In [1]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
```

In [2]:

```
df=pd.read_csv('car data.csv')
```

In [3]:

```
df.head()
```

Out[3]:

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmiss
0	ritz	2014	3.35	5.59	27000	Petrol	Dealer	Mar
1	sx4	2013	4.75	9.54	43000	Diesel	Dealer	Mar
2	ciaz	2017	7.25	9.85	6900	Petrol	Dealer	Mar
3	wagon r	2011	2.85	4.15	5200	Petrol	Dealer	Mar
4	swift	2014	4.60	6.87	42450	Diesel	Dealer	Mar

In [4]:

```
df.tail()
```

Out[4]:

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmi
296	city	2016	9.50	11.6	33988	Diesel	Dealer	M
297	brio	2015	4.00	5.9	60000	Petrol	Dealer	M
298	city	2009	3.35	11.0	87934	Petrol	Dealer	M
299	city	2017	11.50	12.5	9000	Diesel	Dealer	M
300	brio	2016	5.30	5.9	5464	Petrol	Dealer	M

In [5]:
```
df.shape
```

Out[5]:
```
(301, 9)
```

In [6]:
```
print(df['Seller_Type'].unique())
print(df['Transmission'].unique())
print(df['Owner'].unique())
print(df['Fuel_Type'].unique())
```

```
['Dealer' 'Individual']
['Manual' 'Automatic']
[0 1 3]
['Petrol' 'Diesel' 'CNG']
```

In [7]:
```
df.isnull().sum()
```

Out[7]:
```
Car_Name         0
Year             0
Selling_Price    0
Present_Price    0
Kms_Driven       0
Fuel_Type        0
Seller_Type      0
Transmission     0
Owner            0
dtype: int64
```

In [8]:
```
df.describe()
```

Out[8]:

	Year	Selling_Price	Present_Price	Kms_Driven	Owner
count	301.000000	301.000000	301.000000	301.000000	301.000000
mean	2013.627907	4.661296	7.628472	36947.205980	0.043189
std	2.891554	5.082812	8.644115	38886.883882	0.247915
min	2003.000000	0.100000	0.320000	500.000000	0.000000
25%	2012.000000	0.900000	1.200000	15000.000000	0.000000

	Year	Selling_Price	Present_Price	Kms_Driven	Owner
50%	2014.000000	3.600000	6.400000	32000.000000	0.000000
75%	2016.000000	6.000000	9.900000	48767.000000	0.000000
max	2018.000000	35.000000	92.600000	500000.000000	3.000000

In [9]:
```
df.columns
```

Out[9]:
```
Index(['Car_Name', 'Year', 'Selling_Price', 'Present_Price', 'Kms_Drive
n',
       'Fuel_Type', 'Seller_Type', 'Transmission', 'Owner'],
      dtype='object')
```

In [10]:
```
final_dataset=df[['Year', 'Selling_Price', 'Present_Price', 'Kms_Drive
n',
       'Fuel_Type', 'Seller_Type', 'Transmission', 'Owner']]
```

In [11]:
```
final_dataset.head()
```

Out[11]:

	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0
1	2013	4.75	9.54	43000	Diesel	Dealer	Manual	0
2	2017	7.25	9.85	6900	Petrol	Dealer	Manual	0
3	2011	2.85	4.15	5200	Petrol	Dealer	Manual	0
4	2014	4.60	6.87	42450	Diesel	Dealer	Manual	0

In [12]:
```
final_dataset['current_year']=2020
```

In [13]:
```
final_dataset.head()
```

Out[13]:

	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner

	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0
1	2013	4.75	9.54	43000	Diesel	Dealer	Manual	0
2	2017	7.25	9.85	6900	Petrol	Dealer	Manual	0
3	2011	2.85	4.15	5200	Petrol	Dealer	Manual	0
4	2014	4.60	6.87	42450	Diesel	Dealer	Manual	0

In [14]:

```
final_dataset['number_years']=final_dataset['current_year']-final_dataset['Year']
```

In [15]:

```
final_dataset.head()
```

Out[15]:

	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0
1	2013	4.75	9.54	43000	Diesel	Dealer	Manual	0
2	2017	7.25	9.85	6900	Petrol	Dealer	Manual	0
3	2011	2.85	4.15	5200	Petrol	Dealer	Manual	0
4	2014	4.60	6.87	42450	Diesel	Dealer	Manual	0

In [16]:

```
final_dataset.drop(['current_year'],axis=1,inplace=True)
```

In [17]:

```
final_dataset.head()
```

Out[17]:

	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0
1	2013	4.75	9.54	43000	Diesel	Dealer	Manual	0

	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
2	2017	7.25	9.85	6900	Petrol	Dealer	Manual	0
3	2011	2.85	4.15	5200	Petrol	Dealer	Manual	0
4	2014	4.60	6.87	42450	Diesel	Dealer	Manual	0

```
In [18]: final_dataset.drop(['Year'],axis=1,inplace=True)
```

```
In [19]: final_dataset.head()
```

Out[19]:

	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner	numb
0	3.35	5.59	27000	Petrol	Dealer	Manual	0	
1	4.75	9.54	43000	Diesel	Dealer	Manual	0	
2	7.25	9.85	6900	Petrol	Dealer	Manual	0	
3	2.85	4.15	5200	Petrol	Dealer	Manual	0	
4	4.60	6.87	42450	Diesel	Dealer	Manual	0	

```
In [20]: final_dataset=pd.get_dummies(final_dataset,drop_first=True)
```

```
In [21]: final_dataset.head()
```

Out[21]:

	Selling_Price	Present_Price	Kms_Driven	Owner	number_years	Fuel_Type_Diesel	Fuel_Type_
0	3.35	5.59	27000	0	6	0	
1	4.75	9.54	43000	0	7	1	
2	7.25	9.85	6900	0	3	0	
3	2.85	4.15	5200	0	9	0	
4	4.60	6.87	42450	0	6	1	

In [22]:

```
final_dataset.corr()
```

Out[22]:

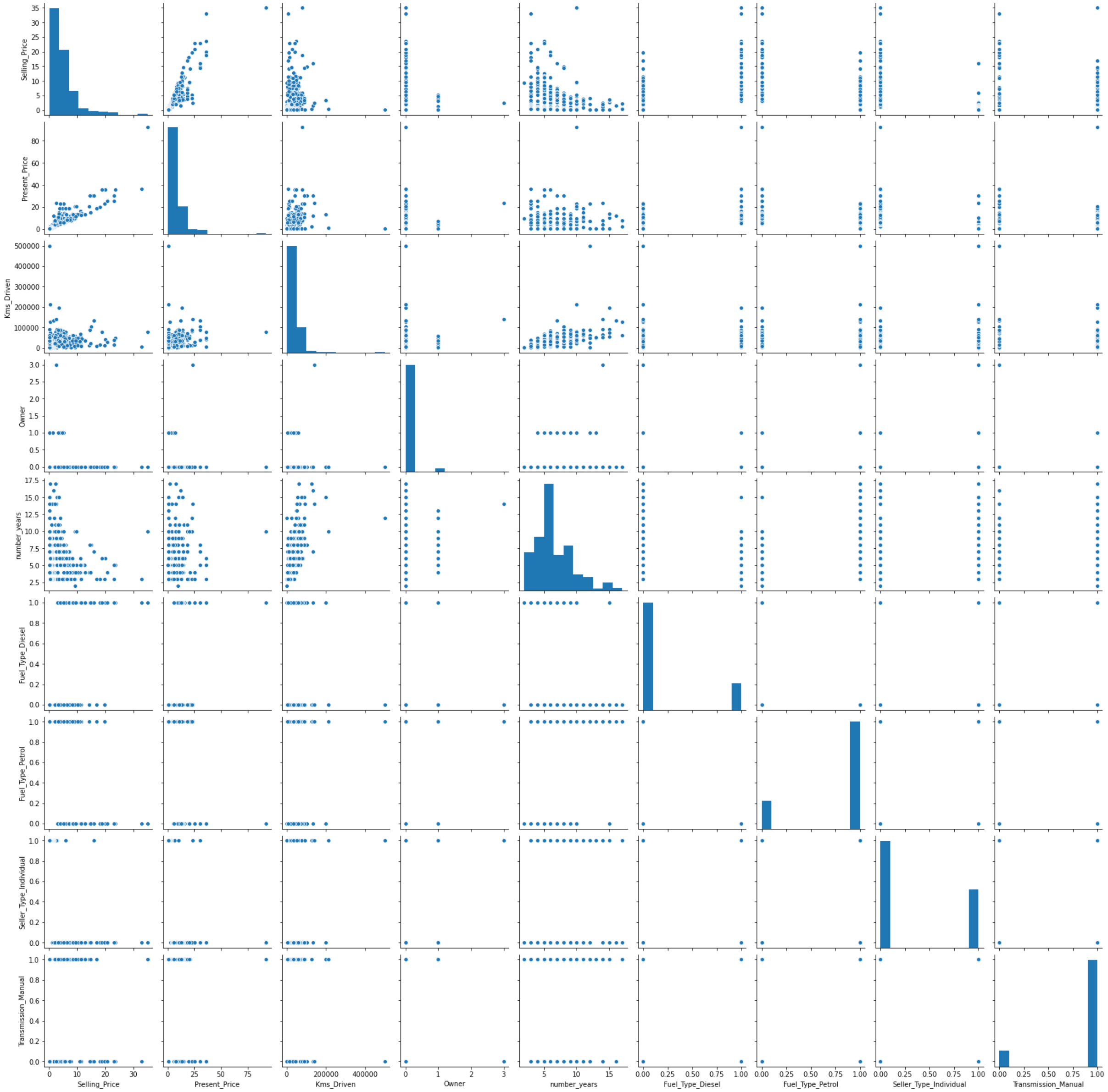
	Selling_Price	Present_Price	Kms_Driven	Owner	number_years	Fuel_T
Selling_Price	1.000000	0.878983	0.029187	-0.088344	-0.236141	
Present_Price	0.878983	1.000000	0.203647	0.008057	0.047584	
Kms_Driven	0.029187	0.203647	1.000000	0.089216	0.524342	
Owner	-0.088344	0.008057	0.089216	1.000000	0.182104	
number_years	-0.236141	0.047584	0.524342	0.182104	1.000000	
Fuel_Type_Diesel	0.552339	0.473306	0.172515	-0.053469	-0.064315	
Fuel_Type_Petrol	-0.540571	-0.465244	-0.172874	0.055687	0.059959	
Seller_Type_Individual	-0.550724	-0.512030	-0.101419	0.124269	0.039896	
Transmission_Manual	-0.367128	-0.348715	-0.162510	-0.050316	-0.000394	

In [23]:

```
sns.pairplot(final_dataset)
```

Out[23]:

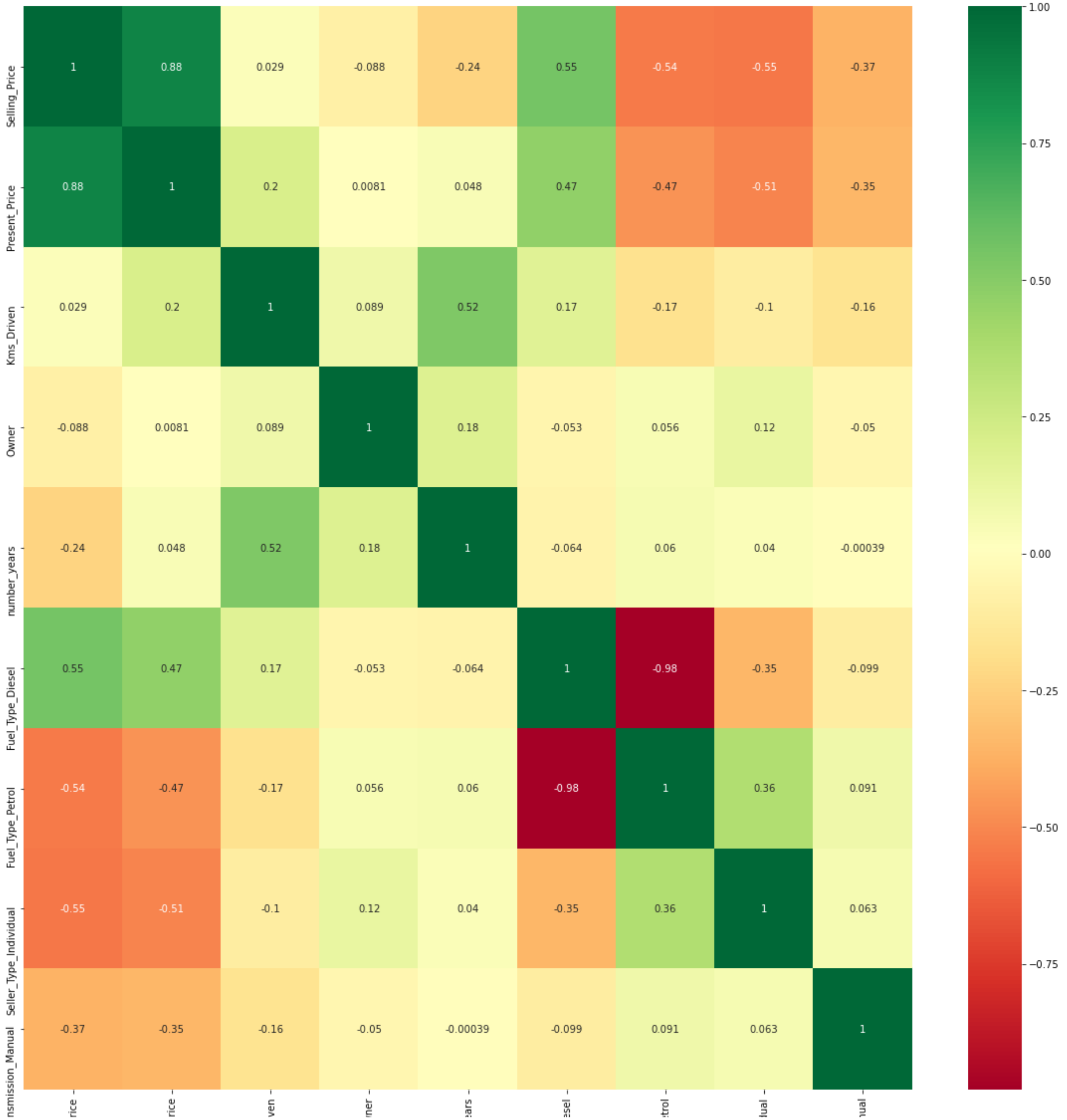
```
<seaborn.axisgrid.PairGrid at 0x19aafafea90>
```



In [24]:

```
cormat=final_dataset.corr()
top_corr_features=cormat.index
plt.figure(figsize=(20,20))
```

```
g=sns.heatmap(final_dataset[top_corr_features].corr(),annot=True,cmap=
"RdYlGn")
```



In [25]:

```
final_dataset.head()
```

Out[25]:

	Selling_Price	Present_Price	Kms_Driven	Owner	number_years	Fuel_Type_Diesel	Fuel_Type_
0	3.35	5.59	27000	0	6	0	
1	4.75	9.54	43000	0	7	1	
2	7.25	9.85	6900	0	3	0	
3	2.85	4.15	5200	0	9	0	
4	4.60	6.87	42450	0	6	1	

In [26]:

```
x=final_dataset.iloc[:,1:]
y=final_dataset.iloc[:,0]
```

In [27]:

```
x.head()
```

Out[27]:

	Present_Price	Kms_Driven	Owner	number_years	Fuel_Type_Diesel	Fuel_Type_Petrol	Seller_
0	5.59	27000	0	6	0	1	
1	9.54	43000	0	7	1	0	
2	9.85	6900	0	3	0	1	
3	4.15	5200	0	9	0	1	
4	6.87	42450	0	6	1	0	

In [28]:

```
y.head()
```

Out[28]:

```
0    3.35
1    4.75
```

```
2    7.25
3    2.85
4    4.60
Name: Selling_Price, dtype: float64
```

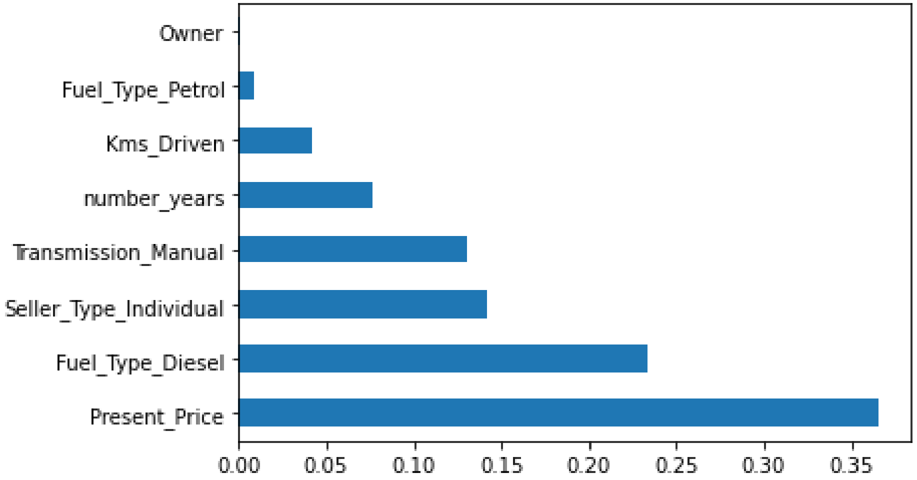
In [29]:
```
from sklearn.ensemble import ExtraTreesRegressor
model=ExtraTreesRegressor()
model.fit(x,y)
```

Out[29]:
```
ExtraTreesRegressor()
```

In [30]:
```
print(model.feature_importances_)
```

```
[0.36563119 0.04251827 0.00057315 0.07699896 0.23392282 0.00922645
 0.14134753 0.12978164]
```

In [31]:
```
feat_importances = pd.Series(model.feature_importances_, index=x.columns)
feat_importances.nlargest(8).plot(kind='barh')
plt.show()
```



In [32]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

In [33]:
```
x_train.shape
```

Out[33]:
```
(240, 8)
```

In [34]:
```
x_test.shape
```

Out[34]:
```
(61, 8)
```

In [35]:
```
y_train.shape
```

Out[35]:
```
(240,)
```

In [36]:
```
y_test.shape
```

Out[36]:
```
(61,)
```

In [37]:
```
from sklearn.ensemble import RandomForestRegressor
rf_random=RandomForestRegressor()
from sklearn.model_selection import RandomizedSearchCV
```

In [38]:
```
n_estimators=[int(x) for x in np.linspace(start=100,stop=1200,num=12)]
max_features=['auto','sqrt']
max_depth=[int(x) for x in np.linspace(5,30,num=6)]
min_samples_split=[2,5,10,15,100]
min_samples_leaf=[1,2,5,10]
```

In [39]:
```
random_grid= {'n_estimators':n_estimators,
              'max_features':max_features,
              'max_depth':max_depth,
              'min_samples_split':min_samples_split,
              'min_samples_leaf':min_samples_leaf}
print(random_grid)
```

```
{'n_estimators': [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 11
00, 1200], 'max_features': ['auto', 'sqrt'], 'max_depth': [5, 10, 15, 2
0, 25, 30], 'min_samples_split': [2, 5, 10, 15, 100], 'min_samples_lea
f': [1, 2, 5, 10]}
```

In [40]:
```
rf=RandomForestRegressor()
```

In [41]:
```
rf_random=RandomizedSearchCV(estimator=rf,param_distributions=random_gr
id,scoring='neg_mean_squared_error',n_iter=10,cv=5,verbose=2,random_sta
te=42,n_jobs=1)
```

In [42]:
```
rf_random.fit(x_train,y_train)
```

```
Fitting 5 folds for each of 10 candidates, totalling 50 fits
[CV] n_estimators=900, min_samples_split=5, min_samples_leaf=5, max_fea
tures=sqrt, max_depth=10
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent
workers.
```

```
[CV]  n_estimators=900, min_samples_split=5, min_samples_leaf=5, max_fe
atures=sqrt, max_depth=10, total=   1.6s
[CV] n_estimators=900, min_samples_split=5, min_samples_leaf=5, max_fea
tures=sqrt, max_depth=10
```

```
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    1.5s remaining:
    0.0s
```

```
[CV]  n_estimators=900, min_samples_split=5, min_samples_leaf=5, max_fe
atures=sqrt, max_depth=10, total=   1.4s
[CV] n_estimators=900, min_samples_split=5, min_samples_leaf=5, max_fea
tures=sqrt, max_depth=10
[CV]  n_estimators=900, min_samples_split=5, min_samples_leaf=5, max_fe
atures=sqrt, max_depth=10, total=   1.4s
[CV] n_estimators=900, min_samples_split=5, min_samples_leaf=5, max_fea
tures=sqrt, max_depth=10
[CV]  n_estimators=900, min_samples_split=5, min_samples_leaf=5, max_fe
atures=sqrt, max_depth=10, total=   1.6s
[CV] n_estimators=900, min_samples_split=5, min_samples_leaf=5, max_fea
tures=sqrt, max_depth=10
[CV]  n_estimators=900, min_samples_split=5, min_samples_leaf=5, max_fe
```

```
atures=sqrt, max_depth=10, total=   1.6s
[CV] n_estimators=1100, min_samples_split=10, min_samples_leaf=2, max_f
eatures=sqrt, max_depth=15
[CV]  n_estimators=1100, min_samples_split=10, min_samples_leaf=2, max_
features=sqrt, max_depth=15, total=   2.1s
[CV] n_estimators=1100, min_samples_split=10, min_samples_leaf=2, max_f
eatures=sqrt, max_depth=15
[CV]  n_estimators=1100, min_samples_split=10, min_samples_leaf=2, max_
features=sqrt, max_depth=15, total=   1.7s
[CV] n_estimators=1100, min_samples_split=10, min_samples_leaf=2, max_f
eatures=sqrt, max_depth=15
[CV]  n_estimators=1100, min_samples_split=10, min_samples_leaf=2, max_
features=sqrt, max_depth=15, total=   1.7s
[CV] n_estimators=1100, min_samples_split=10, min_samples_leaf=2, max_f
eatures=sqrt, max_depth=15
[CV]  n_estimators=1100, min_samples_split=10, min_samples_leaf=2, max_
features=sqrt, max_depth=15, total=   1.7s
[CV] n_estimators=1100, min_samples_split=10, min_samples_leaf=2, max_f
eatures=sqrt, max_depth=15
[CV]  n_estimators=1100, min_samples_split=10, min_samples_leaf=2, max_
features=sqrt, max_depth=15, total=   1.8s
[CV] n_estimators=300, min_samples_split=100, min_samples_leaf=5, max_f
eatures=auto, max_depth=15
[CV]  n_estimators=300, min_samples_split=100, min_samples_leaf=5, max_
features=auto, max_depth=15, total=   0.5s
[CV] n_estimators=300, min_samples_split=100, min_samples_leaf=5, max_f
eatures=auto, max_depth=15
[CV]  n_estimators=300, min_samples_split=100, min_samples_leaf=5, max_
features=auto, max_depth=15, total=   0.5s
[CV] n_estimators=300, min_samples_split=100, min_samples_leaf=5, max_f
eatures=auto, max_depth=15
[CV]  n_estimators=300, min_samples_split=100, min_samples_leaf=5, max_
features=auto, max_depth=15, total=   0.5s
[CV] n_estimators=300, min_samples_split=100, min_samples_leaf=5, max_f
eatures=auto, max_depth=15
[CV]  n_estimators=300, min_samples_split=100, min_samples_leaf=5, max_
features=auto, max_depth=15, total=   0.5s
[CV] n_estimators=300, min_samples_split=100, min_samples_leaf=5, max_f
eatures=auto, max_depth=15
```

```
[CV]  n_estimators=300, min_samples_split=100, min_samples_leaf=5, max_
features=auto, max_depth=15, total=   0.5s
[CV] n_estimators=400, min_samples_split=5, min_samples_leaf=5, max_fea
tures=auto, max_depth=15
[CV]  n_estimators=400, min_samples_split=5, min_samples_leaf=5, max_fe
atures=auto, max_depth=15, total=   0.7s
[CV] n_estimators=400, min_samples_split=5, min_samples_leaf=5, max_fea
tures=auto, max_depth=15
[CV]  n_estimators=400, min_samples_split=5, min_samples_leaf=5, max_fe
atures=auto, max_depth=15, total=   0.7s
[CV] n_estimators=400, min_samples_split=5, min_samples_leaf=5, max_fea
tures=auto, max_depth=15
[CV]  n_estimators=400, min_samples_split=5, min_samples_leaf=5, max_fe
atures=auto, max_depth=15, total=   0.7s
[CV] n_estimators=400, min_samples_split=5, min_samples_leaf=5, max_fea
tures=auto, max_depth=15
[CV]  n_estimators=400, min_samples_split=5, min_samples_leaf=5, max_fe
atures=auto, max_depth=15, total=   0.8s
[CV] n_estimators=400, min_samples_split=5, min_samples_leaf=5, max_fea
tures=auto, max_depth=15
[CV]  n_estimators=400, min_samples_split=5, min_samples_leaf=5, max_fe
atures=auto, max_depth=15, total=   0.7s
[CV] n_estimators=700, min_samples_split=5, min_samples_leaf=10, max_fe
atures=auto, max_depth=20
[CV]  n_estimators=700, min_samples_split=5, min_samples_leaf=10, max_f
eatures=auto, max_depth=20, total=   1.2s
[CV] n_estimators=700, min_samples_split=5, min_samples_leaf=10, max_fe
atures=auto, max_depth=20
[CV]  n_estimators=700, min_samples_split=5, min_samples_leaf=10, max_f
eatures=auto, max_depth=20, total=   1.2s
[CV] n_estimators=700, min_samples_split=5, min_samples_leaf=10, max_fe
atures=auto, max_depth=20
[CV]  n_estimators=700, min_samples_split=5, min_samples_leaf=10, max_f
eatures=auto, max_depth=20, total=   1.1s
[CV] n_estimators=700, min_samples_split=5, min_samples_leaf=10, max_fe
atures=auto, max_depth=20
[CV]  n_estimators=700, min_samples_split=5, min_samples_leaf=10, max_f
eatures=auto, max_depth=20, total=   1.1s
[CV] n_estimators=700, min_samples_split=5, min_samples_leaf=10, max_fe
```

```
atures=auto, max_depth=20
[CV]  n_estimators=700, min_samples_split=5, min_samples_leaf=10, max_f
eatures=auto, max_depth=20, total=   1.1s
[CV] n_estimators=1000, min_samples_split=2, min_samples_leaf=1, max_fe
atures=sqrt, max_depth=25
[CV]  n_estimators=1000, min_samples_split=2, min_samples_leaf=1, max_f
eatures=sqrt, max_depth=25, total=   1.7s
[CV] n_estimators=1000, min_samples_split=2, min_samples_leaf=1, max_fe
atures=sqrt, max_depth=25
[CV]  n_estimators=1000, min_samples_split=2, min_samples_leaf=1, max_f
eatures=sqrt, max_depth=25, total=   1.9s
[CV] n_estimators=1000, min_samples_split=2, min_samples_leaf=1, max_fe
atures=sqrt, max_depth=25
[CV]  n_estimators=1000, min_samples_split=2, min_samples_leaf=1, max_f
eatures=sqrt, max_depth=25, total=   1.7s
[CV] n_estimators=1000, min_samples_split=2, min_samples_leaf=1, max_fe
atures=sqrt, max_depth=25
[CV]  n_estimators=1000, min_samples_split=2, min_samples_leaf=1, max_f
eatures=sqrt, max_depth=25, total=   1.7s
[CV] n_estimators=1000, min_samples_split=2, min_samples_leaf=1, max_fe
atures=sqrt, max_depth=25
[CV]  n_estimators=1000, min_samples_split=2, min_samples_leaf=1, max_f
eatures=sqrt, max_depth=25, total=   1.8s
[CV] n_estimators=1100, min_samples_split=15, min_samples_leaf=10, max_
features=sqrt, max_depth=5
[CV]  n_estimators=1100, min_samples_split=15, min_samples_leaf=10, max
_features=sqrt, max_depth=5, total=   3.7s
[CV] n_estimators=1100, min_samples_split=15, min_samples_leaf=10, max_
features=sqrt, max_depth=5
[CV]  n_estimators=1100, min_samples_split=15, min_samples_leaf=10, max
_features=sqrt, max_depth=5, total=   2.3s
[CV] n_estimators=1100, min_samples_split=15, min_samples_leaf=10, max_
features=sqrt, max_depth=5
[CV]  n_estimators=1100, min_samples_split=15, min_samples_leaf=10, max
_features=sqrt, max_depth=5, total=   1.7s
[CV] n_estimators=1100, min_samples_split=15, min_samples_leaf=10, max_
features=sqrt, max_depth=5
[CV]  n_estimators=1100, min_samples_split=15, min_samples_leaf=10, max
_features=sqrt, max_depth=5, total=   1.8s
```

```
[CV] n_estimators=1100, min_samples_split=15, min_samples_leaf=10, max_
features=sqrt, max_depth=5
[CV]  n_estimators=1100, min_samples_split=15, min_samples_leaf=10, max
_features=sqrt, max_depth=5, total=   1.7s
[CV] n_estimators=300, min_samples_split=15, min_samples_leaf=1, max_fe
atures=sqrt, max_depth=15
[CV]  n_estimators=300, min_samples_split=15, min_samples_leaf=1, max_f
eatures=sqrt, max_depth=15, total=   0.5s
[CV] n_estimators=300, min_samples_split=15, min_samples_leaf=1, max_fe
atures=sqrt, max_depth=15
[CV]  n_estimators=300, min_samples_split=15, min_samples_leaf=1, max_f
eatures=sqrt, max_depth=15, total=   0.5s
[CV] n_estimators=300, min_samples_split=15, min_samples_leaf=1, max_fe
atures=sqrt, max_depth=15
[CV]  n_estimators=300, min_samples_split=15, min_samples_leaf=1, max_f
eatures=sqrt, max_depth=15, total=   0.5s
[CV] n_estimators=300, min_samples_split=15, min_samples_leaf=1, max_fe
atures=sqrt, max_depth=15
[CV]  n_estimators=300, min_samples_split=15, min_samples_leaf=1, max_f
eatures=sqrt, max_depth=15, total=   0.5s
[CV] n_estimators=300, min_samples_split=15, min_samples_leaf=1, max_fe
atures=sqrt, max_depth=15
[CV]  n_estimators=300, min_samples_split=15, min_samples_leaf=1, max_f
eatures=sqrt, max_depth=15, total=   0.5s
[CV] n_estimators=700, min_samples_split=10, min_samples_leaf=2, max_fe
atures=sqrt, max_depth=5
[CV]  n_estimators=700, min_samples_split=10, min_samples_leaf=2, max_f
eatures=sqrt, max_depth=5, total=   1.1s
[CV] n_estimators=700, min_samples_split=10, min_samples_leaf=2, max_fe
atures=sqrt, max_depth=5
[CV]  n_estimators=700, min_samples_split=10, min_samples_leaf=2, max_f
eatures=sqrt, max_depth=5, total=   1.1s
[CV] n_estimators=700, min_samples_split=10, min_samples_leaf=2, max_fe
atures=sqrt, max_depth=5
[CV]  n_estimators=700, min_samples_split=10, min_samples_leaf=2, max_f
eatures=sqrt, max_depth=5, total=   1.1s
[CV] n_estimators=700, min_samples_split=10, min_samples_leaf=2, max_fe
atures=sqrt, max_depth=5
[CV]  n_estimators=700, min_samples_split=10, min_samples_leaf=2, max_f
```

```
eatures=sqrt, max_depth=5, total=   1.1s
[CV] n_estimators=700, min_samples_split=10, min_samples_leaf=2, max_fe
atures=sqrt, max_depth=5
[CV]  n_estimators=700, min_samples_split=10, min_samples_leaf=2, max_f
eatures=sqrt, max_depth=5, total=   1.1s
[CV] n_estimators=700, min_samples_split=15, min_samples_leaf=1, max_fe
atures=auto, max_depth=20
[CV]  n_estimators=700, min_samples_split=15, min_samples_leaf=1, max_f
eatures=auto, max_depth=20, total=   1.2s
[CV] n_estimators=700, min_samples_split=15, min_samples_leaf=1, max_fe
atures=auto, max_depth=20
[CV]  n_estimators=700, min_samples_split=15, min_samples_leaf=1, max_f
eatures=auto, max_depth=20, total=   1.2s
[CV] n_estimators=700, min_samples_split=15, min_samples_leaf=1, max_fe
atures=auto, max_depth=20
[CV]  n_estimators=700, min_samples_split=15, min_samples_leaf=1, max_f
eatures=auto, max_depth=20, total=   1.3s
[CV] n_estimators=700, min_samples_split=15, min_samples_leaf=1, max_fe
atures=auto, max_depth=20
[CV]  n_estimators=700, min_samples_split=15, min_samples_leaf=1, max_f
eatures=auto, max_depth=20, total=   1.2s
[CV] n_estimators=700, min_samples_split=15, min_samples_leaf=1, max_fe
atures=auto, max_depth=20
[CV]  n_estimators=700, min_samples_split=15, min_samples_leaf=1, max_f
eatures=auto, max_depth=20, total=   1.2s
```

```
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:  1.0min finished
```

Out[42]:

```
RandomizedSearchCV(cv=5, estimator=RandomForestRegressor(), n_jobs=1,
                   param_distributions={'max_depth': [5, 10, 15, 20, 2
5, 30],
                                        'max_features': ['auto', 'sqr
t'],
                                        'min_samples_leaf': [1, 2, 5, 1
0],
                                        'min_samples_split': [2, 5, 10,
15,
                                                              100],
                                        'n_estimators': [100, 200, 300,
400,
                                                        500, 600, 700,
800,
                                                        900, 1000, 110
0,
                                                        1200]},
                   random_state=42, scoring='neg_mean_squared_error',
                   verbose=2)
```

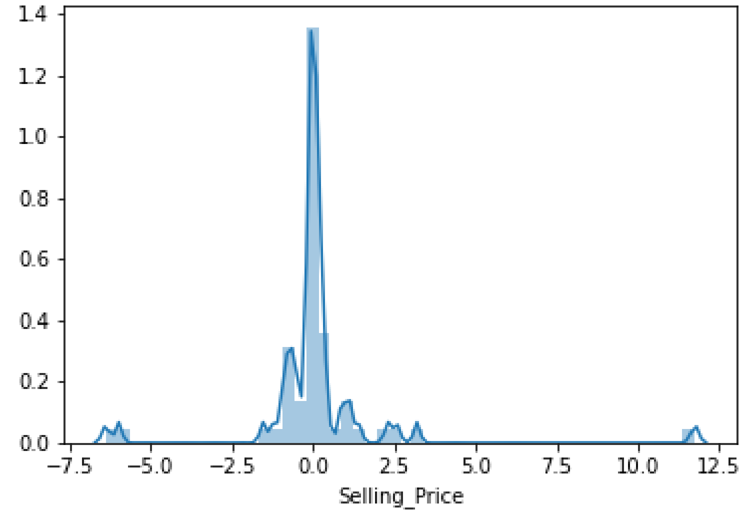
In [43]:
```
predictions=rf_random.predict(x_test)
```

In [44]:
```
predictions
```

Out[44]:
```
array([ 0.26837913,  0.4593147 , 10.52024716,  0.42189643,  0.61896569,
        0.248002  ,  0.41563852,  0.36068559,  0.35680646, 21.2496862 ,
        5.76803736,  1.32785092,  6.46464565,  8.67635595, 20.88769462,
        0.63689375,  0.33204653,  0.44124467,  5.92127311,  0.34891941,
        5.34631997,  4.93134494,  7.21821687,  7.29890374,  4.99671485,
        5.65613885,  0.67917511,  0.40244972,  2.67960329,  3.93751455,
        0.30174501,  2.87700728,  2.90473065,  5.61704894,  5.57064828,
        1.15281181,  1.3347594 ,  9.20620511,  0.67917511,  0.40034976,
        7.40357828,  6.56686081,  0.4178153 ,  0.32935377,  2.85711079,
        7.36201056,  5.63262151,  4.50609382,  7.19810348,  1.11892398,
        1.13665229,  2.62645471,  5.65460883,  0.44767348,  1.77917411,
        0.58966935,  0.29113723,  6.62309514, 20.88769462,  5.67094208,
       19.80242093])
```

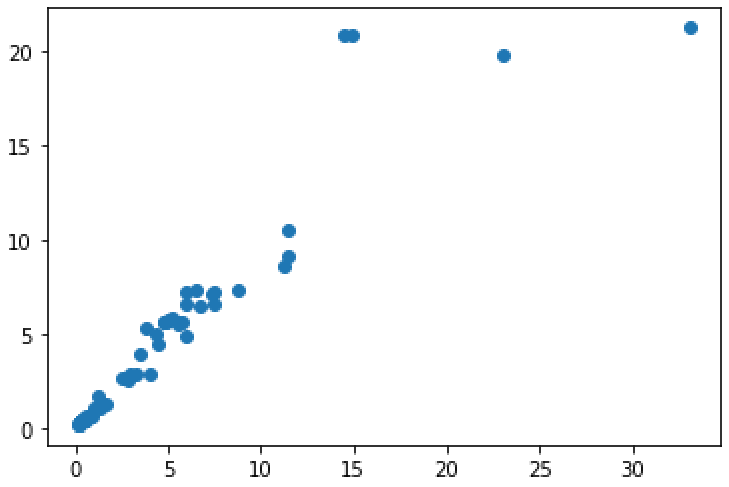
In [45]:
```
sns.distplot(y_test-predictions)
```

Out[45]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x19ab54b45b0>
```



```
In [47]: plt.scatter(y_test,predictions)
```

```
Out[47]: <matplotlib.collections.PathCollection at 0x19ab41d1190>
```



```
In [48]:
```

```
In [ ]:
```