Predicting Customer Pricing

Daniel Piddock

April 4, 2024

Abstract

This report presents an analysis of car insurance pricing using a dataset comprising of 50,000 quotes sourced from the Hastings Direct pricing database. The objective is to construct a predictive model to determine appropriate pricing for customers. Python was utilised for data exploration, visualisation, and model development. Various outlier removal techniques were employed, and 12 initial models were tested. Ensemble methods were explored, with the ensemble model comprising of a Generalised Additive Model and an XGBoost (Extreme Gradient Boosting) model, yielding the best performance of all models. The report concludes with insights into model accuracy and areas for potential improvement.

1 Introduction

With a dataset comprising of 50,000 quotes for motor insurance sourced from the Hastings Direct pricing database, my objective is to go through exploration and analysis of the data. The primary aim is to construct a predictive model capable of determining appropriate pricing for customers. Using Python, I dissected the dataset to uncover meaningful insights, and engineer a predictive model.

Throughout this task, I delved into a multitude of libraries and models available within Python, seeking to optimise and refine my approach continuously. I aimed to leverage Python's capabilities to develop a predictive framework. This framework not only captures the dataset's complexities but also provides insights for informed decision-making in car insurance pricing.

2 Data Exploration

Upon reviewing the Excel file, several interesting immediate observations came to light. Firstly, there appeared to be missing values in the "Driver2 Licence Type" and "Driver2 Licence Years" columns, presenting a challenge for analysis. Additionally, anomalies such as "-9999" values were detected in unexpected places, such as the "Driver1 Convictions" column, where "Yes" or "No" entries were expected, and in columns like "Vehicle Value" and "Tax," where only positive values would be appropriate.

Further investigation revealed inconsistencies in the capitalisation of some "Driver1 Marital Status" entries. To ensure uniformity and avoid indexing issues, all entries were standardised to uppercase. Additionally, I indexed the "Driver1 Licence Type," "Driver1 Convictions," and "Payment Type" columns for further analysis and modeling.

Furthermore, it was noted that all quotes were dated in 2020. To streamline the data, I transformed the "Quote Date" column to reflect only the month of the quote. Lastly, I converted the "Driver1 DOB" column from the "dd/mm/yyyy" format to "Driver Age," enhancing the data's usability for analysis.

In my pursuit of developing a robust predictive model for car insurance pricing, I employed an Isolation Forest method to clean my data to create a dataset to make plots with to ensure variables integrity and suitability for subsequent modeling.

My analysis utilised a variety of visualisations as seen in Figure 1, including a correlation heatmap and an array of graphical representations. Through these visual aids, I discerned the most influential variables, for inclusion in my model from those that were not suitable. Notably, variables such as Vehicle Value, Vehicle Age, Years Having Licence, Tax, Driver Age, Days to Inception, and Payment Type emerged as pivotal contributors to my predictive framework as they have a clear and significant correlation with Premium Price.

Deliberate exclusion of certain variables, such as Quote Date, Driver1 Licence Type, Driver1 Convictions, Driver1 Claims, Vehicle Annual Mileage, and Driver1 Marital Status, the decision to exclude certain variables was based on evidence gathered from my analysis visualised in Figure 2. My scrutiny of the correlation heatmap and box plots revealed little to no correlation or pattern to many of the aforementioned variables, guiding my decision-making process. For instance, while variables like Driver Age and Days to Inception exhibited clear correlations with premium prices, others such as Quote Date and Marital Status displayed negligible impact, thereby warranting omission.

I carefully chose which features to include in the model, considering the risks of overfitting and reduced accuracy. I aimed for a balance between having enough information and keeping the model simple to avoid these risks.

3 Modelling

In the initial phase of modeling, priority was given to data cleaning. Employing a multitude of outlier detection methods including: Isolation Forest, Local Outlier Factor, Median Absolute Deviation, Interquartile Range, and z-score normalisation. The cleaning methods aimed to purge extreme values and enhance the suitability of the dataset for modeling.

Following examination of variables identified through data exploration, twelve initial models were employed in Python. These models had a diverse range, including ElasticNet, XGBoost, Linear Regression, Bayesian Regression, Random Forest Regressor, Generalised Additive Models (GAM), K-Nearest Neighbors Regression (KNN), AdaBoost Regression, Lasso Regression, Ridge Regression, and Decision Tree Regression.

Through a comparison of performance metrics such as coefficient of determination (R^2),

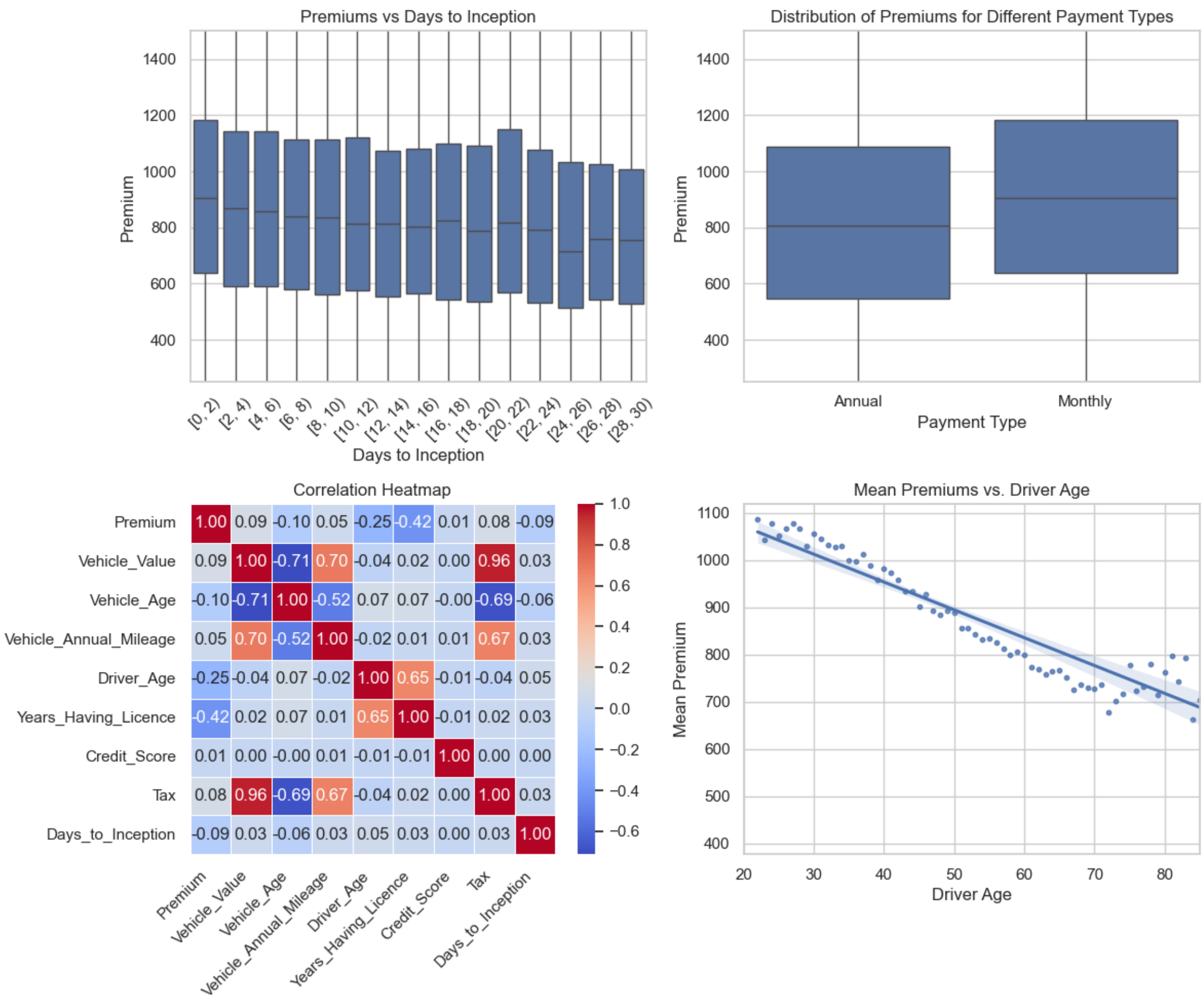

Figure 1: Correlation heatmap, boxplots and a regression plot

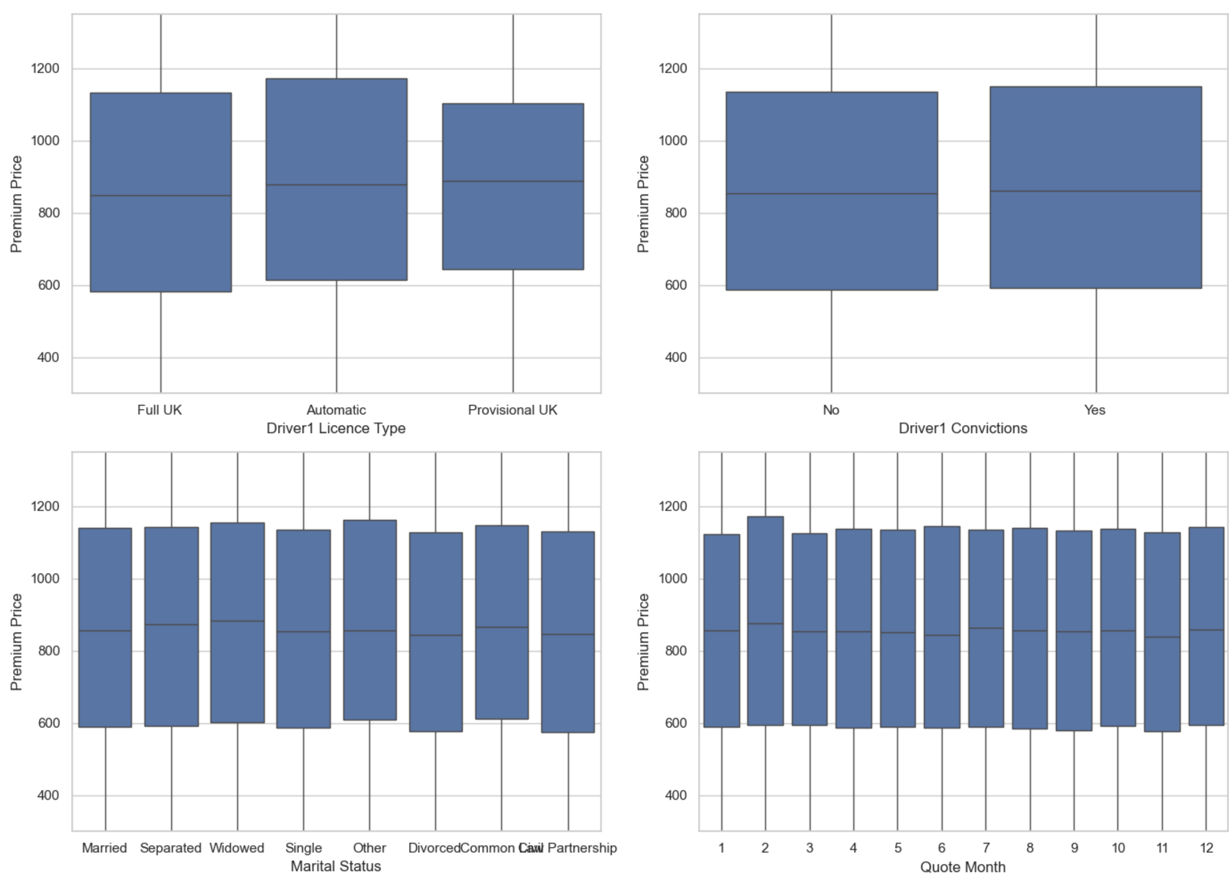

Figure 2: Boxplots showing a few variables vs premium price

Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE), the efficacy of each model was evaluated. Leveraging five cleaned datasets, an iteration process was undertaken to capture the performance of each model across the different cleaned datasets. The resulting compilation of R^2, MSE, RMSE, and MAE values can be seen in Table 1.

Subsequently, an exploration of ensemble modeling techniques was used to utilise the predictive power of multiple models. Commencing with exploration of various ensemble methods, my trial and error ended in the creation of an ensemble model that combined the top-performing individual models. Notably, a combination of GAM and XGBoost emerged as the most promising ensemble, although it has marginal improvements over the individual models.

The GAM model I used in python imported from pyGAM, assume that the relationship between the Premium Price and predictors is additive, allowing for non-linear relationships through the use of smooth functions. Additionally, GAMs assume that predictors contribute independently to the outcome, enabling automatic feature selection and regularisation to control model complexity. On the other hand, XGBoost models, a gradient boosting algorithm, do not rely on specific assumptions about the data distribution. The XGBoost model is itself a form of ensemble of weak learners which are usually decision trees which when combined, create a good predictive model.

4 Results and Conclusions

In this study, I explored many different ensemble models comprising of two or more of my previous models. Out of the ensemble modeling techniques the best was a combination of the Generalised Additive Model (GAM) and the XGBoost model to predict premium pricing for different drivers based on my datasets. Interestingly, the ensemble model consistently emerged as the top performer out of all my models in terms of R^2 value, achieving a maximum of 0.25199 across the datasets and in terms of MSE, RMSE, and MAE giving values of 119805.35002, 346.12909, and 276.85575 respectively.

Upon closer examination, it became evident that the choice of dataset cleaning method influenced the performance metrics of the models. Specifically, the Local Outlier Factor method yielded the highest R^2 value, while the Isolation Forest method resulted in the lowest Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) as demonstrated by my findings in Table 1. This highlights the importance of meticulous data preprocessing in enhancing model accuracy.

Despite the relatively modest performance metrics, it is noteworthy that my models exhibited a level of predictive capability, considering the constraints of limited data availability and variables for analysis. These findings suggest that while my models may not achieve perfect accuracy, they still offer valuable insights into premium pricing within car insurance.

Furthermore, when comparing model predictions with actual quotes obtained from the Hastings Direct website, discrepancies were observed, particularly for younger drivers. This discrepancy can be attributed to the absence of data for drivers under the age of 24 in my dataset, limiting the model's ability to accurately predict premiums for this demographic. Nevertheless, when inputting data for older drivers, such as my father's information, the model's predictions closely aligned with the obtained quotes, the premier policy being quoted at £596 on the website and being predicted as £602 in my model, indicating a degree of reliability.

Tables 2 and 3 demonstrate some possible values of customers and a premium price prediction based off my two different cleaned datasets and my best model based off the two different metrics I was measuring. Personally I believe the dataset which gives the better R^2 value is likely to be a more accurate model for car insurance premiums however, they both give similar results for the most part.

Moving forward, it is essential to acknowledge the limitations of my study and identify areas for future research. While my models provide a foundation for predicting premium pricing, several factors such as the presence of a black box option (YouDrive option), vehicle specifications, driving history, geographic location, and policy coverage details were not incorporated in the dataset given for this case study. Incorporating these factors into future analyses could further enhance the accuracy and applicability of my predictive models.

In conclusion, while my models demonstrate promise in predicting premium pricing for insurance policies, there is room for improvement. By addressing data limitations, refining modeling techniques, and incorporating additional relevant variables, I could develop more robust and accurate models to inform insurance pricing strategies and decision-making processes.

Model	Cleaning Method	R^2	MSE	RMSE	MAE
ElasticNet	None	0.000102394	1722389.524	1312.398386	462.9369022
XGBoost	None	0.001460813	1720049.557	1311.506598	457.9044301
Linear Regression	None	8.14E-05	1722425.753	1312.412189	462.8803999
Bayesian Regression	None	0.000210172	1722203.868	1312.327653	464.1222677
Random Forest Regressor	None	-0.052472982	1812954.074	1346.45983	502.8193635
GAM	None	0.007256187	1710066.644	1307.695165	455.3638255
KNN Regression	None	-0.162672764	2002780.46	1415.196262	548.9583435
AdaBoost Regression	None	-0.003373196	1728376.457	1314.67732	463.0386291
Lasso Regression	None	8.41E-05	1722421.079	1312.410408	462.8864307
Ridge Regression	None	8.14E-05	1722425.728	1312.412179	462.8804289
Decision Tree Regression	None	0.007355863	1709894.944	1307.629513	456.9253095
ElasticNet	Z Score	0.216879476	131943.9121	363.2408458	288.3408104
XGBoost	Z Score	0.231192993	129532.3018	359.9059625	287.0618327
Linear Regression	Z Score	0.217171438	131894.721	363.1731281	288.233184
Bayesian Regression	Z Score	0.217158347	131896.9265	363.1761645	288.2425433
Random Forest Regressor	Z Score	0.161013673	141356.4512	375.9740033	299.834314
GAM	Z Score	0.233599024	129126.9222	359.3423467	285.9084128
KNN Regression	Z Score	0.019276194	165237.0633	406.4936202	324.9586258
AdaBoost Regression	Z Score	0.151359135	142983.0942	378.1310543	307.6554715
Lasso Regression	Z Score	0.21716977	131895.0019	363.1735149	288.2352272
Ridge Regression	Z Score	0.217171389	131894.7292	363.1731394	288.2332377
Decision Tree Regression	Z Score	0.210405323	133034.7084	364.7392335	290.4333179
ElasticNet	IQR	0.226355186	128876.4389	358.9936475	287.7254723
XGBoost	IQR	0.240201947	126569.7974	355.7664929	286.1804034
Linear Regression	IQR	0.227098621	128752.5949	358.8211182	287.5006073
Bayesian Regression	IQR	0.226974427	128773.2836	358.8499457	287.542312
Random Forest Regressor	IQR	0.193409533	134364.6402	366.5578265	294.9229586
GAM	IQR	0.242233947	126231.2998	355.2904443	285.2092643
KNN Regression	IQR	0.034905303	160768.8251	400.9598796	322.2787879
AdaBoost Regression	IQR	0.183950965	135940.2812	368.7008018	300.9416143
Lasso Regression	IQR	0.227079734	128755.7411	358.8255023	287.5075286
Ridge Regression	IQR	0.227098001	128752.6982	358.8212622	287.5008224
Decision Tree Regression	IQR	0.21202387	131263.7993	362.303463	291.1476275
ElasticNet	MAD	0.113129754	168663.0818	410.6861111	305.7541937
XGBoost	MAD	0.137090266	164106.3231	405.1003864	303.836343
Linear Regression	MAD	0.113115411	168665.8097	410.6894322	305.7490663
Bayesian Regression	MAD	0.113196638	168650.362	410.6706247	305.7930178
Random Forest Regressor	MAD	0.071323501	176613.7055	420.2543343	318.3056478
GAM	MAD	0.142911433	162999.2661	403.7316758	302.3854208
KNN Regression	MAD	-0.06887699	203276.7343	450.8622121	345.1866106
AdaBoost Regression	MAD	-0.043209673	198395.3788	445.4159616	354.5252669
Lasso Regression	MAD	0.113123688	168664.2355	410.6875157	305.7489498
Ridge Regression	MAD	0.113115426	168665.8067	410.6894286	305.7490742
Decision Tree Regression	MAD	0.13271952	164937.5422	406.1250327	304.3061347

ElasticNet	LOF	0.228990418	131843.2167	363.1022125	290.3728156
XGBoost	LOF	0.250106879	128232.2859	358.0953587	286.4872106
Linear Regression	LOF	0.228823994	131871.6754	363.1413986	290.3539689
Bayesian Regression	LOF	0.228866605	131864.3887	363.1313657	290.3516379
Random Forest Regressor	LOF	0.196208353	137448.9743	370.741115	296.067221
GAM	LOF	0.250119886	128230.0617	358.0922531	286.2742999
KNN Regression	LOF	0.037923148	164515.8631	405.6055511	324.1177036
AdaBoost Regression	LOF	0.087339252	156065.6722	395.0514805	324.4906292
Lasso Regression	LOF	0.228832344	131870.2475	363.1394325	290.354499
Ridge Regression	LOF	0.228824233	131871.6345	363.1413423	290.353952
Decision Tree Regression	LOF	0.224580417	132597.3301	364.1391631	291.7231218
ElasticNet	Isolation Forest	0.191965771	123043.3951	350.7754198	279.7683482
XGBoost	Isolation Forest	0.203045709	121356.1978	348.3621647	278.592706
Linear Regression	Isolation Forest	0.192490222	122963.5342	350.6615665	279.6764259
Bayesian Regression	Isolation Forest	0.192429351	122972.8034	350.674783	279.6868062
Random Forest Regressor	Isolation Forest	0.13554353	131635.0907	362.815505	289.0911748
GAM	Isolation Forest	0.20298356	121365.6615	348.3757475	278.3532021
KNN Regression	Isolation Forest	-0.002394107	152639.5413	390.6911072	311.1466593
AdaBoost Regression	Isolation Forest	0.154208861	128792.8277	358.8771764	291.0862441
Lasso Regression	Isolation Forest	0.1924795	122965.1669	350.6638945	279.6788855
Ridge Regression	Isolation Forest	0.192489965	122963.5735	350.6616224	279.6764678
Decision Tree Regression	Isolation Forest	0.182244096	124523.7629	352.8792469	282.5122453
Ensemble (XGBoost + GAM)	None	0.00637786	1711579.62	1308.273527	455.1001365
Ensemble (XGBoost + GAM)	Z Score	0.235217831	128854.1777	358.9626411	285.914036
Ensemble (XGBoost + GAM)	IQR	0.244369484	125875.5546	354.7894511	285.0572294
Ensemble (XGBoost + GAM)	MAD	0.143641731	162860.3796	403.5596357	302.4388871
Ensemble (XGBoost + GAM)	LOF	0.251993021	127909.7544	357.6447321	286.0689043
Ensemble (XGBoost + GAM)	Isolation Forest	0.222357272	119805.35	346.1290944	276.8557513

Table 1: Every model evaluated with every cleaned dataset

Vehicle Value (£)	Vehicle Age	Driver Age (Years)	Years Having Licence	Days to Inception	Tax (£)	Payment Type	Premium Predictions (£)
1500	13	20	1	14	150	Annual	993.76
2250	10	25	6	10	225	Annual	934.48
3500	8	30	10	25	350	Monthly	909.27
5000	6	35	15	1	500	Annual	882.18
8000	4	40	19	7	800	Annual	805.23
10000	4	45	24	11	1000	Monthly	799.61
13500	4	50	28	3	1350	Annual	775.74
16000	3	55	33	22	1600	Annual	703.99
18000	2	60	38	10	1800	Annual	692.44
20000	1	65	40	28	2000	Annual	708.68

Table 2: GAM and XGBoost ensemble model results evaluated with Isolation Forest cleaning method (Better MSE, RMSE, and RAE values)

Vehicle Value (£)	Vehicle Age (Years)	Driver Age (Years)	Years Having Licence	Days to Inception	Tax (£)	Payment Type	Premium Predictions (£)
1500	13	20	1	14	150	Annual	993.76
2250	10	25	6	10	225	Annual	934.48
3500	8	30	10	25	350	Monthly	909.27
5000	6	35	15	1	500	Annual	852.61
8000	4	40	19	7	800	Annual	813.75
10000	4	45	24	11	1000	Monthly	787.2
13500	4	50	28	3	1350	Annual	728.31
16000	3	55	33	22	1600	Annual	683.66
18000	2	60	38	10	1800	Annual	649.9
20000	1	65	40	28	2000	Annual	621.46

Table 3: GAM and XGBoost ensemble model results evaluated with Local Outlier Factor cleaning method (Better R^2 value)