🥅 Predicting the Sale Price of Bulldozers using Machine Learning 🚜 🍂

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [6]:
```
pd , np ,plt , sns
```

Out[6]:
```
(<module 'pandas' from 'D:\\Projects\\bulldozer price prediction\\env\\Lib
\\site-packages\\pandas\\__init__.py'>,
 <module 'numpy' from 'D:\\Projects\\bulldozer price prediction\\env\\Lib
\\site-packages\\numpy\\__init__.py'>,
 <module 'matplotlib.pyplot' from 'D:\\Projects\\bulldozer price prediction
\\env\\Lib\\site-packages\\matplotlib\\pyplot.py'>,
 <module 'seaborn' from 'D:\\Projects\\bulldozer price prediction\\env\\Lib
\\site-packages\\seaborn\\__init__.py'>)
```

In [7]:
```
df = pd.read_csv("data/TrainAndValid.csv" , low_memory = False)
```

In [8]:
```
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
```

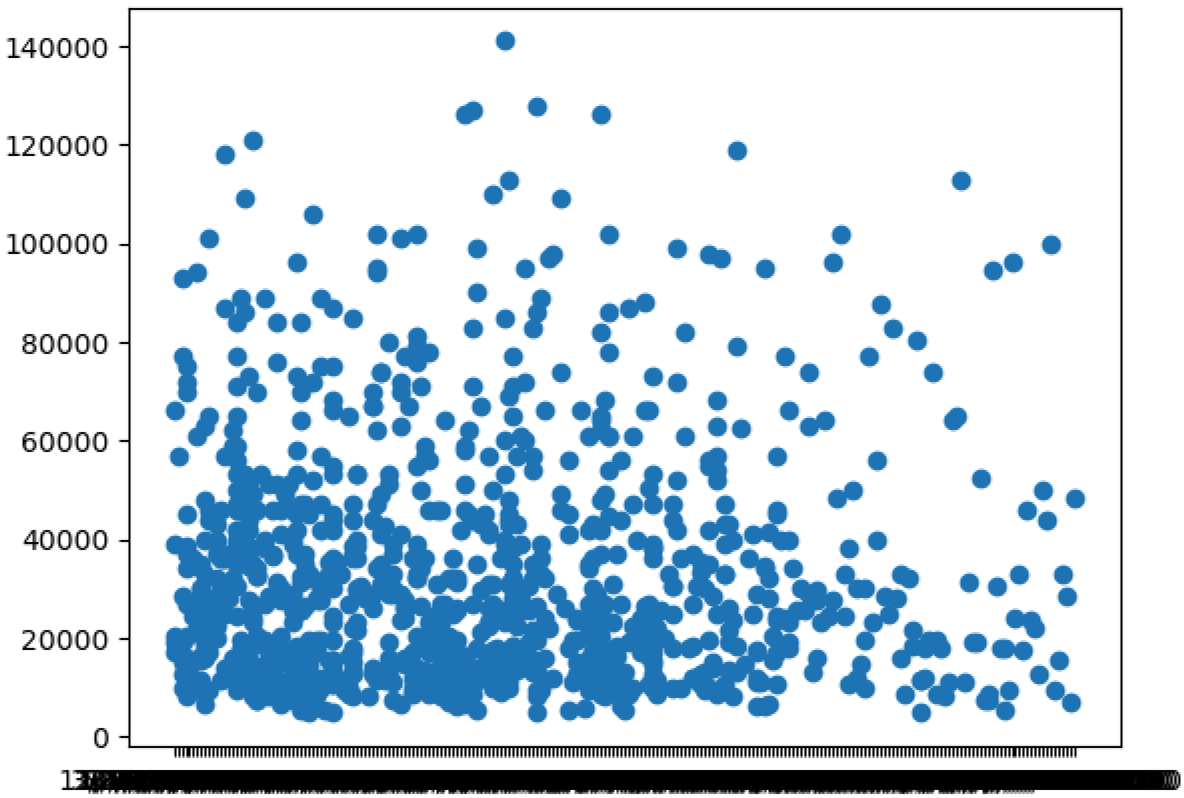
In [9]:
```
df.isna().sum()
```

Out[9]:

```
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 [10]:
```python
fig , ax = plt.subplots()
ax.scatter(df["saledate"][:1000] , df["SalePrice"][:1000])
```

Out[10]: <matplotlib.collections.PathCollection at 0x25f8ed3e3c0>

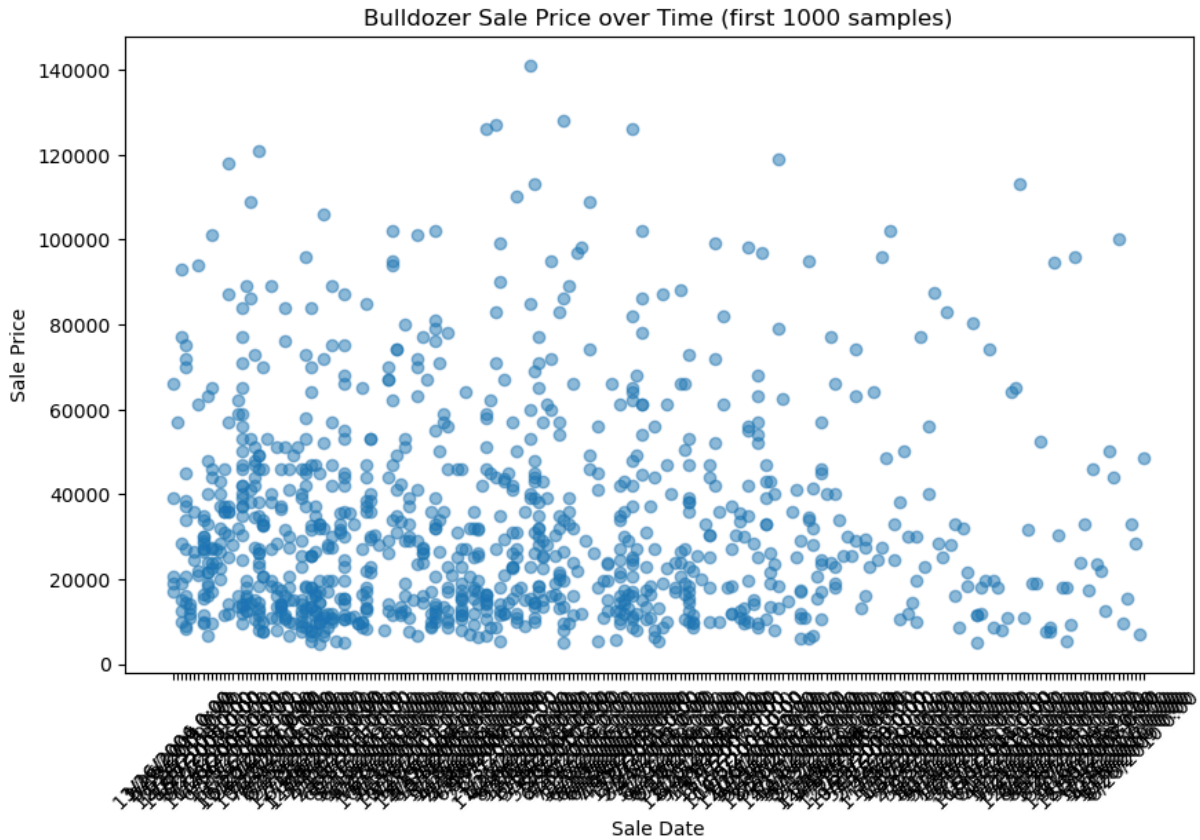


In [11]:
```python
fig, ax = plt.subplots(figsize=(10,6))

ax.scatter(df["saledate"][:1000], df["SalePrice"][:1000], alpha=0.5)

ax.set_xlabel("Sale Date")
ax.set_ylabel("Sale Price")
ax.set_title("Bulldozer Sale Price over Time (first 1000 samples)")

plt.xticks(rotation=45)  # rotate dates for readability
plt.show()
```

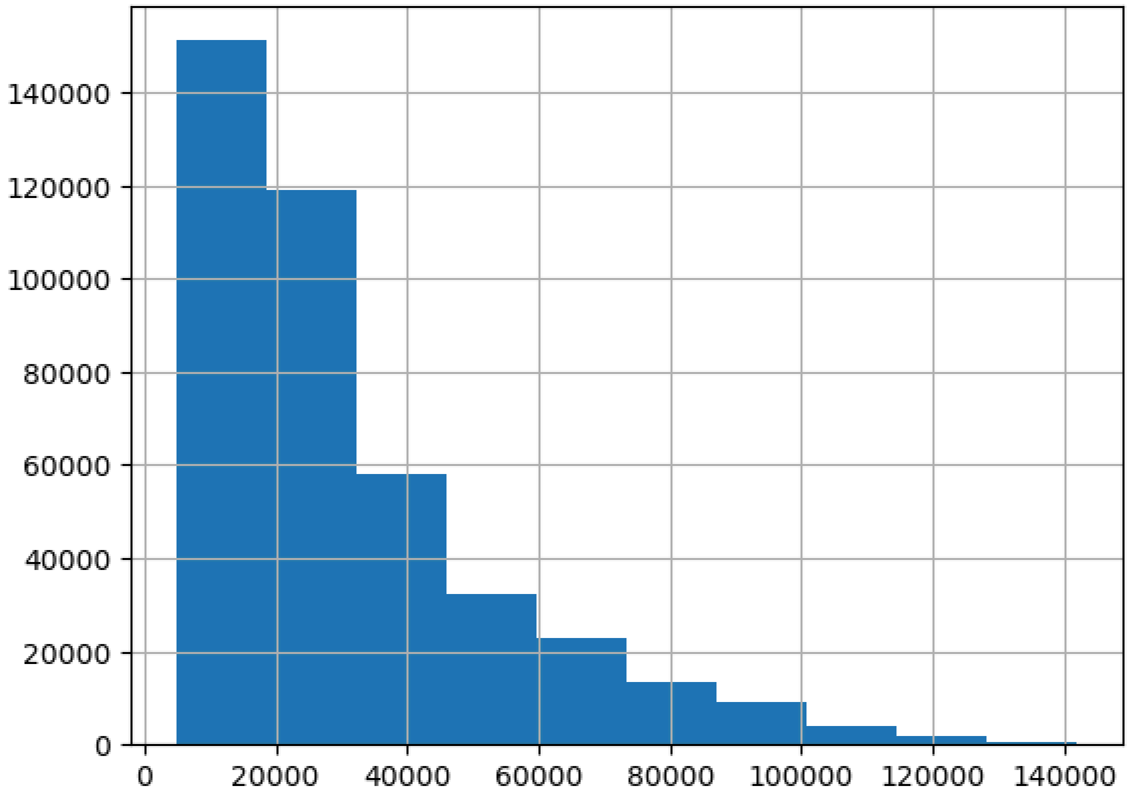


In [12]:
```
df.saledate
```

Out[12]:
```
0         11/16/2006 0:00
1          3/26/2004 0:00
2          2/26/2004 0:00
3          5/19/2011 0:00
4          7/23/2009 0:00
                ...
412693      3/7/2012 0:00
412694     1/28/2012 0:00
412695     1/28/2012 0:00
412696      3/7/2012 0:00
412697     1/28/2012 0:00
Name: saledate, Length: 412698, dtype: object
```

In [13]:
```
# Plot histogram
df["SalePrice"].hist()  # bins = number of intervals
```

Out[13]:
```
<Axes: >
```



Prasing Data

In [16]:
```
df = pd.read_csv("data/TrainAndValid.csv" , low_memory= False , parse_dates=
```

In [38]:
```
df
```

Out[38]:

	SalesID	SalePrice	MachineID	ModelID	datasource	auctioneerID	Y
0	1139246	66000.0	999089	3157	121	3.0	
1	1139248	57000.0	117657	77	121	3.0	
2	1139249	10000.0	434808	7009	121	3.0	
3	1139251	38500.0	1026470	332	121	3.0	
4	1139253	11000.0	1057373	17311	121	3.0	
...	...	...	...	...	...	...	
412693	6333344	10000.0	1919201	21435	149	2.0	
412694	6333345	10500.0	1882122	21436	149	2.0	
412695	6333347	12500.0	1944213	21435	149	2.0	
412696	6333348	10000.0	1794518	21435	149	2.0	
412697	6333349	13000.0	1944743	21436	149	2.0	

412698 rows × 53 columns

In [17]:
```
df.saledate[:1000]
```

Out[17]:
```
0      2006-11-16
1      2004-03-26
2      2004-02-26
3      2011-05-19
4      2009-07-23
          ...
995    2009-07-16
996    2007-06-14
997    2005-09-22
998    2005-07-28
999    2011-06-16
Name: saledate, Length: 1000, dtype: datetime64[ns]
```

In [44]:
```
df["saledate"]
```

Out[44]:
```
0         2006-11-16
1         2004-03-26
2         2004-02-26
3         2011-05-19
4         2009-07-23
             ...
412693   2012-03-07
412694   2012-01-28
412695   2012-01-28
412696   2012-03-07
412697   2012-01-28
Name: saledate, Length: 412698, dtype: datetime64[ns]
```

In [18]:
```
fig , ax = plt.subplots()
ax.scatter(df["saledate"][:1000]  , df["SalePrice"][:1000])
```

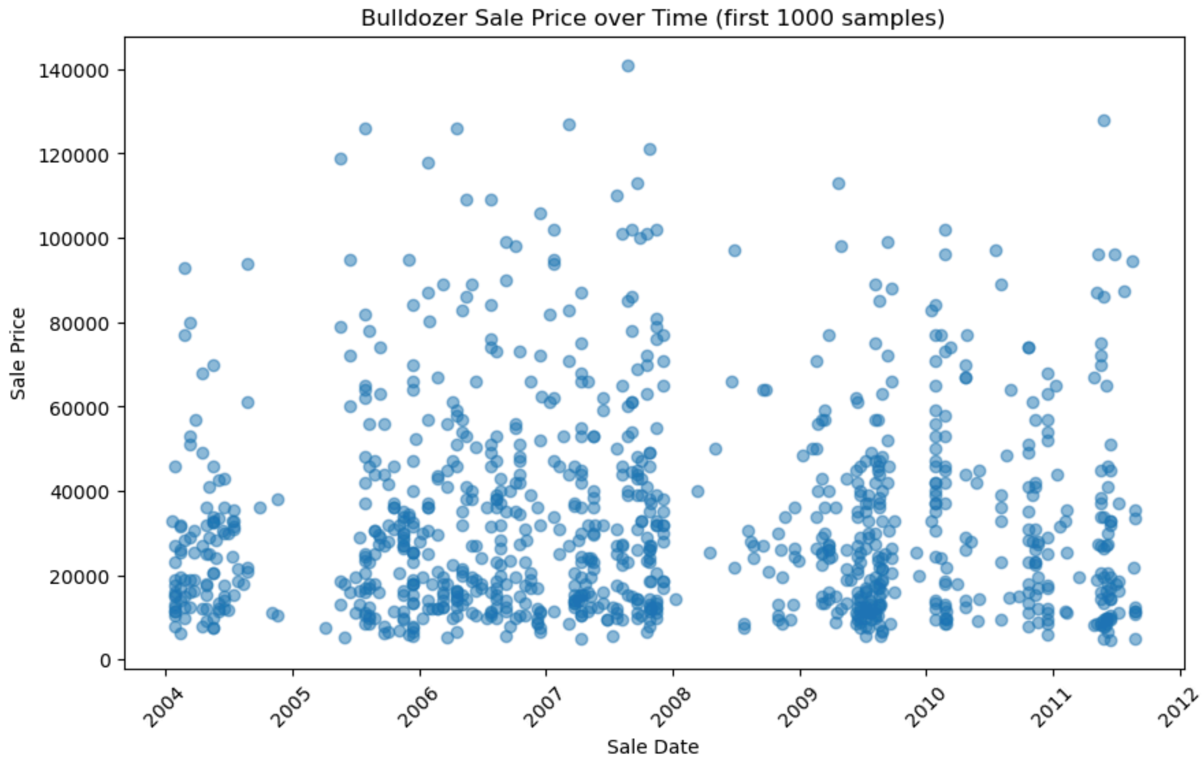
Out[18]: <matplotlib.collections.PathCollection at 0x25f922191d0>

In [19]:
```
fig, ax = plt.subplots(figsize=(10,6))

ax.scatter(df["saledate"][:1000], df["SalePrice"][:1000], alpha=0.5)

ax.set_xlabel("Sale Date")
ax.set_ylabel("Sale Price")
ax.set_title("Bulldozer Sale Price over Time (first 1000 samples)")

plt.xticks(rotation=45)  # rotate dates for readability
plt.show()
```



In [20]:
```
df.head().T
```

Out[20]:

	0	1	2	3
SalesID	1139246	1139248	1139249	1139251
SalePrice	66000.0	57000.0	10000.0	38500.0
MachineID	999089	117657	434808	1026470
ModelID	3157	77	7009	332
datasource	121	121	121	121
auctioneerID	3.0	3.0	3.0	3.0
YearMade	2004	1996	2001	2001
MachineHoursCurrentMeter	68.0	4640.0	2838.0	3486.0
UsageBand	Low	Low	High	High
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
fiModelDesc	521D	950FII	226	PC120-6E
fiBaseModel	521	950	226	PC120
fiSecondaryDesc	D	F	NaN	NaN
fiModelSeries	NaN	II	NaN	-6E
fiModelDescriptor	NaN	NaN	NaN	NaN
ProductSize	NaN	Medium	NaN	Small
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...
state	Alabama	North Carolina	New York	Texas
ProductGroup	WL	WL	SSL	TEX
ProductGroupDesc	Wheel Loader	Wheel Loader	Skid Steer Loaders	Track Excavators
Drive_System	NaN	NaN	NaN	NaN
Enclosure	EROPS w AC	EROPS w AC	OROPS	EROPS w AC
Forks	None or Unspecified	None or Unspecified	None or Unspecified	NaN
Pad_Type	NaN	NaN	NaN	NaN
Ride_Control	None or Unspecified	None or Unspecified	NaN	NaN
Stick	NaN	NaN	NaN	NaN
Transmission	NaN	NaN	NaN	NaN

	0	1	2	3
Turbocharged	NaN	NaN	NaN	NaN
Blade_Extension	NaN	NaN	NaN	NaN
Blade_Width	NaN	NaN	NaN	NaN
Enclosure_Type	NaN	NaN	NaN	NaN
Engine_Horsepower	NaN	NaN	NaN	NaN
Hydraulics	2 Valve	2 Valve	Auxiliary	2 Valve
Pushblock	NaN	NaN	NaN	NaN
Ripper	NaN	NaN	NaN	NaN
Scarifier	NaN	NaN	NaN	NaN
Tip_Control	NaN	NaN	NaN	NaN
Tire_Size	None or Unspecified	23.5	NaN	NaN
Coupler	None or Unspecified	None or Unspecified	None or Unspecified	None or Unspecified
Coupler_System	NaN	NaN	None or Unspecified	NaN
Grouser_Tracks	NaN	NaN	None or Unspecified	NaN
Hydraulics_Flow	NaN	NaN	Standard	NaN
Track_Type	NaN	NaN	NaN	NaN
Undercarriage_Pad_Width	NaN	NaN	NaN	NaN
Stick_Length	NaN	NaN	NaN	NaN
Thumb	NaN	NaN	NaN	NaN
Pattern_Changer	NaN	NaN	NaN	NaN
Grouser_Type	NaN	NaN	NaN	NaN
Backhoe_Mounting	NaN	NaN	NaN	NaN
Blade_Type	NaN	NaN	NaN	NaN
Travel_Controls	NaN	NaN	NaN	NaN
Differential_Type	Standard	Standard	NaN	NaN
Steering_Controls	Conventional	Conventional	NaN	NaN

Sort Dataframe by Saledate

In [23]:
```
df.sort_values(by=["saledate"] , inplace = True , ascending = True)
```

In [24]:
```
df["saledate"].head(50)
```

Out[24]:
```
205615   1989-01-17
233186   1989-01-31
142491   1989-01-31
115536   1989-01-31
92301    1989-01-31
115892   1989-01-31
134080   1989-01-31
92294    1989-01-31
31494    1989-01-31
140922   1989-01-31
66337    1989-01-31
92531    1989-01-31
82122    1989-01-31
92256    1989-01-31
145670   1989-01-31
92780    1989-01-31
238373   1989-01-31
127132   1989-01-31
115102   1989-01-31
32317    1989-01-31
238656   1989-01-31
52508    1989-01-31
127923   1989-01-31
127521   1989-01-31
152689   1989-01-31
82165    1989-01-31
78445    1989-01-31
62665    1989-01-31
113454   1989-01-31
113547   1989-01-31
28820    1989-01-31
168619   1989-01-31
115957   1989-01-31
205782   1989-01-31
114830   1989-01-31
127735   1989-01-31
78382    1989-01-31
127674   1989-01-31
28603    1989-01-31
78278    1989-01-31
231507   1989-01-31
169757   1989-01-31
92803    1989-01-31
75832    1989-01-31
88803    1989-01-31
75378    1989-01-31
169297   1989-01-31
280078   1989-01-31
140257   1989-01-31
128751   1989-01-31
Name: saledate, dtype: datetime64[ns]
```

Make a copy of orginal Data frame

In [25]:
```python
df_tmp = df.copy()
```

In [59]:
```python
df_tmp
```

Out[59]:

	SalesID	SalePrice	MachineID	ModelID	datasource	auctioneerID	Y
205615	1646770	9500.0	1126363	8434	132	18.0	
274835	1821514	14000.0	1194089	10150	132	99.0	
141296	1505138	50000.0	1473654	4139	132	99.0	
212552	1671174	16000.0	1327630	8591	132	99.0	
62755	1329056	22000.0	1336053	4089	132	99.0	
...	...	...	...	...	...	...	
410879	6302984	16000.0	1915521	5266	149	99.0	
412476	6324811	6000.0	1919104	19330	149	99.0	
411927	6313029	16000.0	1918416	17244	149	99.0	
407124	6266251	55000.0	509560	3357	149	99.0	
409203	6283635	34000.0	1869284	4701	149	99.0	

412698 rows × 53 columns

add Datetime parameter for "*SaleDate*" column

In [26]:
```python
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 [7]:
```python
df_tmp.head().T
```

Out[7]:

	205615	274835	141296	212552
SalesID	1646770	1821514	1505138	1671174
SalePrice	9500.0	14000.0	50000.0	16000.0
MachineID	1126363	1194089	1473654	1327630
ModelID	8434	10150	4139	8591
datasource	132	132	132	132
auctioneerID	18.0	99.0	99.0	99.0
YearMade	1974	1980	1978	1980
MachineHoursCurrentMeter	NaN	NaN	NaN	NaN
UsageBand	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
fiModelDesc	TD20	A66	D7G	A62
fiBaseModel	TD20	A66	D7	A62
fiSecondaryDesc	NaN	NaN	G	NaN
fiModelSeries	NaN	NaN	NaN	NaN
fiModelDescriptor	NaN	NaN	NaN	NaN
ProductSize	Medium	NaN	Large	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
state	Texas	Florida	Florida	Florida
ProductGroup	TTT	WL	TTT	WL
ProductGroupDesc	Track Type Tractors	Wheel Loader	Track Type Tractors	Wheel Loader
Drive_System	NaN	NaN	NaN	NaN
Enclosure	OROPS	OROPS	OROPS	EROPS
Forks	NaN	None or Unspecified	NaN	None or Unspecified
Pad_Type	NaN	NaN	NaN	NaN
Ride_Control	NaN	None or Unspecified	NaN	None or Unspecified
Stick	NaN	NaN	NaN	NaN
Transmission	Direct Drive	NaN	Standard	NaN

	205615	274835	141296	212552
Turbocharged	NaN	NaN	NaN	NaN
Blade_Extension	NaN	NaN	NaN	NaN
Blade_Width	NaN	NaN	NaN	NaN
Enclosure_Type	NaN	NaN	NaN	NaN
Engine_Horsepower	NaN	NaN	NaN	NaN
Hydraulics	2 Valve	2 Valve	2 Valve	2 Valve
Pushblock	NaN	NaN	NaN	NaN
Ripper	None or Unspecified	NaN	None or Unspecified	NaN
Scarifier	NaN	NaN	NaN	NaN
Tip_Control	NaN	NaN	NaN	NaN
Tire_Size	NaN	None or Unspecified	NaN	None or Unspecified
Coupler	NaN	None or Unspecified	NaN	None or Unspecified
Coupler_System	NaN	NaN	NaN	NaN
Grouser_Tracks	NaN	NaN	NaN	NaN
Hydraulics_Flow	NaN	NaN	NaN	NaN
Track_Type	NaN	NaN	NaN	NaN
Undercarriage_Pad_Width	NaN	NaN	NaN	NaN
Stick_Length	NaN	NaN	NaN	NaN
Thumb	NaN	NaN	NaN	NaN
Pattern_Changer	NaN	NaN	NaN	NaN
Grouser_Type	NaN	NaN	NaN	NaN
Backhoe_Mounting	None or Unspecified	NaN	None or Unspecified	NaN
Blade_Type	Straight	NaN	Straight	NaN
Travel_Controls	None or Unspecified	NaN	None or Unspecified	NaN
Differential_Type	NaN	Standard	NaN	Standard
Steering_Controls	NaN	Conventional	NaN	Conventional
saleyear	1989	1989	1989	1989
saleMonth	1	1	1	1
saleDay	17	31	31	31
saleDayoFWeek	1	1	1	1

	205615	274835	141296	212552
saleDayofyear	17	31	31	31

Now we going to remove sale date

In [27]:
```
df_tmp.drop("saledate" , axis = 1 , inplace = True)
```

In [28]:
```
df_tmp.columns
```

Out[28]:
```
Index(['SalesID', 'SalePrice', 'MachineID', 'ModelID', 'datasource',
       'auctioneerID', 'YearMade', 'MachineHoursCurrentMeter', 'UsageBand',
       'fiModelDesc', 'fiBaseModel', 'fiSecondaryDesc', 'fiModelSeries',
       'fiModelDescriptor', 'ProductSize', 'fiProductClassDesc', 'state',
       'ProductGroup', 'ProductGroupDesc', 'Drive_System', 'Enclosure',
       'Forks', 'Pad_Type', 'Ride_Control', 'Stick', 'Transmission',
       'Turbocharged', 'Blade_Extension', 'Blade_Width', 'Enclosure_Type',
       'Engine_Horsepower', 'Hydraulics', 'Pushblock', 'Ripper', 'Scarifie
r',
       'Tip_Control', 'Tire_Size', 'Coupler', 'Coupler_System',
       'Grouser_Tracks', 'Hydraulics_Flow', 'Track_Type',
       'Undercarriage_Pad_Width', 'Stick_Length', 'Thumb', 'Pattern_Change
r',
       'Grouser_Type', 'Backhoe_Mounting', 'Blade_Type', 'Travel_Controls',
       'Differential_Type', 'Steering_Controls', 'saleyear', 'saleMonth',
       'saleDay', 'saleDayoFWeek', 'saleDayofyear'],
      dtype='object')
```

In [29]:
```
# Check the values of different columns
df_tmp.state.value_counts()
```

```
Out[29]: state
Florida           67320
Texas             53110
California        29761
Washington        16222
Georgia           14633
Maryland          13322
Mississippi       13240
Ohio              12369
Illinois          11540
Colorado          11529
New Jersey        11156
North Carolina    10636
Tennessee         10298
Alabama           10292
Pennsylvania      10234
South Carolina     9951
Arizona            9364
New York           8639
Connecticut        8276
Minnesota          7885
Missouri           7178
Nevada             6932
Louisiana          6627
Kentucky           5351
Maine              5096
Indiana            4124
Arkansas           3933
New Mexico         3631
Utah               3046
Unspecified        2801
Wisconsin          2745
New Hampshire      2738
Virginia           2353
Idaho              2025
Oregon             1911
Michigan           1831
Wyoming            1672
Montana            1336
Iowa               1336
Oklahoma           1326
Nebraska            866
West Virginia       840
Kansas              667
Delaware            510
North Dakota        480
Alaska              430
Massachusetts       347
Vermont             300
South Dakota        244
Hawaii              118
Rhode Island         83
Puerto Rico          42
Washington DC         2
Name: count, dtype: int64
```

5. Modelling

In [30]:
```
from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor(n_jobs = -1 , random_state=42)
```

In [15]:
```
df
```

Out[15]:

	SalesID	SalePrice	MachineID	ModelID	datasource	auctioneerID	Y
205615	1646770	9500.0	1126363	8434	132	18.0	
274835	1821514	14000.0	1194089	10150	132	99.0	
141296	1505138	50000.0	1473654	4139	132	99.0	
212552	1671174	16000.0	1327630	8591	132	99.0	
62755	1329056	22000.0	1336053	4089	132	99.0	
...	...	...	...	...	...	...	
410879	6302984	16000.0	1915521	5266	149	99.0	
412476	6324811	6000.0	1919104	19330	149	99.0	
411927	6313029	16000.0	1918416	17244	149	99.0	
407124	6266251	55000.0	509560	3357	149	99.0	
409203	6283635	34000.0	1869284	4701	149	99.0	

412698 rows × 53 columns

In [31]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 412698 entries, 205615 to 409203
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  datetime64[ns]
 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: datetime64[ns](1), float64(3), int64(5), object(44)
memory usage: 170.0+ MB
```

In [32]:
```
df.isna().sum()
```

```
Out[32]: 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
```

Convert String to categories

In [27]:
```
df_tmp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 412698 entries, 205615 to 409203
Data columns (total 57 columns):
 #   Column                    Non-Null Count   Dtype  
---  ------                    --------------   -----  
 0   SalesID                   412698 non-null  int64  
 1   SalePrice                 412698 non-null  float64
 2   MachineID                 412698 non-null  int64  
 3   ModelID                   412698 non-null  int64  
 4   datasource                412698 non-null  int64  
 5   auctioneerID              392562 non-null  float64
 6   YearMade                  412698 non-null  int64  
 7   MachineHoursCurrentMeter  147504 non-null  float64
 8   UsageBand                 73670 non-null   object 
 9   fiModelDesc               412698 non-null  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  int32
 53  saleMonth                  412698 non-null  int32
 54  saleDay                    412698 non-null  int32
 55  saleDayoFWeek              412698 non-null  int32
 56  saleDayofyear              412698 non-null  int32
dtypes: float64(3), int32(5), int64(5), object(44)
memory usage: 174.7+ MB
```

In [33]:
```python
pd.api.types.is_string_dtype(df_tmp["UsageBand"])
```

Out[33]:
```
False
```

In [34]:
```python
for label , content in df_tmp.items():
    if pd.api.types.is_string_dtype(content):

        print(label)
    #if pd.api.types.is_string_dtype(df_tmp())
```

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

In [35]:
```python
for label ,content in df_tmp.items():
    if pd.api.types.is_object_dtype(content):
        df_tmp[label] = content.astype("category").cat.as_ordered()
```

In [36]:
```python
 df_tmp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 412698 entries, 205615 to 409203
Data columns (total 57 columns):
 #   Column                    Non-Null Count   Dtype   
---  ------                    --------------   -----   
 0   SalesID                   412698 non-null  int64   
 1   SalePrice                 412698 non-null  float64 
 2   MachineID                 412698 non-null  int64   
 3   ModelID                   412698 non-null  int64   
 4   datasource                412698 non-null  int64   
 5   auctioneerID              392562 non-null  float64 
 6   YearMade                  412698 non-null  int64   
 7   MachineHoursCurrentMeter  147504 non-null  float64 
 8   UsageBand                 73670 non-null   category
 9   fiModelDesc               412698 non-null  category
 10  fiBaseModel               412698 non-null  category
 11  fiSecondaryDesc           271971 non-null  category
 12  fiModelSeries             58667 non-null   category
 13  fiModelDescriptor         74816 non-null   category
 14  ProductSize               196093 non-null  category
 15  fiProductClassDesc        412698 non-null  category
 16  state                     412698 non-null  category
 17  ProductGroup              412698 non-null  category
 18  ProductGroupDesc          412698 non-null  category
 19  Drive_System              107087 non-null  category
 20  Enclosure                 412364 non-null  category
 21  Forks                     197715 non-null  category
 22  Pad_Type                  81096 non-null   category
 23  Ride_Control              152728 non-null  category
 24  Stick                     81096 non-null   category
 25  Transmission              188007 non-null  category
 26  Turbocharged              81096 non-null   category
 27  Blade_Extension           25983 non-null   category
 28  Blade_Width               25983 non-null   category
 29  Enclosure_Type            25983 non-null   category
 30  Engine_Horsepower         25983 non-null   category
 31  Hydraulics                330133 non-null  category
 32  Pushblock                 25983 non-null   category
 33  Ripper                    106945 non-null  category
 34  Scarifier                 25994 non-null   category
 35  Tip_Control               25983 non-null   category
 36  Tire_Size                 97638 non-null   category
 37  Coupler                   220679 non-null  category
 38  Coupler_System            44974 non-null   category
 39  Grouser_Tracks            44875 non-null   category
 40  Hydraulics_Flow           44875 non-null   category
 41  Track_Type                102193 non-null  category
 42  Undercarriage_Pad_Width   102916 non-null  category
 43  Stick_Length              102261 non-null  category
 44  Thumb                     102332 non-null  category
 45  Pattern_Changer           102261 non-null  category
 46  Grouser_Type              102193 non-null  category
 47  Backhoe_Mounting          80712 non-null   category
 48  Blade_Type                81875 non-null   category
 49  Travel_Controls           81877 non-null   category
 50  Differential_Type         71564 non-null   category
```

```
 51  Steering_Controls          71522 non-null   category
 52  saleyear                   412698 non-null  int32
 53  saleMonth                  412698 non-null  int32
 54  saleDay                    412698 non-null  int32
 55  saleDayoFWeek              412698 non-null  int32
 56  saleDayofyear              412698 non-null  int32
dtypes: category(44), float64(3), int32(5), int64(5)
memory usage: 55.4 MB
```

In [37]:
```
df_tmp.state.cat.codes
```

Out[37]:
```
205615    43
233186     8
142491     8
115536     8
92301      8
          ..
409901     4
405777     4
411889     4
411890     4
409203     4
Length: 412698, dtype: int8
```

Thanks for categories

FIll Missing values

Fill numerical missing values first

In [38]:
```
for label , content in df_tmp.items():
    if pd.api.types.is_numeric_dtype(content):
        print(df_tmp[label].isna().sum() , label)
```

```
0 SalesID
0 SalePrice
0 MachineID
0 ModelID
0 datasource
20136 auctioneerID
0 YearMade
265194 MachineHoursCurrentMeter
0 saleyear
0 saleMonth
0 saleDay
0 saleDayoFWeek
0 saleDayofyear
```

In [61]:
```
df_tmp.ModelID
```

```
Out[61]: 205615     8434
         274835    10150
         141296     4139
         212552     8591
         62755      4089
                   ...
         410879     5266
         412476    19330
         411927    17244
         407124     3357
         409203     4701
         Name: ModelID, Length: 412698, dtype: int64
```

```
In [39]: # 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 [40]: # Fill 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():
            print(label)
```

```
auctioneerID
MachineHoursCurrentMeter
```

```
In [41]: # Fill 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():
            print(label)
            df_tmp[label + "_is_missing"] = pd.isnull(content)
            df_tmp[label] = content.fillna(content.median())
```

```
auctioneerID
MachineHoursCurrentMeter
```

```
In [42]: # Fill numeric rows with the median
```

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

In [43]:
```python
df_tmp.auctioneerID_is_missing.value_counts()
```

Out[43]:
```
auctioneerID_is_missing
False    392562
True      20136
Name: count, dtype: int64
```

Filling and turning categorical variables into numbers

In [44]:
```python
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 [45]:

```
# Turn categorical variables into numbers

for label , content in df_tmp.items():
    if not pd.api.types.is_numeric_dtype(content):
        # Add binary column to indicate
        df_tmp[label+"_is_missing"] = pd.isnull(content)

        df_tmp[label] = pd.Categorical(content).codes+1
```

In [46]:
```
df_tmp.isna().sum()
```

Out[46]:
```
SalesID                           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
```

In [47]:
```
df.isna().sum()
```

```
Out[47]: 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 [48]:
```
df_tmp.head()
```

Out[48]:

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

5 rows × 103 columns

In [51]:
```
%%time

model = RandomForestRegressor(n_jobs = -1 , random_state=42)
model.fit(df_tmp.drop("SalePrice" , axis =1) ,df_tmp["SalePrice"])
```

```
CPU times: total: 31min 4s
Wall time: 3min 5s
```

Out[51]:

▾ RandomForestRegressor

▶ Parameters

In [52]:
```
%%time
model.score(df_tmp.drop("SalePrice" , axis =1) ,df_tmp["SalePrice"])
```

```
CPU times: total: 38.8 s
Wall time: 4.1 s
```

Out[52]: 0.9875966080326709

Spliting data into train/validation sets

In [53]:
```
df_val = df_tmp[df_tmp.saleyear == 2012]
df_train = df_tmp[df_tmp.saleyear != 2012]
```

In [55]:
```
len(df_train) , len(df_val)
```

Out[55]: (401125, 11573)

In [56]:
```
len(df_train) + len(df_val)
```

Out[56]: 412698

In [57]:
```
x_train , y_train = df_train.drop("SalePrice" , axis = 1) , df_train.SalePri
```

In [58]:
```
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[58]:
```
((401125, 102), (401125,), (11573, 102), (11573,))
```

Building and evaluation Funtion

In [59]:
```
# Create evaluation function

from sklearn.metrics import mean_squared_log_error , mean_absolute_error
from sklearn.metrics import r2_score


def rmsle(y_test,y_preds):
    """
    Calculate root mean squard log error
    """

    return np.sqrt(mean_squared_log_error(y_test , y_preds))

# Create function to evaluate model on a few different levels

def show_scores(model):
    train_preds = model.predict(x_train)
    val_preds = model.predict(x_valid)

    scores = {"Training MAE" : mean_absolute_error(y_train, train_preds),
             "Valid MAE" : mean_absolute_error(y_valid , val_preds),
              "Training RMSLE" : rmsle(y_train , train_preds),
              "Valid RMSLE" : rmsle(y_valid , val_preds) ,
              "Training R2" : r2_score(y_train , train_preds),
              "Valid R2" : r2_score(y_valid , val_preds)
             }

    return scores
```

Testing our model on subset

%%time

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

model.fit(x_train , y_train)

In [60]:
```
# CHnage max sample value

model = RandomForestRegressor(n_jobs=-1 , random_state=42 , max_samples = 10


model
```

Out[60]:

▾ RandomForestRegressor

▶ Parameters

In [61]:
```
x_train.shape[0]
```

Out[61]:
```
401125
```

In [62]:
```
%%time

model.fit(x_train , y_train)
```

```
CPU times: total: 59.1 s
Wall time: 6.07 s
```

Out[62]:

▾ RandomForestRegressor

▶ Parameters

In [63]:
```
show_scores(model)
```

Out[63]:
```
{'Training MAE': 5548.995840324088,
 'Valid MAE': 7179.6961392897265,
 'Training RMSLE': np.float64(0.25737726780537257),
 'Valid RMSLE': np.float64(0.29404344200903443),
 'Training R2': 0.8610738743845617,
 'Valid R2': 0.8320179198265637}
```

In [64]:
```
%%time

from sklearn.model_selection import RandomizedSearchCV

rf_grid = {"n_estimators"}
```

```
CPU times: total: 0 ns
Wall time: 31.2 µs
```

Train a model with best hyparameters

In [65]:
```
%%time

ideal_model = RandomForestRegressor(n_estimators=40 , min_samples_leaf=1, mi

ideal_model.fit(x_train , y_train)
```

```
CPU times: total: 5min 19s
Wall time: 32.5 s
```

Out[65]:

▾ RandomForestRegressor

▶ Parameters

In [92]:
```
show_scores(ideal_model)
```

Out[92]:
```
{'Training MAE': 2954.3860545002144,
 'Valid MAE': 5934.273509420097,
 'Training RMSLE': np.float64(0.14445948886340432),
 'Valid RMSLE': np.float64(0.24607663981354877),
 'Training R2': 0.9588911792610285,
 'Valid R2': 0.8824941147924968}
```

In [66]:
```
# Scores for ideal model(trained on all data)
show_scores(ideal_model)
```

Out[66]:
```
{'Training MAE': 2956.4887000110048,
 'Valid MAE': 5957.457704159022,
 'Training RMSLE': np.float64(0.14448478225577735),
 'Valid RMSLE': np.float64(0.24544663828024385),
 'Training R2': 0.9587936671021189,
 'Valid R2': 0.8820814857765893}
```

Make predictions on test data

In [67]:
```
# Import the test datatest

test_df =pd.read_csv("Data/Test.csv" , low_memory=False , parse_dates=["sale
```

In [68]:
```
len(test_df.columns)
```

Out[68]:
```
52
```

Make predictions

''' test_preds = ideal_model.predict(test_df) '''

> This has error because this is not same as the training set Now we are going to preprocess the data and make sure its same as ...

In [69]:
```
def preprocess_data(test_df):
    """
    Perform transformation on df and returns transformed df.
    """'''
    test_df["saleyear"]       = test_df.saledate.dt.year
    test_df["saleMonth"]      = test_df.saledate.dt.month
    test_df["saleDay"]        =test_df.saledate.dt.day
    test_df["saleDayoFWeek"]  = test_df.saledate.dt.dayofweek
    test_df["saleDayofyear"]  =test_df.saledate.dt.dayofyear

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

    # Fill the numeric rows with median
    for label , content in test_df.items():
        if pd.api.types.is_numeric_dtype(content):
```

```python
            if pd.isnull(content).sum():
                print(label)
                test_df[label + "_is_missing"] = pd.isnull(content)
                test_df[label] = content.fillna(content.median())




    #Filled categorical missing data and turned in numbers


        if not pd.api.types.is_numeric_dtype(content):
        # Add binary column to indicate
            test_df[label+"_is_missing"] = pd.isnull(content)

            test_df[label] = pd.Categorical(content).codes+1



    return test_df
```

In [70]:

```python
test_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12457 entries, 0 to 12456
Data columns (total 52 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   SalesID                   12457 non-null  int64
 1   MachineID                 12457 non-null  int64
 2   ModelID                   12457 non-null  int64
 3   datasource                12457 non-null  int64
 4   auctioneerID              12457 non-null  int64
 5   YearMade                  12457 non-null  int64
 6   MachineHoursCurrentMeter  2129 non-null   float64
 7   UsageBand                 1834 non-null   object
 8   saledate                  12457 non-null  datetime64[ns]
 9   fiModelDesc               12457 non-null  object
 10  fiBaseModel               12457 non-null  object
 11  fiSecondaryDesc           8482 non-null   object
 12  fiModelSeries             2006 non-null   object
 13  fiModelDescriptor         3024 non-null   object
 14  ProductSize               6048 non-null   object
 15  fiProductClassDesc        12457 non-null  object
 16  state                     12457 non-null  object
 17  ProductGroup              12457 non-null  object
 18  ProductGroupDesc          12457 non-null  object
 19  Drive_System              2759 non-null   object
 20  Enclosure                 12455 non-null  object
 21  Forks                     6308 non-null   object
 22  Pad_Type                  2108 non-null   object
 23  Ride_Control              4241 non-null   object
 24  Stick                     2108 non-null   object
 25  Transmission              4818 non-null   object
 26  Turbocharged              2108 non-null   object
 27  Blade_Extension           651 non-null    object
 28  Blade_Width               651 non-null    object
 29  Enclosure_Type            651 non-null    object
 30  Engine_Horsepower         651 non-null    object
 31  Hydraulics                10315 non-null  object
 32  Pushblock                 651 non-null    object
 33  Ripper                    2704 non-null   object
 34  Scarifier                 651 non-null    object
 35  Tip_Control               651 non-null    object
 36  Tire_Size                 2778 non-null   object
 37  Coupler                   7601 non-null   object
 38  Coupler_System            2066 non-null   object
 39  Grouser_Tracks            2066 non-null   object
 40  Hydraulics_Flow           2066 non-null   object
 41  Track_Type                3394 non-null   object
 42  Undercarriage_Pad_Width   3398 non-null   object
 43  Stick_Length              3394 non-null   object
 44  Thumb                     3395 non-null   object
 45  Pattern_Changer           3394 non-null   object
 46  Grouser_Type              3394 non-null   object
 47  Backhoe_Mounting          2051 non-null   object
 48  Blade_Type                2058 non-null   object
 49  Travel_Controls           2058 non-null   object
 50  Differential_Type         2129 non-null   object
```

```
 51  Steering_Controls         2129 non-null   object
dtypes: datetime64[ns](1), float64(1), int64(6), object(44)
memory usage: 4.9+ MB
```

In [71]:
```
preprocess_data(test_df)
```

```
MachineHoursCurrentMeter
```

Out[71]:

	SalesID	MachineID	ModelID	datasource	auctioneerID	YearMade	M
0	1227829	1006309	3168	121	3	1999	
1	1227844	1022817	7271	121	3	1000	
2	1227847	1031560	22805	121	3	2004	
3	1227848	56204	1269	121	3	2006	
4	1227863	1053887	22312	121	3	2005	
...	...	...	...	...	...	...	
12452	6643171	2558317	21450	149	2	2008	
12453	6643173	2558332	21434	149	2	2005	
12454	6643184	2558342	21437	149	2	1000	
12455	6643186	2558343	21437	149	2	2006	
12456	6643196	2558346	21446	149	2	2008	

12457 rows × 98 columns

In [85]:
```
test_df
```

Out[85]:

	SalesID	MachineID	ModelID	datasource	auctioneerID	YearMade	M
0	1227829	1006309	3168	121	False	1999	
1	1227844	1022817	7271	121	False	1000	
2	1227847	1031560	22805	121	False	2004	
3	1227848	56204	1269	121	False	2006	
4	1227863	1053887	22312	121	False	2005	
...	...	...	...	...	...	...	
12452	6643171	2558317	21450	149	False	2008	
12453	6643173	2558332	21434	149	False	2005	
12454	6643184	2558342	21437	149	False	1000	
12455	6643186	2558343	21437	149	False	2006	
12456	6643196	2558346	21446	149	False	2008	

12457 rows × 102 columns

In [72]:
```
test_df.isna().sum()
```

Out[72]:
```
SalesID                         0
MachineID                       0
ModelID                         0
datasource                      0
auctioneerID                    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: 98, dtype: int64
```

In [79]:
```
test_df.head()
```

Out[79]:

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

5 rows × 102 columns

In [58]:
```
x_train.head()
```

Out[58]:

	SalesID	MachineID	ModelID	datasource	auctioneerID	YearMade	I
205615	1646770	1126363	8434	132	18.0	1974	
274835	1821514	1194089	10150	132	99.0	1980	
141296	1505138	1473654	4139	132	99.0	1978	
212552	1671174	1327630	8591	132	99.0	1980	
62755	1329056	1336053	4089	132	99.0	1984	

5 rows × 102 columns

In [74]:
```
# GOING TO FIND THE COLLUMN DIFFER

set(x_train.columns) - set(test_df.columns)
```

Out[74]:
```
{'auctioneerID_is_missing',
 'saleDay',
 'saleDayoFWeek',
 'saleDayofyear',
 'saleMonth',
 'saleyear'}
```

In [75]:
```
## adjust df_test to have aunctioneer id

test_df["auctioneerID_is_missing"]  = False
```

In [76]:
```
len(x_valid) , len(test_df)
```

Out[76]:
```
(11573, 12457)
```

In [76]:
```
test_df
```

Out[76]:

	SalesID	MachineID	ModelID	datasource	auctioneerID	YearMade	M
0	1227829	1006309	3168	121	False	1999	
1	1227844	1022817	7271	121	False	1000	
2	1227847	1031560	22805	121	False	2004	
3	1227848	56204	1269	121	False	2006	
4	1227863	1053887	22312	121	False	2005	
...	...	...	...	...	...	...	
12452	6643171	2558317	21450	149	False	2008	
12453	6643173	2558332	21434	149	False	2005	
12454	6643184	2558342	21437	149	False	1000	
12455	6643186	2558343	21437	149	False	2006	
12456	6643196	2558346	21446	149	False	2008	

12457 rows × 102 columns

Make predictions on test data

test_preds = ideal_model.predict(test_df)

In [77]:
```
test_df = preprocess_data(test_df)
```

In [90]:
```
test_df
```

Out[90]:

	SalesID	MachineID	ModelID	datasource	auctioneerID	YearMade	M
0	1227829	1006309	3168	121	False	1999	
1	1227844	1022817	7271	121	False	1000	
2	1227847	1031560	22805	121	False	2004	
3	1227848	56204	1269	121	False	2006	
4	1227863	1053887	22312	121	False	2005	
...	...	...	...	...	...	...	
12452	6643171	2558317	21450	149	False	2008	
12453	6643173	2558332	21434	149	False	2005	
12454	6643184	2558342	21437	149	False	1000	
12455	6643186	2558343	21437	149	False	2006	
12456	6643196	2558346	21446	149	False	2008	

12457 rows × 102 columns

In [78]:

```
# Preprocess test data
test_df_proc = preprocess_data(test_df)

# Match the columns to training data
test_df_proc = test_df_proc.reindex(columns=x_train.columns, fill_value=0)

# Now predict
test_preds = ideal_model.predict(test_df_proc)
```

In [79]:

```
len(test_preds)
```

Out[79]: 12457

In [99]:

```
test_df_proc
```

Out[99]:

	SalesID	MachineID	ModelID	datasource	auctioneerID	YearMade	M
0	1227829	1006309	3168	121	False	1999	
1	1227844	1022817	7271	121	False	1000	
2	1227847	1031560	22805	121	False	2004	
3	1227848	56204	1269	121	False	2006	
4	1227863	1053887	22312	121	False	2005	
...	...	...	...	...	...	...	
12452	6643171	2558317	21450	149	False	2008	
12453	6643173	2558332	21434	149	False	2005	
12454	6643184	2558342	21437	149	False	1000	
12455	6643186	2558343	21437	149	False	2006	
12456	6643196	2558346	21446	149	False	2008	

12457 rows × 102 columns

In [80]:

```
# Format predictions into the same format

df_preds = pd.DataFrame()

df_preds["SalesID"] = test_df["SalesID"]
df_preds["SalesPrice"] = test_preds
```

In [105…

```
df_preds
```

Out[105…

	SalesID	SalesPrice
0	1227829	17623.337125
1	1227844	14566.296570
2	1227847	46662.254410
3	1227848	71305.266295
4	1227863	61762.999424
...	...	...
12452	6643171	40469.885910
12453	6643173	12196.277617
12454	6643184	11964.850733
12455	6643186	16342.165338
12456	6643196	27119.990440

12457 rows × 2 columns

In [81]:
```
df_preds.to_csv("test_prediction.csv" , index = False)
```

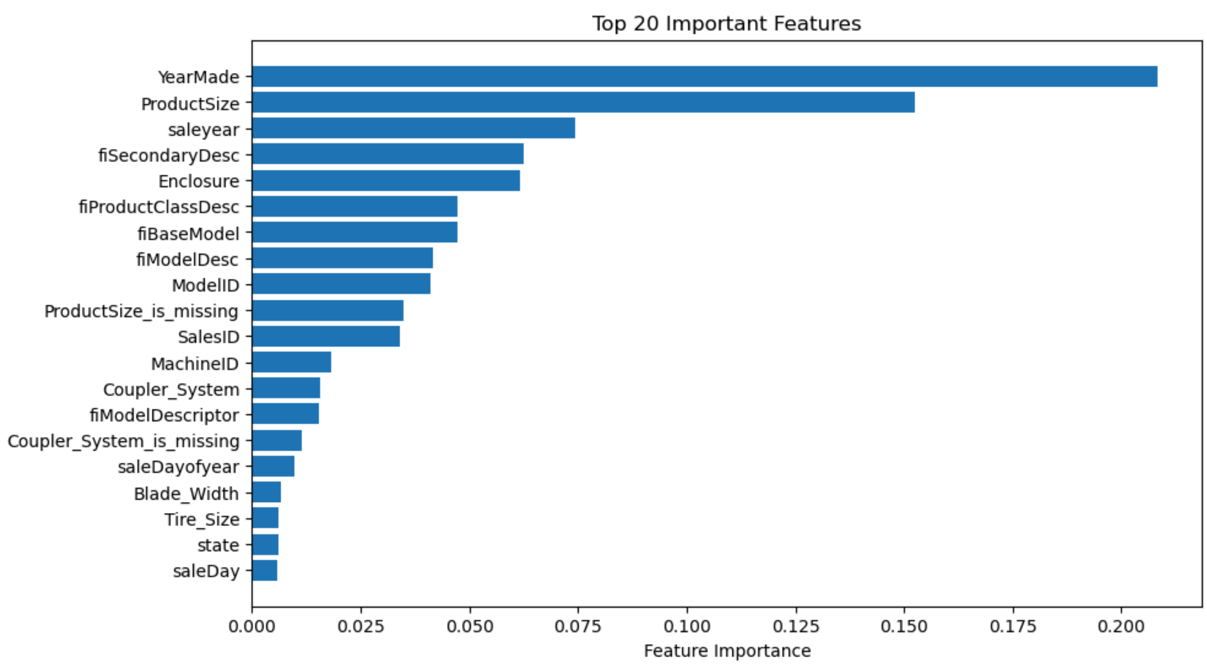
In [82]:
```
len(ideal_model.feature_importances_)
```

Out[82]: 102

In [83]:
```
# Helper function for plotting feature importance
def plot_features(columns, importances, n=20):
    # Create a DataFrame with features and their importance
    df = (
        pd.DataFrame({
            "feature": columns,
            "importance": importances
        })
        .sort_values("importance", ascending=False)
        .reset_index(drop=True)
    )

    # Plot
    fig, ax = plt.subplots(figsize=(10, 6))
    ax.barh(df["feature"][:n][::-1], df["importance"][:n][::-1])  # horizont
    ax.set_xlabel("Feature Importance")
    ax.set_title(f"Top {n} Important Features")
    plt.show()
```

In [118…
```
plot_features(x_train.columns , ideal_model.feature_importances_)
```



In [119…
```
x_train.head()
```

Out[119…

	SalesID	MachineID	ModelID	datasource	auctioneerID	YearMade	I
205615	1646770	1126363	8434	132	18.0	1974	
274835	1821514	1194089	10150	132	99.0	1980	
141296	1505138	1473654	4139	132	99.0	1978	
212552	1671174	1327630	8591	132	99.0	1980	
62755	1329056	1336053	4089	132	99.0	1984	

5 rows × 102 columns

In [ ]: