In [1]:
```
import pandas as pd
import numpy as np
import statsmodels.api as sm
from collections import Counter



df = pd.read_csv('Train_pass3_10cols.csv', parse_dates=True)
df.head(10)
```

```
/anaconda3/lib/python3.6/site-packages/statsmodels/compat/pandas.py:56: FutureWarning: The pandas.core.datetools modul
e is deprecated and will be removed in a future version. Please use the pandas.tseries module instead.
  from pandas.core import datetools
```

Out[1]:

	SalesID	SalePrice	MachineID	ModelID	datasource	auctioneerID	YearMade	saledate	ProductGroup	Enclosure
0	1263845	11000	1542938	3208	132	5.0	1919	4/18/98 0:00	BL	OROPS
1	1271235	8500	1393999	6633	132	15.0	1919	4/13/11 0:00	SSL	OROPS
2	1284397	20000	1226376	3178	132	15.0	1919	10/12/11 0:00	BL	OROPS
3	1285083	20000	1346350	3178	132	15.0	1919	10/12/11 0:00	BL	OROPS
4	1288729	18000	1525079	3238	132	15.0	1919	4/13/11 0:00	TTT	OROPS
5	1362939	15500	1227772	7267	132	6.0	1919	3/2/95 0:00	WL	EROPS
6	1363002	17000	1387642	7267	132	2.0	1919	2/7/00 0:00	WL	EROPS
7	1368428	8500	1283255	4146	132	7.0	1919	3/7/07 0:00	TTT	OROPS
8	1369229	10500	1429950	4146	132	2.0	1919	9/25/99 0:00	TTT	OROPS
9	1369230	16000	1479540	4146	132	24.0	1919	8/12/99 0:00	TTT	OROPS

In [2]:
```
df.info()
#df.reset_index()
#df['AC'] = df[df['Enclosure']]
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 362940 entries, 0 to 362939
Data columns (total 10 columns):
SalesID         362940 non-null int64
SalePrice       362940 non-null int64
MachineID       362940 non-null int64
ModelID         362940 non-null int64
datasource      362940 non-null int64
auctioneerID    345366 non-null float64
YearMade        362940 non-null int64
saledate        362940 non-null object
ProductGroup    362940 non-null object
Enclosure       362685 non-null object
dtypes: float64(1), int64(6), object(3)
memory usage: 27.7+ MB
```

In [4]:
```
#df['YearSold'] = df['DateSold']

df['saledate'] = pd.to_datetime(df['saledate'])

df['saleyear'] = df['saledate'].dt.year
df['salemonth'] = df['saledate'].dt.month
#df.drop(columns=['YearSold'],inplace = True)

df.head()
```

Out[4]:

	SalesID	SalePrice	MachineID	ModelID	datasource	auctioneerID	YearMade	saledate	ProductGroup	Enclosure	saleyear	salemonth
0	1263845	11000	1542938	3208	132	5.0	1919	1998-04-18	BL	OROPS	1998	4
1	1271235	8500	1393999	6633	132	15.0	1919	2011-04-13	SSL	OROPS	2011	4
2	1284397	20000	1226376	3178	132	15.0	1919	2011-10-12	BL	OROPS	2011	10
3	1285083	20000	1346350	3178	132	15.0	1919	2011-10-12	BL	OROPS	2011	10
4	1288729	18000	1525079	3238	132	15.0	1919	2011-04-13	TTT	OROPS	2011	4

In [164]:
```
#df.dropna(inplace=True)
df.info()
Counter(df.ProductGroup)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400856 entries, 0 to 400855
Data columns (total 13 columns):
SalesID         400856 non-null int64
SalePrice       400856 non-null int64
MachineID       400856 non-null int64
ModelID         400856 non-null int64
saledate        400856 non-null object
fiModelDesc     400856 non-null object
fiBaseModel     400856 non-null object
state           400856 non-null object
ProductGroup    400856 non-null object
saleyear        400856 non-null int64
salemonth       400856 non-null int64
Enclosure       400856 non-null int64
AC              400856 non-null int64
dtypes: int64(8), object(5)
memory usage: 39.8+ MB
```

Out[164]:
```
Counter({'BL': 79408,
         'MG': 25489,
         'SSL': 43488,
         'TEX': 101055,
         'TTT': 80404,
         'WL': 71012})
```

In [5]:
```
df.groupby('ProductGroup').mean()
```

Out[5]:

ProductGroup	SalesID	SalePrice	MachineID	ModelID	datasource	auctioneerID	YearMade	saleyear	salemonth
BL	1.835075e+06	21229.234108	1.249628e+06	5361.512685	134.422435	6.946548	1995.364525	2003.779656	6.556175
MG	1.840781e+06	49566.169471	1.162381e+06	7320.708979	134.501789	7.454354	1985.300473	2002.911930	6.231483
SSL	2.056529e+06	10701.891039	1.275930e+06	10465.944861	135.259359	6.128482	1999.531559	2005.769996	6.576997
TEX	2.008122e+06	37690.883496	1.201414e+06	7809.703837	135.230625	5.477695	1996.690508	2004.856058	6.420167
TTT	1.857726e+06	37586.693241	1.150703e+06	5356.358655	134.652835	7.147664	1990.685358	2002.950922	6.412130
WL	1.900488e+06	38420.446074	1.216930e+06	6155.224828	134.901601	6.383828	1990.792114	2003.446246	6.276651

In [7]:
```
df['PGNum'] = df['ProductGroup']


df.head()

Counter(df.PGNum)
df.head()
df.to_csv('0708_industry_mach_dfpass2_1.csv')
```

In [8]:
```
def PGconv(df):
    if df['PGNum'] == 'SSL':
        return 10000
    elif df['PGNum'] == 'BL':
        return 21000
    elif df['PGNum'] == 'TEX':
        return 37700
    elif df['PGNum'] == 'TTT':
        return 37600
    elif df['PGNum'] == 'WL':
        return 38400
    elif df['PGNum'] == 'MG':
        return 49500
    else:
        return 0

df['PGNum'] = df.apply(PGconv, axis=1)

#df.reset_index(inplace = True)

#df.drop(columns=['level_0'],inplace = True)
df.head(10)
```

Out[8]:

	SalesID	SalePrice	MachineID	ModelID	datasource	auctioneerID	YearMade	saledate	ProductGroup	Enclosure	saleyear	salemonth	PGNum
0	1263845	11000	1542938	3208	132	5.0	1919	1998-04-18	BL	OROPS	1998	4	21000
1	1271235	8500	1393999	6633	132	15.0	1919	2011-04-13	SSL	OROPS	2011	4	10000
2	1284397	20000	1226376	3178	132	15.0	1919	2011-10-12	BL	OROPS	2011	10	21000
3	1285083	20000	1346350	3178	132	15.0	1919	2011-10-12	BL	OROPS	2011	10	21000
4	1288729	18000	1525079	3238	132	15.0	1919	2011-04-13	TTT	OROPS	2011	4	37600
5	1362939	15500	1227772	7267	132	6.0	1919	1995-03-02	WL	EROPS	1995	3	38400
6	1363002	17000	1387642	7267	132	2.0	1919	2000-02-07	WL	EROPS	2000	2	38400
7	1368428	8500	1283255	4146	132	7.0	1919	2007-03-07	TTT	OROPS	2007	3	37600
8	1369229	10500	1429950	4146	132	2.0	1919	1999-09-25	TTT	OROPS	1999	9	37600
9	1369230	16000	1479540	4146	132	24.0	1919	1999-08-12	TTT	OROPS	1999	8	37600

In [9]:
```
Counter(df.Enclosure)
```

Out[9]:
```
Counter({'EROPS': 121622,
         'EROPS AC': 15,
         'EROPS w AC': 82059,
         'NO ROPS': 2,
         'None or Unspecified': 2,
         'OROPS': 158985,
         nan: 255})
```

In [17]:
```
df.groupby('EnclosureNum').mean()
#df.groupby('EnclosureNum').max()
#df['EnclosureNum'] = df['Enclosure']
#df.head()
```

Out[17]:

EnclosureNum	SalesID	SalePrice	MachineID	ModelID	datasource	auctioneerID	YearMade	saleyear	salemonth	PGNum
EROPS	1.826493e+06	30005.778716	1.237961e+06	6539.016650	134.110761	6.626178	1990.635872	2002.598872	6.504382	35862.473895
EROPS AC	1.950231e+06	26000.000000	1.077282e+06	5027.600000	137.600000	6.466667	1991.066667	2004.733333	7.733333	40266.666667
EROPS w AC	2.158537e+06	52935.864500	1.124256e+06	7358.457415	136.925078	5.247004	1998.673869	2006.752044	6.473281	37066.990824
NO ROPS	1.839422e+06	45250.000000	1.318022e+06	4143.000000	134.000000	3.500000	1981.000000	2004.500000	3.000000	37600.000000
None or Unspecified	1.582880e+06	16500.000000	1.220914e+06	4649.500000	132.000000	1.500000	1992.000000	2003.000000	10.000000	37700.000000
OROPS	1.863697e+06	23223.868239	1.231898e+06	6658.943775	134.340290	6.988388	1993.612982	2003.602591	6.342007	26632.974180

In [18]:
```
def Enclconv(df):
    if df['EnclosureNum'] == 'EROPS':
        return 30000
    elif df['EnclosureNum'] == 'EROPS AC':
        return 26000
    elif df['EnclosureNum'] == 'EROPS w AC':
        return 53000
    elif df['EnclosureNum'] == 'NO ROPS':
        return 45000
    elif df['EnclosureNum'] == 'OROPS':
        return 23000
    elif df['EnclosureNum'] == 'Unspecified':
        return 16000
    elif df['EnclosureNum'] == 'None':
        return 16000
    else:
        return 0

df['EnclosureNum'] = df.apply(Enclconv, axis=1)
```

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

Out[19]:

	SalesID	SalePrice	MachineID	ModelID	datasource	auctioneerID	YearMade	saledate	ProductGroup	Enclosure	saleyear	salemonth	PGNum	EnclosureNu
0	1263845	11000	1542938	3208	132	5.0	1919	1998-04-18	BL	OROPS	1998	4	21000	2300
1	1271235	8500	1393999	6633	132	15.0	1919	2011-04-13	SSL	OROPS	2011	4	10000	2300
2	1284397	20000	1226376	3178	132	15.0	1919	2011-10-12	BL	OROPS	2011	10	21000	2300
3	1285083	20000	1346350	3178	132	15.0	1919	2011-10-12	BL	OROPS	2011	10	21000	2300
4	1288729	18000	1525079	3238	132	15.0	1919	2011-04-13	TTT	OROPS	2011	4	37600	2300

In [20]:
```
#Counter(df.PGNum)
#Counter(df.ModelID)
#df.info()
dfpass2 = df
dfpass2.head()
#pd.to_numeric(dfnum1['fiBaseModel'], errors='coerce')
#dfnum1.drop(columns=['state'],inplace = True)
dfpass2.info()
dfpass2.to_csv('0708_industry_mach_reg_dfpass2_2converts')
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 362940 entries, 0 to 362939
Data columns (total 14 columns):
SalesID         362940 non-null int64
SalePrice       362940 non-null int64
MachineID       362940 non-null int64
ModelID         362940 non-null int64
datasource      362940 non-null int64
auctioneerID    345366 non-null float64
YearMade        362940 non-null int64
saledate        362940 non-null datetime64[ns]
ProductGroup    362940 non-null object
Enclosure       362685 non-null object
saleyear        362940 non-null int64
salemonth       362940 non-null int64
PGNum           362940 non-null int64
EnclosureNum    362940 non-null int64
dtypes: datetime64[ns](1), float64(1), int64(10), object(2)
memory usage: 38.8+ MB
```

In [31]:
```
dfpass2.head()
dfpass2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 362940 entries, 0 to 362939
Data columns (total 12 columns):
SalesID         362940 non-null int64
SalePrice       362940 non-null int64
MachineID       362940 non-null int64
ModelID         362940 non-null int64
datasource      362940 non-null int64
auctioneerID    345366 non-null float64
YearMade        362940 non-null int64
saledate        362940 non-null datetime64[ns]
saleyear        362940 non-null int64
salemonth       362940 non-null int64
PGNum           362940 non-null int64
EnclosureNum    362940 non-null int64
dtypes: datetime64[ns](1), float64(1), int64(10)
memory usage: 33.2 MB
```

In [43]:
```
#dfpass3 = dfpass2

#dfpass2.drop(columns=['auctioneerID'],inplace = True)
dfpass2.to_csv('0708_industry_mach_reg_dfpass2_3allnumeric')
dfpass2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 362940 entries, 0 to 362939
Data columns (total 10 columns):
SalesID         362940 non-null int64
SalePrice       362940 non-null int64
MachineID       362940 non-null int64
ModelID         362940 non-null int64
datasource      362940 non-null int64
YearMade        362940 non-null int64
saleyear        362940 non-null int64
salemonth       362940 non-null int64
PGNum           362940 non-null int64
EnclosureNum    362940 non-null int64
dtypes: int64(10)
memory usage: 27.7 MB
```

In [44]:
```
y = dfpass2['SalePrice'].values
X = dfpass2[['SalesID','MachineID','ModelID','datasource','YearMade','saleyear','salemonth','PGNum','EnclosureNum']].valu
#'fiModelDesc','fiBaseModel','salemonth','Enclosure','AC','PGNum'

X = sm.add_constant(X)
#beta = np.linalg.solve(X.T.dot(X), X.T.dot(y))
#beta

model = sm.OLS(y, X).fit()
model.summary()
```

Out[44]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.461
Model:	OLS	**Adj. R-squared:**	0.461
Method:	Least Squares	**F-statistic:**	3.446e+04
Date:	Sun, 08 Jul 2018	**Prob (F-statistic):**	0.00
Time:	09:41:01	**Log-Likelihood:**	-4.0547e+06
No. Observations:	362940	**AIC:**	8.109e+06
Df Residuals:	362930	**BIC:**	8.109e+06
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-4.653e+04	1.14e+04	-4.064	0.000	-6.9e+04	-2.41e+04
x1	-6.164e-05	5.4e-05	-1.142	0.253	-0.000	4.41e-05
x2	-0.0067	6.93e-05	-97.369	0.000	-0.007	-0.007
x3	-0.1856	0.005	-38.104	0.000	-0.195	-0.176
x4	116.7649	5.217	22.383	0.000	106.540	126.990
x5	1054.9518	4.163	253.428	0.000	1046.793	1063.111
x6	-1037.3042	6.826	-151.954	0.000	-1050.684	-1023.925
x7	-200.0551	8.433	-23.723	0.000	-216.583	-183.527
x8	0.9558	0.003	314.010	0.000	0.950	0.962
x9	0.5782	0.003	207.319	0.000	0.573	0.584

Omnibus:	68009.370	**Durbin-Watson:**	0.888
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	164485.614
Skew:	1.053	**Prob(JB):**	0.00
Kurtosis:	5.538	**Cond. No.**	9.77e+08

In [ ]:
```

```

In [45]:
```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

#df_x --> new dataframe of independent variables
df_x = dfpass2.drop(['SalePrice'], 1)

#df_y --> new dataframe of dependent variable views
df_y = dfpass2.ix[:, ['SalePrice']]

names = [i for i in list(df_x)]

regr = LinearRegression()
x_train, x_test, y_train, y_test = train_test_split(df_x, df_y, test_size = 0.2)

#Fitting the model to the training dataset
regr.fit(x_train,y_train)
regr.intercept_
print('Coefficients: \n', regr.coef_)
print("Mean Squared Error(MSE): %.2f"
      % np.mean((regr.predict(x_test) - y_test) ** 2))
print('Variance Score: %.2f' % regr.score(x_test, y_test))
regr.coef_[0].tolist()
```

```
Coefficients:
 [[-4.08783328e-05 -6.77444754e-03 -1.86746003e-01  1.16720034e+02
   1.05520984e+03 -1.03871180e+03 -1.97270544e+02  9.56855544e-01
   5.77172864e-01]]
Mean Squared Error(MSE): 295641725.61
Variance Score: 0.46
```

Out[45]:
```
[-4.087833281106201e-05,
 -0.006774447539389855,
 -0.1867460033028154,
 116.7200338371154,
 1055.2098392112766,
 -1038.7118019330098,
 -197.27054403267798,
 0.956855543830569,
 0.5771728635452518]
```

In [46]:
```
import statsmodels.api as sm
from statsmodels.sandbox.regression.predstd import wls_prediction_std
model=sm.OLS(y_train,x_train)
result = model.fit()
print(result.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              SalePrice   R-squared:                       0.813
Model:                            OLS   Adj. R-squared:                  0.813
Method:                 Least Squares   F-statistic:                 1.406e+05
Date:                Sun, 08 Jul 2018   Prob (F-statistic):               0.00
Time:                        09:41:17   Log-Likelihood:            -3.2438e+06
No. Observations:              290352   AIC:                         6.488e+06
Df Residuals:                  290343   BIC:                         6.488e+06
Df Model:                           9
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
SalesID       3.059e-05   5.68e-05      0.538      0.590   -8.08e-05       0.000
MachineID       -0.0068   7.74e-05    -87.850      0.000      -0.007      -0.007
ModelID         -0.1851      0.005    -34.137      0.000      -0.196      -0.174
datasource     115.1617      5.821     19.785      0.000     103.753     126.570
YearMade      1054.0426      4.645    226.915      0.000    1044.938    1063.147
saleyear     -1059.5750      4.662   -227.279      0.000   -1068.712   -1050.438
salemonth     -200.1567      9.390    -21.315      0.000    -218.562    -181.752
PGNum            0.9547      0.003    285.134      0.000       0.948       0.961
EnclosureNum     0.5797      0.003    190.854      0.000       0.574       0.586
==============================================================================
Omnibus:                    54502.413   Durbin-Watson:                   1.999
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           131794.870
Skew:                           1.055   Prob(JB):                         0.00
Kurtosis:                       5.538   Cond. No.                     7.17e+05
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 7.17e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [161]:

In [ ]:

In [ ]:

In [ ]: