-04, 1.61629835e-04,
        3.14984997e-04, 9.62904946e-05]),
 102)
```

Above array is representing 102 columns of our dataset.

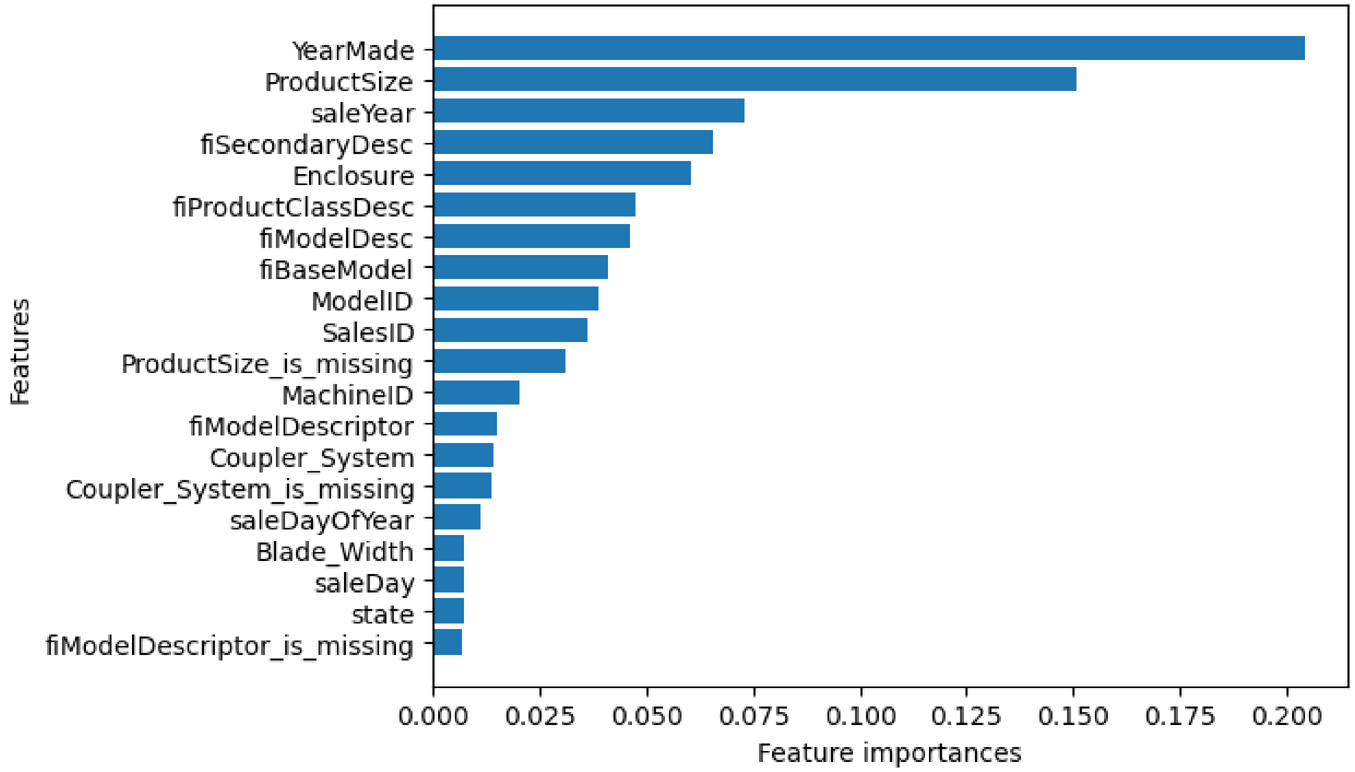
In [92]:

```
# Helper function for ploting feature importance
def plot_features(columns, importances, n=20):
    df = (pd.DataFrame({"features": columns,
                        "feature_importances": importances})
          .sort_values("feature_importances", ascending=False)
          .reset_index(drop=True))
```

```
    # Plot the DataFrame
    fig, ax = plt.subplots()
    ax.barh(df["features"][:n], df["feature_importances"][:20])
    ax.set_ylabel("Features")
    ax.set_xlabel("Feature importances")
    ax.invert_yaxis()
```

In [93]:

```
plot_features(X_train.columns, ideal_model.feature_importances_)
```



In [94]:

```
df["ProductSize"].value_counts()
```

Out[94]:

```
Medium            64342
Large / Medium    51297
Small             27057
Mini              25721
Large             21396
Compact            6280
Name: ProductSize, dtype: int64
```

In [95]:

```
df["Enclosure"].value_counts()
```

Out[95]:

```
OROPS                     177971
EROPS                     141769
EROPS w AC                 92601
EROPS AC                      18
NO ROPS                        3
None or Unspecified            2
Name: Enclosure, dtype: int64
```