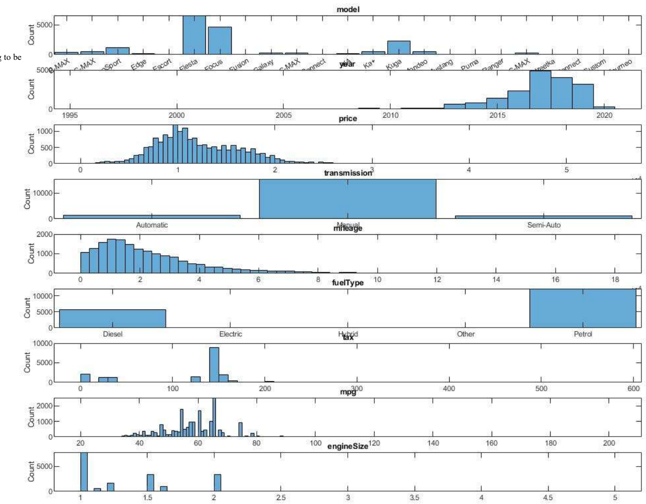
Linear Regression Versus Random Forest Regression



Problem Description

In this dataset we have 17,965 rows of data about used Ford cars and nine columns with their model, registration year, price, transmission type, mileage, fuel type, road tax, miles per gallon and engine size. From this data we are going to predict the price of a car from some, or all, of the remaining eight columns. We are going to be holding back 30% of the dataset for testing and training on the remaining 70% (12576 training rows and 5389 testing rows).

Exploratory Data Analysis

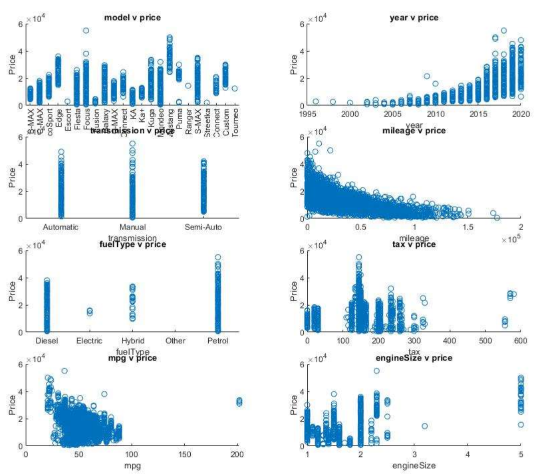
Summary Statistics:

	Year	Price	Mileage	Tax	MPG	Engine Size
Mean	2,016.87	12,279.76	23,363.63	113.33	57.91	1.35
Median	2017	11,291	18,243	145	58.9	1.2
Min	1996	495	1	0	20.8	0
Max	2060	54,995	177,644	580	201.8	5
Standard Deviation	2.05	4,741.38	19,472.11	62.01	10.13	0.43

	Year	Price	Mileage	Tax	MPG	Engine Size
Mean	2,016.86	12,279.76	23,363.63	113.33	57.91	1.35
Median	2017	11,291	18,243	145	58.9	1.2
Min	1996	495	1	0	20.8	0
Max	2020	54,995	177,644	580	201.8	5
Deviation	2.03	4,741.38	19,472.11	62.01	10.13	0.43

Post Change

From this we can see that the maximum registration year is 2060, which is wrong, as a car can't be registered in the future (it is one car, which we will update to 2020, as that is assumed to be what it should be). For the engine size we have 51 petrol, diesel and hybrid cars with an engine size of 0 (the two electric cars have an engine size of 2 litres, which is either a filler number or incorrect, as the "engine size is the amount of air and fuel that can be forced into the cylinders of the engine" [1]).
For now, where the engine size is zero, we will replace this with the mean of the non-zero engine sizes, as in reality the petrol, diesel and hybrid cars will have a non-zero engine size. For the electric cars we are assuming that the engine size is meant to convey the power of the electric engine (as there are only two of them this might be the only way their price estimate is anywhere near accurate).
From the histograms we can see that Fiestas and Focuses are the most common cars in the data set.
Most of the cars were registered in the last 3 or so years, with occasional much older cars.
The price is very approximately normally distributed around £10k (the picture cropped the scale) with it varying down to almost £0 and trailing off prior to £30k (with a few cars going up to almost £60k). So, we may not have sufficient data to be able to generalise well to the higher end of the market (at the lower end of the scale there comes a point where the car may be worth more as scrap).
The overwhelming majority of the cars in the dataset are manual cars, with 1307 automatic and 1087 semi-automatic cars, so this may be enough to generalise to those.
The mileage is right skewed around the 10k miles mark, with the majority being under 40k miles.
Most of the cars are either petrol or diesel, with 2 electric, 22 hybrids and 1 other (which given it has the same MPG, engine size and model, is almost certain to be an electric Mondeo).
Road tax is clustered around the £150 mark, with another around £0. As road tax is calculated based upon fuel type, engine size and age it may not be especially useful as a predictor.
Miles Per Gallon doesn't seem to follow any particular distribution, but it is mainly centred around 60, with the hybrid cars claiming to be around 200 MPG. The electric cars are claiming 67 (for electric cars it's a meaningless number).
The engine size of most of the cars are clustered around 1 litre, with a few sports cars at the higher end.
From the scatter plots we can see that some of the models have a lot of variability in price and others don't.
For the year we can see that there is something of an exponential relationship or at an extreme push a linear relationship to the price. This might be better as the age of the car as if we were going to be getting more data in the future, the age of the car is what is more important than the actual year (for classic cars some might start gaining in value after a while, but we don't have any evidence for that in this dataset).
From looking at the transmission type it doesn't look as if that is going to be of much use in predicting the price.
The mileage looks like it has an exponentially decreasing relationship with price, with the price capped at zero.
The fuel type looks like an interesting one, as petrol appears useless as a predictor of price, diesel looks like it has a maximum price below that of petrol (likely as there aren't any diesel sports cars). Electric cars seem to be all one specific price but there are only three of them (thanks to moving the one with the other type into electric) so that is likely a lack of data. Finally, hybrids are rather spread out but we only have 22 data points.
The tax against price appears quite an interesting shape, but likely not too useful for predicting anything as it is worked out from various other columns.
The MPG column against price is horribly distorted by the hybrids, it looks like it could be argued to be linear or exponentially decreasing (or some sort of squared relationship if you really wanted to get to the hybrids (the values over 200).
Finally, the engine size column against price looks like it has something of a linear relationship (with a lot of missing points).

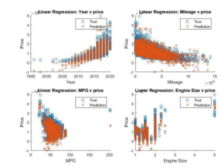


Linear Regression

Linear Regression is a type of algorithm designed to fit an equation of the form:
Y = wX + c + e
Where Y is the target variable, X is the predictor, w is the slope, c is the intercept and e is the error (which we can't model so our prediction should be wrong by that amount).
In this case we would have w and X as vectors for each of the different predictors that we are using.
It tries to generate the w vector and intercept such that:
"The sum-of-squares error is then equal (up to a factor of 1/2) to the squared Euclidean distance between y and t. Thus the least-squares solution for w corresponds to that choice of y that lies in subspace S and that is closest to t" [2]
The computer will iterate through potential solutions until it finds one with the smallest error.

Pros:
- The solution to linear regression is easy to use and can be easily transferred over to other systems
- Using the solution to linear regression is just putting the relevant values for the X vector in and then using the w vector and intercept to find the y value (though you may have to rescale the X vector first)
- A *"way of extending this model is to include a third predictor, called an interaction term, which is constructed by computing the product of X1 and X2"* [6]. This allows the Linear Regression model to capture more complicated relationships such as year times mileage against price.

Cons:
- The solution might not make that much sense to people, as it may attach undue importance to particular variables if every possible variable has been put into the model (e.g. we may have the weather of when the car went to the dealership, which might be found to be useful by the computer)
- Generally the model needs normalised data
- *Important assumption in linear modeling is the assumption of linearity"* [3]

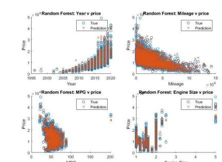


Random Forest Regression

Random Forest Regression combines lots of decision tree regressors and takes their mean result to give the prediction. They use "bagging and random features" [4] when training each decision tree, or in other words they sample with replacement from the training data (bagging) to create artificial training data and then use a random sample of the columns to train from.

Pros:
- *"Because of the Law of Large Numbers they do not overfit"* [4]
- They don't need the data to be normalised for them
- "*The bias of the full model is equivalent to the bias of a single decision tree. The variance of the final model can be greatly reduced over that of a single tree"* [7]

Cons:
- As it is a combination of many decision tree regressor models, the reason why it came up with a particular result can be hard to understand. It is not the black box of a neural network model, as we have access to the underlying trees, but effectively the why is buried in the detail of the trees
- You need to use the full model in production, you can't just pluck the W vector



Hypothesis Statement

Our hypothesis is that the Random Forest Regressor will have better accuracy (lower MAE and RMSE) than Linear Regression in predicting price. This is because the random forest regressor should be able to generalise better to all of the different columns and pick up on smaller patterns. By contrast, linear regression will just try to place a line through the data and not find subtler patterns.

Choice of Training and Evaluation Methodology

We are using Mean Absolute Error for checking the average error and the Root Mean Squared Error to see if there are higher errors (as RMSE penalises high errors more). We are also charting the predictions and real results against each other to see graphically what the errors are like and charting a histogram of the residuals to make sure that they are normally distributed and within a reasonable range. We have also calculated the Normalised Mean Squared Error so that we can see a percentage error.

Choice of parameters and experimental results

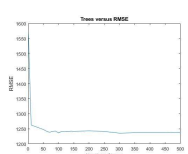
We have chosen to use the columns: model, year, mileage, fuel type, MPG and engine size. This is because all of these have some relationship with the price column and aren't derived from other columns. Using the script OptimizeLinearRegression we found the general parameters for linear regression that gave good results. Normalising the data dramatically improved the performance of linear regression.
The best Linear Regression model has RMSE 1818.02 and MAE of 1344.83.
- Lambda: 0.000010015, the regularisation parameter used to penalise large coefficients.
- Learner: Least Squares. Which algorithm to use, this one uses Least Squares with Mean Squared Error as the loss function.
- Regularisation: Ridge. *Ridge regression helps to minimise the coefficient vector w* [5](paraphrased).
- Solver: BFGS. The objective function being used (Broyden-Fletcher-Goldfarb-Shanno quasi-Newton algorithm)
- Kfold: 5. This trains 5 models and then we picked the best one. (The average prediction is the same as not using the cross-validation).

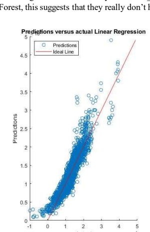
For the Random Forest regressor we used the script OptimiseRandomForest to find the best hyperparameters.
The best Random Forest model has RMSE 1172.39 and MAE 836.57. Even the worst Random Forest Regression model was better than the best Linear Regression model.
- Min leaf size: 1. Minimum number of observations per leaf.
- Method: Bag. Random forest bagging.
- Number of learning cycles: 100. Number of decision tree regressors to use in the forest.

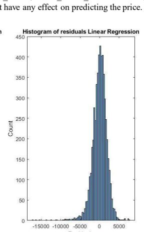
From the graph Trees versus RMSE you can see that there is not much difference between using 100 trees and 500.
Using K-fold cross-validation, 100 trees and taking the average of the results we get almost as good a result as just using 100 trees, but not quite as good as just using 100 trees.

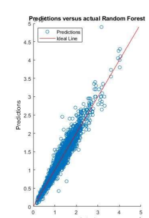
Analysis and critical evaluation of results

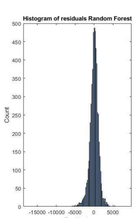
From the RMSE and MAE results we can see that Random Forest has an MAE of £508.3 less than Linear Regression and £645.6 less of an RMSE than Linear Regression. From the 1 - NMSE we can see that Linear Regression is at 0.85 and Random Forest is at 0.94 (I'm not too keen on this as a measure of accuracy, as you could very easily treat it as a percentage accuracy but it doesn't quite emphasise how far away the average prediction could be from its true value). The histogram plots of the two models' residuals show that Random Forest's residuals are centred much tighter on zero (although both suffer from having some residuals very far from zero). In the plots of the continuous variables against the predicted and true prices we can see that generally Random Forest is closer to actual price (admittedly it is easier to spot some of the silly mistakes made by Linear Regression).
The core problem faced by the Random Forest model is that it uses 100 decision trees within its forest. This makes it fairly slow to train. It also means that is effectively a black box. You could drill into all of its constituent trees to find out which features are more or less important as predictors but you would be unlikely to truly understand the model. Once it is trained it is then very quick to use and mostly accurate (94% accurate if 1 – NMSE is to be believed). The question is whether or not you would trust it as a predicting tool without fully understanding it.
For the linear regression model on the other hand you can drill into the model variable in MATLAB and see its Beta variable with all of its coefficients. You can see that columns 18 and 23 (model_Ranger and model_Transit_Tourneo) both have coefficients of 0, so these cars are either perfectly explained by their other attributes or they aren't explained at all. The disadvantage of the model is that it obviously didn't generalise particularly well to the data. Its residuals are spread rather wide and a few of them are very far from £0. The advantage of the model though comes from its explain-ability, even if you wouldn't trust it to predict the price of a car (and the data says that you shouldn't trust it that much). You can see the effect of engine size and other parameters on the price (admittedly these are on the normalised data, but you could translate them back to the unnormalized data).
From the feature importance graphs (whose scales cannot directly be compared), we can see that Random Forest doesn't really use model_Escort, model_ranger, model_streetka, model_transit_Tourneo and engineSize. Linear Regression doesn't really use model_Escort, model_transit_Tourneo and engineSize (and barely uses model_Kuga). As linear regression also doesn't use the same features as Random Forest, this suggests that they really don't have any effect on predicting the price.

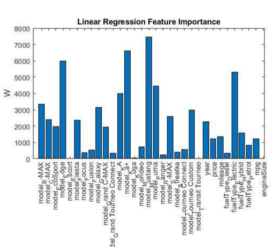


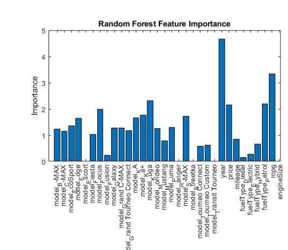












Lessons Learned and Future work

Something we spotted as a potential issue and possible improvement when trying to analyse the models, was that there were a lot of binary columns as to what model a car was. For random forest this is unlikely to be an issue, but could be something that didn't help the Linear Regression model. It might have been better to group the models into categories such as SUV, sports car, people carrier, etc.. This at least would mean that you could train a model on Ford's car data and then take the data for another manufacturer and see how well it predicts their prices (the model may not be aware of the subtlety that a Mercedes car with all the same variables as a Ford car might be perceived as more valuable due to its brand).

References

1. *Car engine sizes: What you need to know | webuyanycar.com* (no date). Available at: https://www.webuyanycar.com/guides/car-ownership/car-engine-sizes (Accessed: 21 November 2021).
2. Bishop, Christopher (2006) 'Pattern Recognition and Machine Learning', in *Pattern Recognition and Machine Learning*. Chapter 3: Springer Science+Business Media LLC. Available at: https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf (Accessed: 30 November 2021).
3. Osborne, J.W. (2017) *Regression & Linear Modeling: Best Practices and Modern Methods*. 2455 Teller Road, Thousand Oaks California 91320: SAGE Publications, Inc. doi:10.4135/9781071802724.
4. Schapire Robert (2001) *Random Forests*, *Random Forests*. Available at: https://link.springer.com/content/pdf/10.1023/A:1010933404324.pdf (Accessed: 1 December 2021).
5. Bhattacharyya, Saptashwa. 'Ridge and Lasso Regression: L1 and L2 Regularization'. Medium, 28 September 2020. https://towardsdatascience.com/ridge-and-lasso-regression-a-complete-guide-with-python-scikit-learn-e20e34bcbf0b.
6. James, Gareth, Witten, Daniela, Hastie, Trevor, and Tibshirani, Robert. *An Introduction to Statistical Learning with Applications in R*. 2nd ed. Vol. 1. 1 vols. Chapter 3: Springer Science+Business Media LLC, 2021. https://ebookcentral.proquest.com/lib/city/reader.action?docID=6686746. Page 88.
7. *Understanding the Bias-Variance Tradeoff* (no date). Available at: https://scott.fortmann-roe.com/docs/BiasVariance.html (Accessed: 9 December 2021).