Stats 504 Assignment 1: Buying a laptop from eBay

Introduction

This report analyzes laptop prices, in an attempt to understand what features of a laptop influence its selling price. The findings from this analysis are aimed to enable the client to select a favorable laptop deal on EBay. The report also addresses specific questions of interest such as the influence on price by the presence of a Solid State Drive (SSD) and the mode of buying – auction or Buy-It-Now (BIN). The choice of model for this analysis draws from real-world intuition that laptops with better specifications tend to be priced higher, and prices are likely to grow in a linear fashion. Following the analysis, the report also makes laptop recommendations which are considered "good deals" and also describes how BIN and SSD options raise the price of a laptop.

Methods

The goal of the analysis is to help depict and understand the features that influence laptop prices, and provide concrete evidence in identifying a suitable recommendation for a laptop that the client can purchase. Exploratory analysis on the data points towards linear trends in the data, and this is in adherence to real-world intuition. These box plots indicate a positive trend, where the median price of laptops with increasing levels of RAM and GHz, is growing roughly linearly. Owing to this observation, a linear model seemed most appropriate to capture the necessary information from the data.

Figure 1: Seemingly linear increase in price, with a linear increase in RAM size

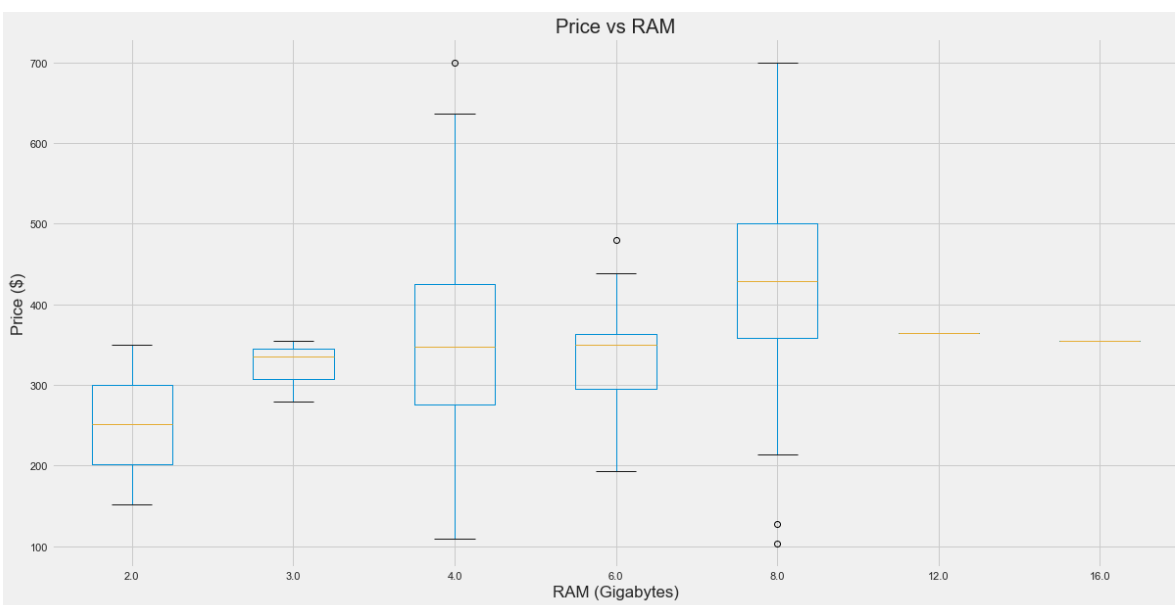


Figure 2: Laptops with Solid State Hard Drives tend to be more expensive than ones without

A Linear Regression Model attempts to fit the best possible straight line through all the data points, such that if given the features of the laptop like RAM, GHz, HD, etc we can (with some level of confidence) estimate the price of a laptop with those specifications, as a point falling on the fitted line. While this model is easily interpretable (discussed in the Results section), it does come with its share of complexities and limitations. If the data were to have inherently non-linear trends, like exponential growth in laptop prices, then this model would not be able to sufficiently capture those trends resulting in a poorly performing model. The model, like most models, also relies heavily on the data that is used to build it. Much care must be taken in vetting the data for the analysis, as even a few outliers can affect the model by a lot, resulting in very different outcomes. All of these were accounted for by performing model diagnostics, and are discussed further in the appendix.

Results

The data provided by the client had details about laptop specifications as well as information about its sale on EBay. Each row of the data represents a different laptop, corresponding to its specific details. However, some rows had missing values (see Table 1 for proportions) or anomalous values (laptop prices less than $5). The data was cleaned using appropriate statistical methods. The anomalous rows were discarded as they were not large in number, and for the missing values, the mode was used. The mode is the most frequently occurring value in the dataset, so this meant missing values were filled in with a value that was most common for that feature (See Appendix for more details). This strategy seemed appropriate for the data at hand due to its discrete nature. Since features like RAM or GHz can only take up certain values and are not truly numerical, using the most commonly occurring value is reasonable. The regression analysis was finally performed on 215 different laptops with each laptop having 7 different features which are further described below.

Feature	**Median (25th, 75th percentile) or Percentage**
Sold (%)	73.02%
Price ($)	361 (300, 450)
Processing Speed (GHz) (%) – 22.2% missing	
2.5	63.72%
2.6	25.11%
2.7	8.88%
2.8	0.93%
3.2	1.39%
RAM (GB) (%) – 19.09% missing	
2	0.93%
4	56.74%
6	7.44%
8	32.55%
12	0.46%
16	0.46%
Hard Disk (GB) 31.36% missing	300 (128, 320)
Solid State Drive (%)	37.20%
Buy-It-Now (%)	53.02%

Table 1: Baseline table indicating summary statistics of data used in the analysis

Multiple variations of regression models were fitted, but only the (subjective) best one is discussed here. The table below describes the effects of statistically significant laptop features on the price.

Feature	Feature Change	Price Change (95% Confidence Interval)
Gigahertz (GHz)	+ 0.1 GHz	+17.00 (32.23 307.80)
RAM (GB)	+ 1 GB	+9.81 (2.51 17.11)
Solid State Drive (SSD)	No -> Yes	+87.97 (47.50 128.44)
Buy-It-Now (BIN)	No -> Yes	+67.12 (39.35 94.89)

Table 2: Price change that is expected to be seen from upgrading a feature

As the table suggests, SSD, BIN, Gigahertz, and RAM, were the features that most influenced the price of a laptop, while Hard Disk Space was deemed as mostly irrelevant to the price by the model. SSD was the feature that influenced price the most ($87.97). To break it down further, if there were two exactly similar features laptops, one with SSD and one without, the one with the SSD would cost $87.97 more. Other features can be interpreted similarly from Table 2.

This model, while being the most sensible fit for the data, was unfortunately not flexible enough to answer some questions regarding the effect of SSD storage on the Hard Disk (HD) Space. Adding an "interaction term" to this model, as a way of modeling the relationship between SSD and HD, makes the updated model unstable and the ability to perform further accurate inferential analysis is lost. While we cannot guarantee a high level of certainty for the following results, it may still be useful to know them. As per the new model, an SSD laptop with an additional 1 GB of HD space would cost $31.87 less than a laptop with the exact specifications for other features. The new model seems to indicate that SSD storage is inconsequential which is intuitively contradictory to what the boxplots and t-tests suggest, which is also another reason the former model was preferred.

In order to make recommendations of good deals for the client, the model can be used to compute the expected price of every laptop and filter desirable ones from the set of laptops that cost lower than what the model suggests is their "true" price. The recommendations can further be filtered based on requirements like presence of SSD, large HD space, BIN option, etc. The following table presents the best deals for the client to consider.

#	Price ($)	GHz	RAM (GB)	BIN Available	SSD	HD Space (GB)	Expected Saving ($)
23	565.00	2.5	8	Yes	Yes	240	90.63
30	500.00	2.7	8	No	Yes	160	58.35
53	564.95	2.5	8	Yes	Yes	128	90.03
100	579.00	2.7	8	No	Yes	160	137.35

Table 3: Recommendations for unsold laptops that are good bargains

All of these recommendations were chosen such that their price on EBay was cheaper as compared to the expected value predicted by the fitted model. This difference in price in depicted in the last column as a measure of how "good" of a deal it is. Since the client preferred to have an SSD, all recommendations have that feature available. The client was undecided about the BIN option, but preferred to not partake in an auction only on the condition that it did not affect the price. The analysis revealed that using an auction may result in landing a better deal, which is why the two recommendations (#30 and #100) have been made. All recommendations have large HD space available as that was another requirement. Considering all of the client's requirements, these recommendations met most of the checkboxes, as well as had a high expected saving, and are expected to be ideal buys for the client.

Conclusion

This report presents the results of an analysis performed by on laptop prices, with the objective of trying to understand what features influence price. It also addresses the key questions posed by the client regarding specific features like SSD, and BIN and their influence on HD space. The presence of SSD and BIN options increase a laptop price by $87.97 and $67.12 respectively, and the cost of adding HD space is statistically insignificant with respect to whether or not the memory type is SSD. While these results are expected to be useful, it must be noted that the model comes with a fair share of limitations. The strategy for filling in missing data could have strongly biased or affected the model, so re-running the analysis with a better dataset may yield different (and perhaps better) results. Finally, the laptop recommendations made in the results section aim to strike a balance between subjectively trying to address the client's needs, while at the same time use the information provided by the model. These recommendations could change based on the reader's interpretation, and the reader is encouraged to look at the appendix for a larger candidate set of potential laptop recommendations.

Appendix

In [11]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.linear_model import LinearRegression
import patsy
import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor
from sklearn import tree
import seaborn as sns
```

In [12]:

```
sns.set(rc = {'figure.figsize': (12, 8)})
plt.style.use('fivethirtyeight')
```

Read in the data and make necessary conversions

In [13]:

```
data = pd.read_csv("laptopData.csv", index_col=0)
data["ssd"] = data["ssd"].map({"SSD": 1, "No": 0})
data["BIN"] = data["BIN"].map({False: 0, True: 1})
data["sale"] = data["sale"].map({"SOLD": 1, "NOT SOLD": 0})
data
```

Out[13]:

	sale	price	ghz	ram	hd	ssd	BIN
1	1	404.99	2.7	8.0	NaN	1	0
2	1	355.00	2.5	8.0	128.0	1	0
3	1	449.99	2.6	4.0	128.0	0	1
4	0	499.99	2.5	4.0	320.0	0	1
5	0	199.99	NaN	NaN	NaN	0	1
...	...	...	...	...	...	...	...
216	1	480.00	NaN	6.0	128.0	1	1
217	1	452.00	2.6	8.0	240.0	1	0
218	1	358.00	2.6	4.0	128.0	0	0
219	1	450.00	NaN	4.0	128.0	1	0
220	1	299.95	2.5	4.0	320.0	0	1

220 rows × 7 columns

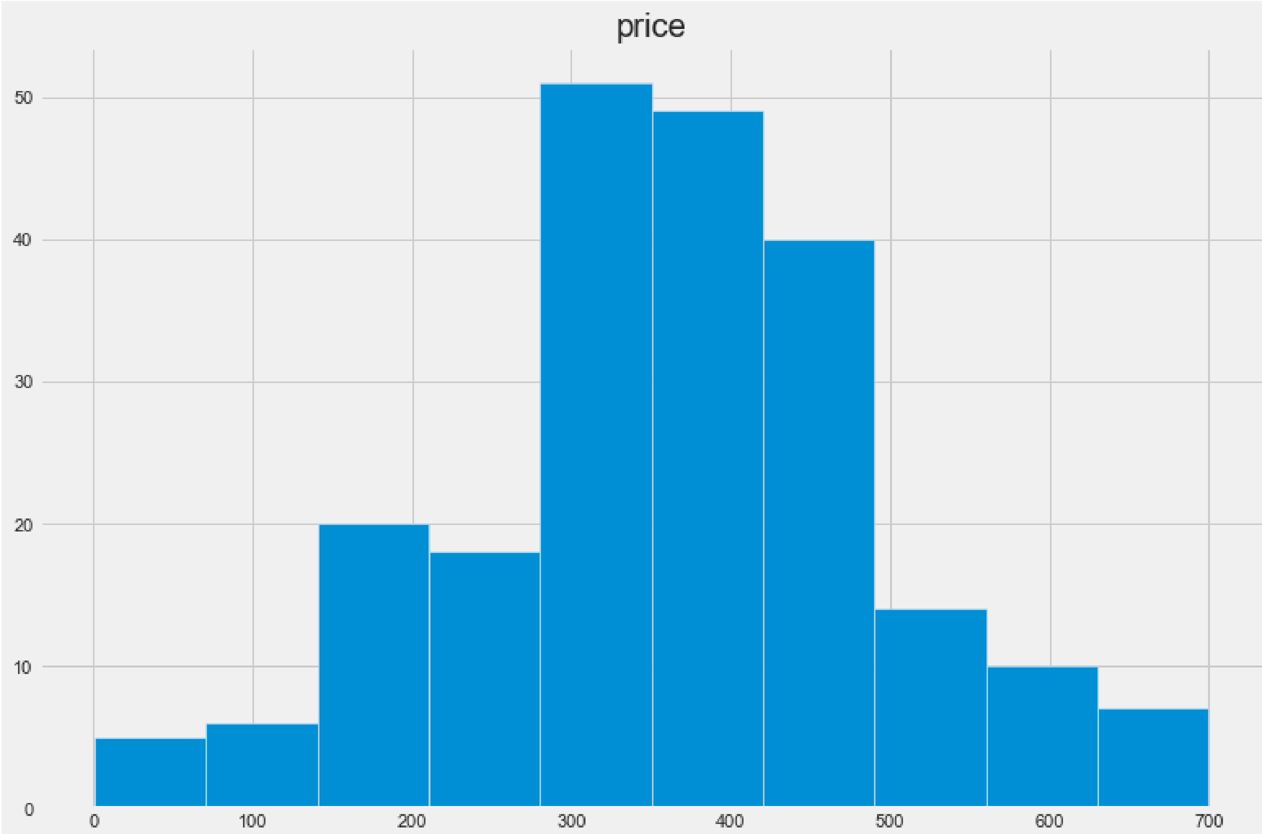
Exploratory analysis on the response variable

In [14]:

```
data.hist(column="price")
```

Out[14]:

```
array([[<AxesSubplot:title={'center':'price'}>]], dtype=object)
```



In [15]:

```
data["price"].describe()
```

Out[15]:

```
count    220.000000
mean     364.162727
std      132.301459
min        1.000000
25%      299.980000
50%      357.500000
75%      449.992500
max      699.990000
Name: price, dtype: float64
```

Drop rows where the price is anomalous

In [16]:

```
data = data.drop(data[data["price"] < 10].index)
```

In [17]:

```
data.isna().sum()
```

Out[17]:

```
sale      0
price     0
ghz      49
ram      42
hd       69
ssd       0
BIN       0
dtype: int64
```

Data Imputation for missing values

In [18]:

```
data.corr().round(2)
```

Out[18]:

	sale	price	ghz	ram	hd	ssd	BIN
sale	1.00	-0.12	0.10	-0.05	-0.05	0.16	-0.22
price	-0.12	1.00	0.28	0.25	-0.16	0.44	0.26
ghz	0.10	0.28	1.00	0.32	-0.08	0.19	-0.02
ram	-0.05	0.25	0.32	1.00	-0.02	0.29	-0.02
hd	-0.05	-0.16	-0.08	-0.02	1.00	-0.54	-0.17
ssd	0.16	0.44	0.19	0.29	-0.54	1.00	-0.01
BIN	-0.22	0.26	-0.02	-0.02	-0.17	-0.01	1.00

In [19]:

```
data[data["ssd"] == 1]["hd"].mode()
```

Out[19]:

```
0    128.0
dtype: float64
```

In [20]:

```
data[data["ssd"] == 0]["hd"].mode()
```

Out[20]:

```
0    320.0
dtype: float64
```

Fill in `hd` with most common (mode) value amongst laptops with same `ssd` value. This is because the highest correlated variable for `hd` is `ssd`

In [21]:

```
data["hd"].fillna(data["ssd"].map({1: 128, 0: 320}), inplace=True)
data.isna().sum()
```

Out[21]:

```
sale      0
price     0
ghz      49
ram      42
hd        0
ssd       0
BIN       0
dtype: int64
```

Similarly, for `ghz` and `ram`, use a stratified mode to fill in missing values

In [22]:

```
data.groupby("ghz")["ram"].agg(pd.Series.mode)
```

Out[22]:

```
ghz
2.5           4.0
2.6           4.0
2.7           8.0
2.8    [4.0, 8.0]
3.2           8.0
Name: ram, dtype: object
```

In [23]:

```
data.groupby("ram")["ghz"].agg(pd.Series.mode)
```

Out[23]:

```
ram
2.0     2.5
```

```
3.0     2.5
4.0     2.5
6.0     2.6
8.0     2.5
12.0     []
16.0     []
Name: ghz, dtype: object
```

In [24]:

```
imputed_data_ram = data["ram"].fillna(data["ghz"].map({2.5: 4, 2.6: 4, 2.7: 8, 2
imputed_data_ghz = data["ghz"].fillna(2.5) # most values are 2.5, so fill with t
data["ram"] = imputed_data_ram
data["ghz"] = imputed_data_ghz
data.isna().sum()
```

Out[24]:

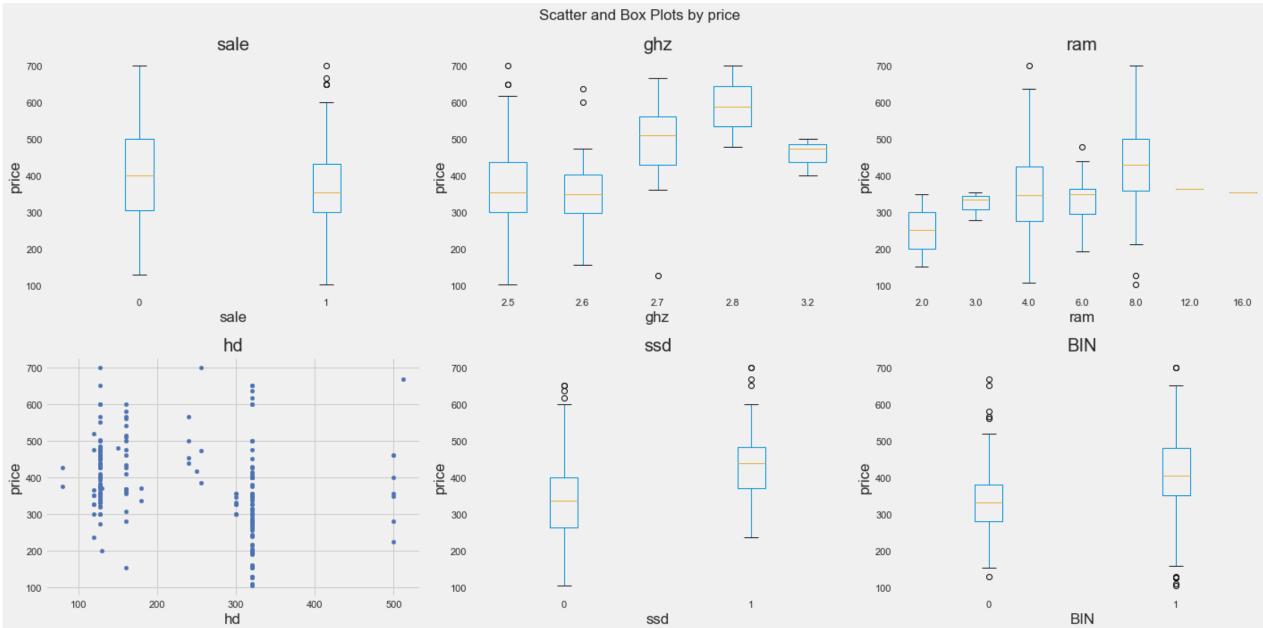
```
sale      0
price     0
ghz       0
ram       0
hd        0
ssd       0
BIN       0
dtype: int64
```

Model Selection & Variable Selection

In [25]:

```
plt.rcParams["figure.figsize"] = (20,10)
def draw_outcome_plots(df, outcome, n_rows, n_cols):
    fig=plt.figure()
    variables = df.columns.drop(outcome)
    for i, var_name in enumerate(variables):
        ax=fig.add_subplot(n_rows,n_cols,i+1)
        if len(df[var_name].unique()) > 10:
            df.plot.scatter(x= var_name, y= outcome, ax=ax)
        else:
            df.boxplot(column=outcome, by=var_name, grid = False, ax=ax)
        ax.set(ylabel=outcome)
        ax.set_title(var_name)
    fig.suptitle('Scatter and Box Plots by '+outcome)
    fig.tight_layout()
    plt.show()
draw_outcome_plots(data, 'price', 2, 3)
```

```
*c* argument looks like a single numeric RGB or RGBA sequence, which should be a
voided as value-mapping will have precedence in case its length matches with *x*
& *y*.  Please use the *color* keyword-argument or provide a 2-D array with a si
ngle row if you intend to specify the same RGB or RGBA value for all points.
```



Looking at boxplots, there seem to be linear trends across all variables except `sale` and `hd`. This may need further investigation.

This is indicative of a linear regression model, also because the response variable is normally distributed.

Exclude `sale` since that is being influenced by the response variable

In [26]:

```
y1, X1 = patsy.dmatrices("price ~ ghz + ram + hd + ssd + BIN", data, return_type
model1 = sm.OLS(y1, X1).fit()
print(model1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.312
Model:                            OLS   Adj. R-squared:                  0.296
Method:                 Least Squares   F-statistic:                     18.95
Date:                Thu, 08 Sep 2022   Prob (F-statistic):           1.58e-15
Time:                        15:33:47   Log-Likelihood:                -1296.4
No. Observations:                 215   AIC:                             2605.
Df Residuals:                     209   BIC:                             2625.
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept   -183.0675    178.151     -1.028      0.305    -534.270     168.135
ghz          170.0113     69.891      2.433      0.016      32.230     307.793
ram            9.8137      3.703      2.650      0.009       2.514      17.113
hd            -0.0050      0.092     -0.054      0.957      -0.186       0.176
ssd           87.9694     20.529      4.285      0.000      47.498     128.441
BIN           67.1178     14.087      4.765      0.000      39.347      94.888
==============================================================================
Omnibus:                        5.752   Durbin-Watson:                   1.836
Prob(Omnibus):                  0.056   Jarque-Bera (JB):                6.366
Skew:                           0.242   Prob(JB):                       0.0415
Kurtosis:                       3.691   Cond. No.                     7.12e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
```

```
specified.
[2] The condition number is large, 7.12e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Also check with an interaction term of `hd:ssd` since that is an inference we are interested in

In [27]:

```
y2, X2 = patsy.dmatrices("price ~ ghz + ram + hd*ssd + BIN", data, return_type="
model2 = sm.OLS(y2, X2).fit()
print(model2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.341
Model:                            OLS   Adj. R-squared:                  0.322
Method:                 Least Squares   F-statistic:                     17.96
Date:                Thu, 08 Sep 2022   Prob (F-statistic):           9.41e-17
Time:                        15:33:47   Log-Likelihood:                -1291.7
No. Observations:                 215   AIC:                             2597.
Df Residuals:                     208   BIC:                             2621.
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    -25.7613    182.228     -0.141      0.888    -385.012     333.489
ghz          124.4547     70.168      1.774      0.078     -13.877     262.786
ram            9.0168      3.641      2.476      0.014       1.838      16.195
hd            -0.1403      0.100     -1.397      0.164      -0.338       0.058
ssd          -32.5746     44.453     -0.733      0.465    -120.211      55.062
hd:ssd         0.7034      0.231      3.042      0.003       0.248       1.159
BIN           70.3974     13.859      5.080      0.000      43.076      97.719
==============================================================================
Omnibus:                        7.678   Durbin-Watson:                   1.848
Prob(Omnibus):                  0.022   Jarque-Bera (JB):                9.815
Skew:                           0.260   Prob(JB):                      0.00739
Kurtosis:                       3.908   Cond. No.                     7.50e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 7.5e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

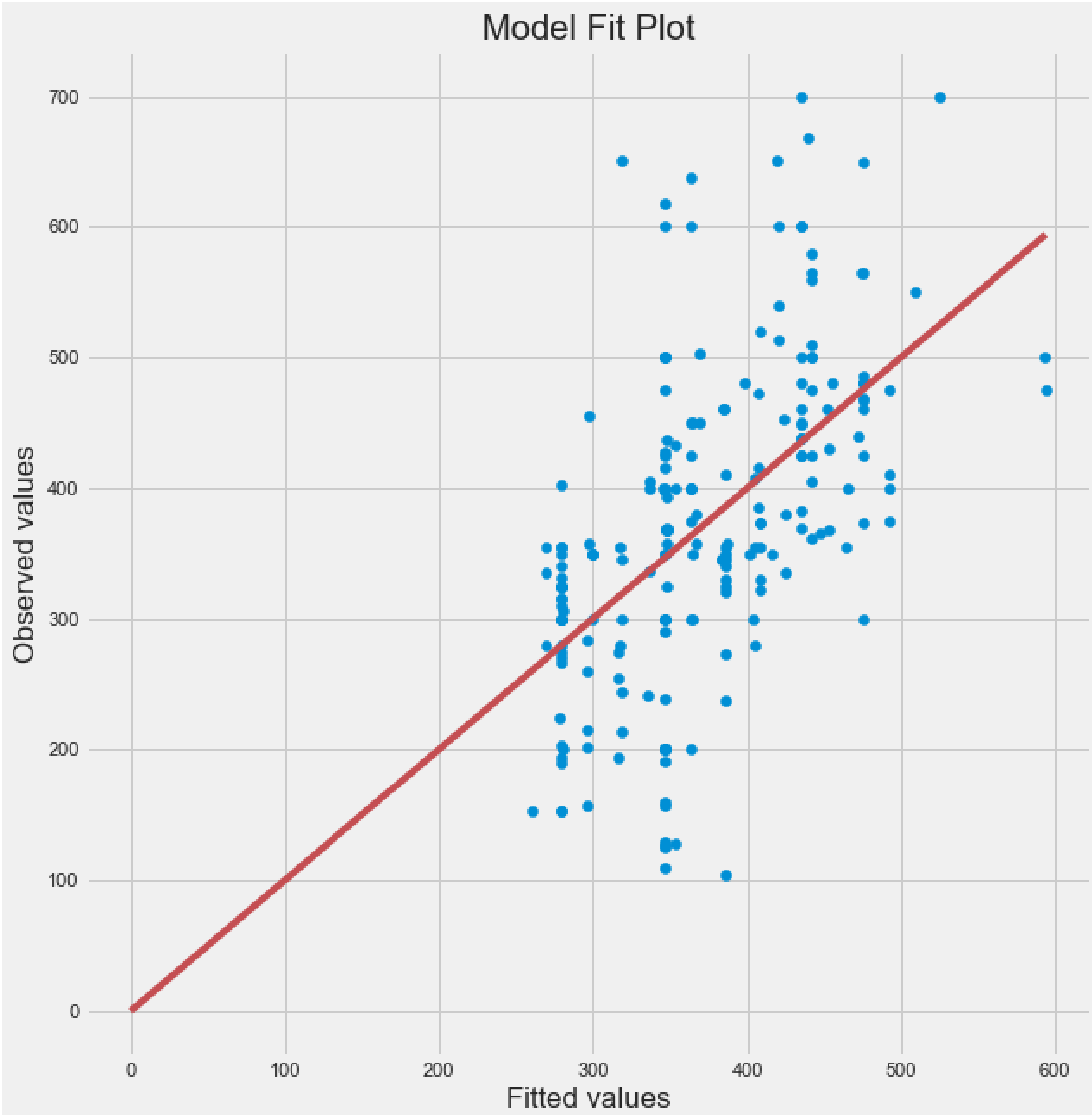
AIC values are very similar, so prefer to keep the model without the interaction term as it is more easily interpretable, and the coefficients seem more reasonable and closer to reality

Model Diagnostics

In [44]:

```
def diagnostic_plots(fit, response):
    fig, (ax1) = plt.subplots(1,1)
    ax1.scatter(fit.fittedvalues, response)
    abline_max = min(max(fit.fittedvalues), max(response))
    ax1.plot([0, abline_max], [0, abline_max], color='r')
    ax1.set_title('Model Fit Plot')
    ax1.set_ylabel('Observed values')
    ax1.set_xlabel('Fitted values');
    ax1.set_box_aspect(1)

diagnostic_plots(model1, data.price)
```



Recommendation

From the best model, predict prices and look at laptops that have lower prices than the predictions. These are "good" deals. Further filter based on requirements like `ssd=1` , and `BIN=0` (since this makes the price cheaper) and large HD capacity

In [36]:

```
data["savings"] =  data["price"] - model1.predict(X1)
data[(model1.predict(X1) < data["price"]) & (data["ssd"] == 1)  & (data["sale"]
```

Out[36]:

	sale	price	ghz	ram	hd	ssd	BIN	savings
6	0	699.95	2.5	4.0	128.0	1	1	264.285959
7	0	437.71	2.5	4.0	128.0	1	1	2.045959
23	0	565.00	2.5	8.0	240.0	1	1	90.639956
30	0	500.00	2.7	8.0	160.0	1	0	58.356284

	sale	price	ghz	ram	hd	ssd	BIN	savings
36	0	437.71	2.5	4.0	128.0	1	1	2.045959
39	0	499.99	2.7	8.0	128.0	1	0	58.186608
53	0	564.95	2.5	8.0	128.0	1	1	90.031090
54	0	560.00	2.7	8.0	160.0	1	0	118.356284
75	0	565.00	2.7	8.0	160.0	1	0	123.356284
100	0	579.00	2.7	8.0	160.0	1	0	137.356284

In [37]:

```
data.loc[[23, 30, 53, 100]]
```

Out[37]:

	sale	price	ghz	ram	hd	ssd	BIN	savings
23	0	565.00	2.5	8.0	240.0	1	1	90.639956
30	0	500.00	2.7	8.0	160.0	1	0	58.356284
53	0	564.95	2.5	8.0	128.0	1	1	90.031090
100	0	579.00	2.7	8.0	160.0	1	0	137.356284

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: