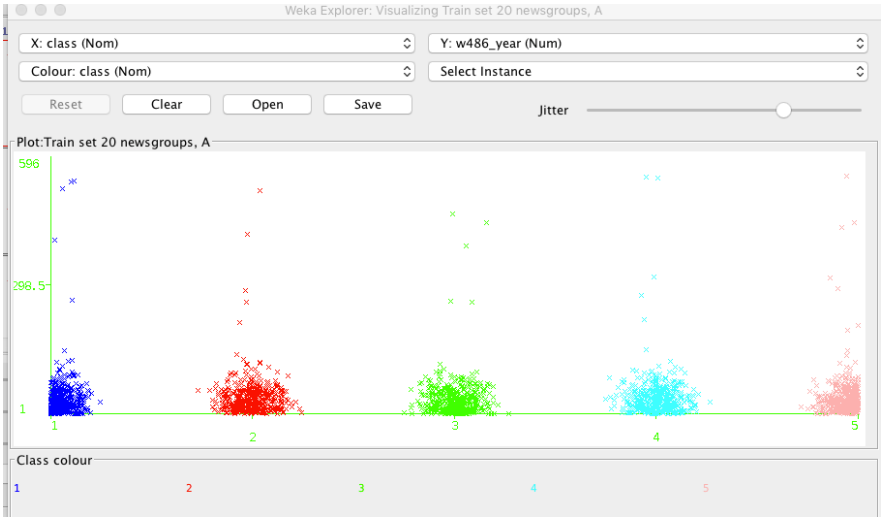
1.1 Exploration of dataset

a) Observations about the data



Above scatter plot (jitter =very high) shows almost all data points concentrated on x-axis (intervals 1-5). This is for all plots where x-axis is one of the attributes(words) and y-axis is class. For most of instances, the 520 attributes are present in a small number (~1-10)while for some it's large. Since data is bag-of-words this means some documents are very large while most of the documents are small.

(b) ZeroR reasonable baseline (Weka default classifier) against which both classifiers can be compared. Selects most frequent value from its frequency table to make predictions, ignoring other factors, relying only on target.
Evaluation, comparison, 5-fold CV

	Decision tree (J48)	**Naive Bayes(NB)**	**ZeroR**
PC (Percent Correct)	**77.9353%**	**20.1595%**	21.2672 %
Mean absolute error	0.0942	0.3194	0.3195
Root mean squared error	0.279	0.5651	0.3997

Table 1

NB

```
=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
   1 154  77  76  80 |   a = 1
   1 180  90  91  97 |   b = 2
   0 179  87  90  93 |   c = 3
   0 190  95  94 101 |   d = 4
   1 192  95 100  93 |   e = 5
```

J48

```
=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 333  13  11  22   9 |   a = 1
  19 338  64  26  12 |   b = 2
  15  64 345  16   9 |   c = 3
  33  18  19 363  47 |   d = 4
  14  13  13  61 380 |   e = 5
```

Above (highlighted in bold) PC of J48 is much higher than NB. NB performs a little worse than baseline ZeroR. From the confusion matrix, NB is incorrectly classifying a lot of classes to class2 and almost none to class1. Since NB makes the strong assumption that all attributes are independent of each other whereas for this data set (looking at confusion matrix for J48), it is clear that class2 and 3 are closely related, so is class4 and 5, it might be the reason while NB performs so poorly here.

c) After opening scatter plot and clicking on the far off data points manually , following extreme/outlier instances were observed e.g. (124,160,197....)
Starting with first index(w1_aaa)
Filter used - weka.filters.unsupervised.attribute.interquartilerange –R 1 –O 3.0-E 6.0
No outliers detected ,but 59 extreme values detected
Removed using filter - weka.filters.unsupervised.instance.RemoveWithValues (attribute index 523, nominal indices last)
Slight improvement in PC (78%) using above.

(If instead of taking one attribute at a time, all attributes taken together, interquartilerange –R first -last –O 3.0-E 6.0 is used, the filter classifies almost all (2251) as extreme values and 387 as outliers)

1.2 Feature Selection

(a) InfoGainAttributeEval evaluates the worth of an attribute by measuring the information gain with respect to the class. It works along with ranker search of weka. But sometimes this evaluator causes overfitting as it does not see the relations among the attributes but only between attribute and final class.

Top 5 team,hockey,game,mac,good
Bottom 5 sooner,constantly,resulting,old,sit

Visualize-scatter plot with high jitter (y axis is always the class) shows that top 5 words concentrated around y axis(values 1-5) while its x axis values between (1-2) and bottom 5 attributes are scattered evenly on x-axis(1-7) and y axis(1-5)

(b) Retrain and evaluate the Naive Bayes (NB) classifier on the reduced dataset from Part (a) after removing attributes 501-520(lowest ranked)
5-fold CV, number_instances = 2257

	Naïve Bayes(full dataset)	Naive Bayes (20 attributes removed)
(PC –Percent Correct)	**72.1599 %**	**71.9802 %**
Mean absolute error	0.1121	0.1122
Root mean squared error	0.3195	0.3194

Table 2

Removing the least useful 20 suggests there is hardly any contribution of these 20 attributes to the training and performance of classifier since the PC mostly remains unchanged (0.2% change). So these 20 attributes can be safely removed.

Part II - 2.1 Simple Linear Regression

(a) Scatter plot - Instance 33,45,47,48, and 49 are scattered way too far from the other data points, and its possible that they are outliers.
No, engine-power alone is not sufficient for predicting the price the way in which the data points are spread out, a simple line cannot be drawn on this plot, engine-power is weakly correlated to price. To improve the performance of linear regression, the dataset can be preprocessed by removing the outlier and extreme values (for instances = 46,47,48,49,33) , after which a straight line can fit better in the plot.

(b) Simple linear regression model (default setting 10 CV)

Price = 0.09 * engine-power + 3038.37 ----------equation(i)

Correlation coefficient	0.405
Mean absolute error	3999.335
Root mean squared error	6155.6971
Relative absolute error	80.8826 %
Root relative squared error	90.7412 %
Total Number of Instances	159

Table 3

As one more unit of engine-power is added, the price increases by **0.03%** (~0.029)

Eqaution(i) indicates that engine-power is **not** an important influential variable on price since the price fluctuates only by a small amount (0.03%) as value of engine_power changes.

(c) RMSE , MAE and CC are recorded in Table3. These metrics measure how the much the model is deviating from the predicting the correct values given a training dataset. From (a) it can be observed that the RMSE of this model is high because of the outliers/extreme values present in data due to which the model is not fitting properly. Also as seen in (a) fitting a simple linear model in this dataset (when it looks like there is no obvious linear relationship) is causing the RMSE and MAE to become very high.

Part II - 2.2 Multivariate Linear Regression

(a) Visualize tab - examining all the attributes following 5 look like they are particularly good at predicting the price.

1. Length
2. Width
3. Engine-size
4. Highway-mpg
5. Mean-effective-pressure

Torque attributes appear useless for predicting the price as it hardly changes with price and can be safely removed.
Engine_size increases very rapidly with price.

Attribute torque and Mean-effective-pressure exhibit significant correlations (torque decreases exponentially as Mean-effective-pressure increases)

From equation(ii) below, engine_power contributes very less (0.0065 coefficient) to the value of price.

(b)

	Linear Regression Model	Simple Linear Regression Model (1c)	Comments
CC	0.7307	0.405	CC increases by 0.3 % showing that the variables are more strongly related with the price now, than in 1(c)
MAE	3121.3782	3999.335	Engine_size in 1(c) was not good at predicting price Alone, so the error was 6155, but after more variables Were added to the equation(i), the model became better
RMSE	4837.2388	6155.6971	And predicts better with RMSE drops to 4837. Also MAE shows improvement but less than RMSE, so the extreme values are present which are causing RMSE to change much rapidly than MAE.

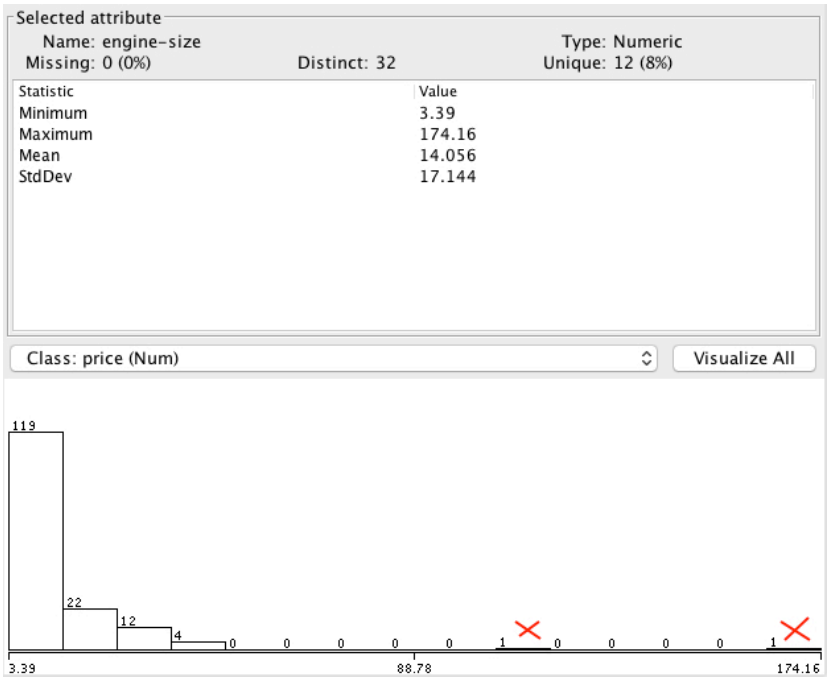
Table 4

Linear Regression Model

Price =

-1.6594 * normalized-losses +
5.5417 * wheel-base +
48.9261 * length +
778.5359 * width +
236.0547 * height +
199.6714 * engine-size +
-216.9238 * bore +
-1105.1252 * stroke +
236.9192 * compression-ratio +
-0.0065 * engine-power +
0.7614 * peak-rpm +
-31.9451 * city-mpg +
-246.3297 * highway-mpg +
-20.5548 * mean-effective-pressure +
0.1126 * torque +
-55438.8267 ----------equation(ii)

(c) Histogram for the engine-size attribute- distribution shows 2 data points (crossed out in red in image below) for engine_size = 107.77 and 174.16) far off which will degrade the performance of the model.



Transformation technique used - removing outliers and extreme values- by running filter –interquartile range (default values)

	Engine_size (original value)	engine_size (transformed value)
min	3.39	3.39
max	**174.16**	**38.36**
mean	14.056	12.44
standard deviation	17.144	8.627

Table 5

After removing 2 extreme from engine-size the rebuilt linear regression model is:

Price =
-9.3835 * normalized-losses +
122.5559 * wheel-base +
11.0998 * length +
551.8369 * width +
178.9042 * height +
439.4961 * engine-size +
-441.0359 * bore +
-1904.2023 * stroke +
173.6366 * compression-ratio +
-0.0142 * engine-power +
1.1732 * peak-rpm +
48.4132 * city-mpg +
-198.3955 * highway-mpg +
-20.4632 * mean-effective-pressure +
-0.2216 * torque +
-45545.3883

	Linear Regression Model (Original)	Linear Regression Model (Rebuilt)
Total Number of Instances	159	157
CC	0.7307	0.7929
MAE	3121.3782	2601.4881
RMSE	**4837.2388**	**3672.2438**

Table 6

The table above shows there is considerable improvement in squared error of the model , after transforming the engine_size and removing the two extreme values because the histogram is no longer distorted and the model can fit better with the given data.

(d)

Interaction terms attempt to capture the relation between the set of independent variables and the dependent one(price). If this relation is non-linear, then by multiplying (or other operations) these interaction terms, a better fitting model can be obtained.

By adding all the attributes (interaction terms) one by one with engine_size and multiplying using addExpression in Weka, eng_power and compression_ratio are the two attributes, which improved the performance of the model. But the biggest performance improvement was achieved by multiplying engine_size with itself. The table shows the comparision of all 3 scenarios. Rest of 11 remaining attributes made the performance worse.

	eng_size * eng_power)	(eng_size * compression ratio)	(eng_size * eng_size)
CC	0.7918	0.8229	0.8227
MAE	2928.1583	2750.0205	2686.8541
RMSE	4246.8189	**3931.2963**	**3833.0373**

Table 7

References:

1. Weka Manual Pg 16 1.2.2)
2. ZeroR – Youtube tutorial
3. Weka Tutorial 10: Feature Selection with Filter (Data Dimensionality)