Studying Auto Insurance Data

Ashutosh Nandeshwar

February 23, 2010

1 Introduction

To study auto insurance data using traditional and non-traditional tools, I downloaded a well-studied data from `http://www.statsci.org/data/general/motorins.html`, which was originally studied in Hallin and Ingenbleek (1983); Andrews and Herzberg (1985). These data consisted of third party motor insurance claims in Sweden for the year 1977. "In Sweden all motor insurance companies apply identical risk arguments to classify customers, and thus their portfolios and their claims statistics can be combined. The data were compiled by a Swedish Committee on the Analysis of Risk Premium in Motor Insurance. The Committee was asked to look into the problem of analyzing the real influence on claims of the risk arguments and to compare this structure with the actual tariff."

According to the data description, "the number of claims in each category can be treated as Poisson to a good approximation. The amount of each claim can be treated as gamma. The total payout is therefore compound Poisson." In my cursory literature search, I found that these data were studied using decision-trees and combination of regression techniques (Andrews and Herzberg 1985; Christmann 2006; Loh 2008; Marin-Galiano and Christmann 2004). For this project, I studied the total payment as a dependent variable. All the variables in these data are listed in Table 1, and the descriptive statistics of total payment are given in Table 2, and the histogram is given in Figure 1.

2 Data Exploration

Figure 1 shows the average payments by Kilometers driven and the zones in which the cars were driven. The size of the circle depicts the size of the payments. Kilometer traveled per year ranges 1 (<1000) and 2 (1000-15000) stand out for the average payments and also the size of the payments, especially 1000-15000 category. Cars driven in Zone 4 also had high payments regardless of the kilometers driven.

As seen in Figure 2, it is clear that make 1 had very high total payments across all bonus types. Also, make 6 had high total payments. Payments were high in the seventh bonus year and the first bonus year.

Variable	Values
Kilometres	Kilometres travelled per year
	1: $<$ 1000 2: 1000-15000 3: 15000-20000 4: 20000-25000 5: $>$ 25000
Zone	Geographical zone
	1: Stockholm, Gteborg, Malm with surroundings 2: Other large cities with surroundings 3: Smaller cities with surroundings in southern Sweden 4: Rural areas in southern Sweden 5: Smaller cities with surroundings in northern Sweden 6: Rural areas in northern Sweden 7: Gotland
Bonus	No claims bonus. Equal to the number of years, plus one, since last claim
Make	1-8 represent eight different common car models. All other models are combined in class 9
Insured	Number of insured in policy-years
Claims	Number of claims
Payment	Total value of payments in Skr

Table 1: Data Description

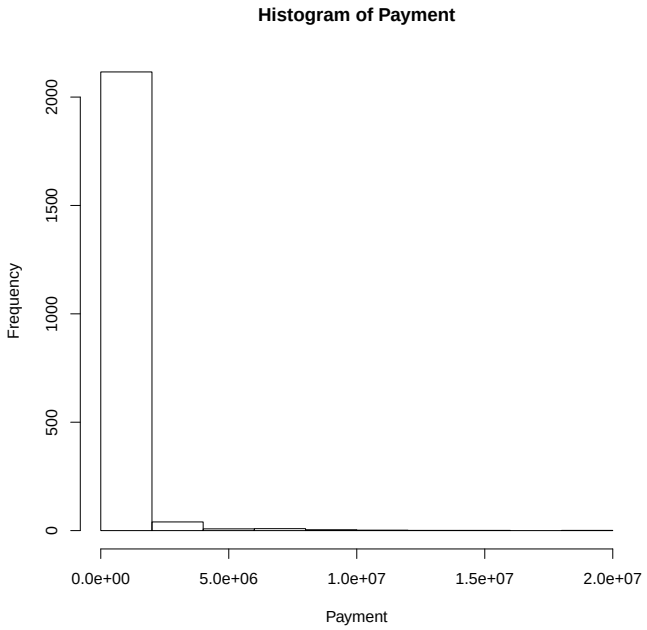
3 Experiment

After experimenting with some numeric-class learners, I ran the following learners in Weka (Hall et al. 2009):

1. Linear regression
2. Neural network
3. Additive regression with DecisionStump: a boosting technique to improve the robustness and the accuracy of the base learner; this is achieved by selecting a random sub-sample of the training data at each iteration and the model is updated using the values from this sample (Friedman 2002).
4. M5P: a decision tree learning technique specially designed for continuous class variable that applies multivariate linear regression at each node of the tree (Quinlan 1992; Wang and Witten 1997).
5. IBk: an instance based learner that selects and stores some (similar) instances of the data using nearest neighbor technique (Aha et al. 1991).
6. Bagging with REPTree: bagging repeats the learning algorithm n times and take an average of the performance measures over n versions. Boot-

Statistic	Value
min	0
max	18245026
sd	1017283
avg	257007.6
count	2182

Table 2: Descriptive Statistic of Payment Variable



strap sampling is used to generate n datasets. It has been shown that bagging can improve the accuracy of the base learner (Breiman 1996). REPTree is a fast decision tree learner.

7. Bagging with linear regression

Each learner was trained on 10 cross-validation folds, meaning that the data were split into 10 sets, and trained on 9 of them and tested on the 10th one. This procedure was repeated 10 times. Therefore, 10 result sets were generated for each algorithm, and the complete experiment was re-randomized and repeated 10 times resulting in 700 sets for seven learners.

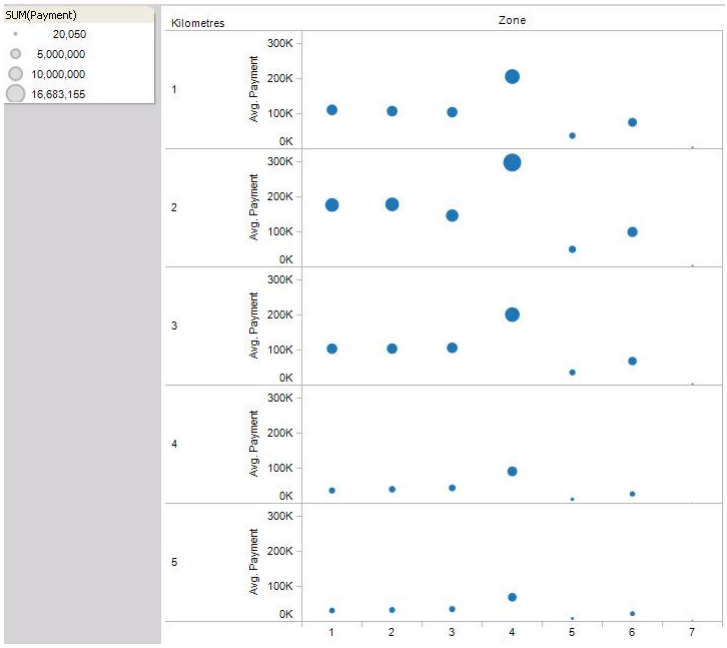


Figure 1: Kilometers Driven, Zone and Payments Occurred (the size of the circle denotes the payments occurred for that combination). It is clear that Zone number 4 and Kilometers range 1 and 2 had the highest payouts.

4 Results

The average of 10 *times* 10 run of mean absolute error is given in Table 3 and in Figure 3, and the average of root mean square error for these 10 runs is shown in Figure 4. After examining the results of mean absolute error and root mean squared error, it became clear that M5P and neural network performed much better than the other learners. Since neural network results are incomprehensible, I am including the rules generated using M5P given in Appendix A.

In addition, Weka currently does not offer a way to apply generalized linear models to the data, therefore, I ran a GLM learner using Poisson distribution and a log link function in SPSS's Clementine software and obtained a mean absolute error of 75,173 and a correlation coefficient of 0.985 on the test dataset. That means that GLM performed much better than the state-of-the art techniques for this specific dataset.

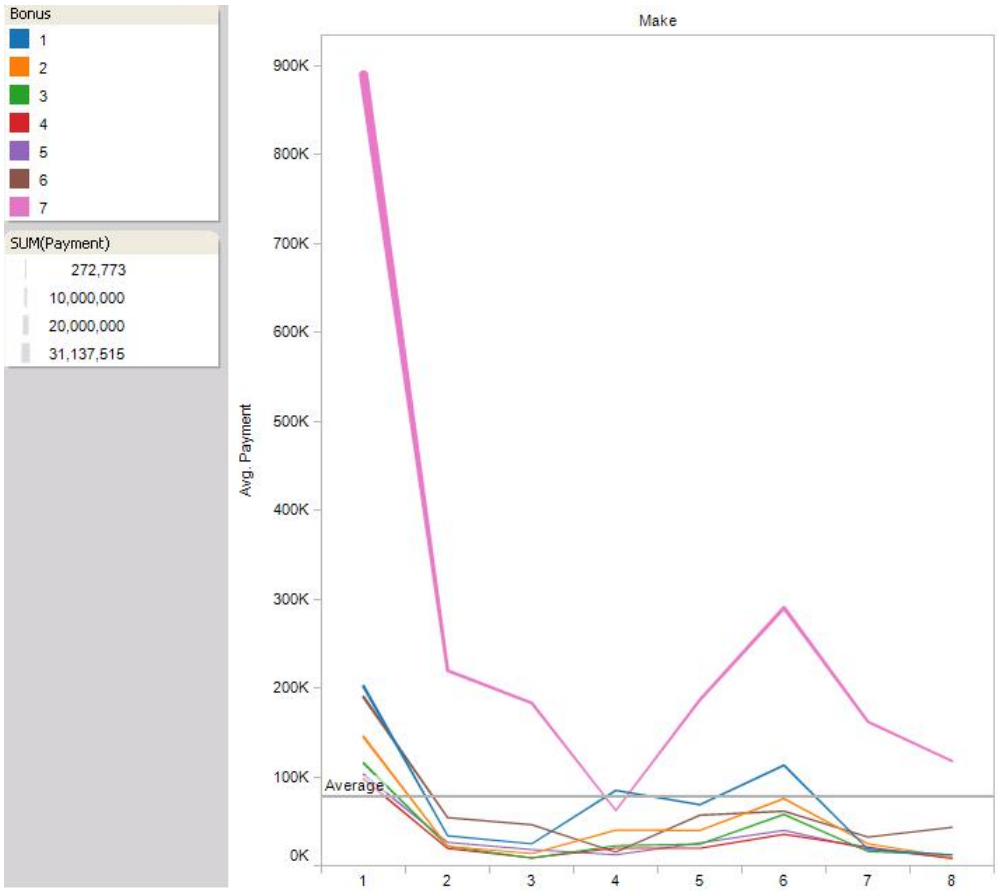


Figure 2: Make vs Avg. Payment. The pink line, which has the highest payouts across all makes, denotes the average payouts by the make of the car

Table 3: Mean Absolute Errors Obtained by Learners

Learner	Mean Absolute Error
M5P	101638.03
Neural Net	114390.41
IBk	167871.21
Bagging REPTree	172455.61
AdditiveRegression	333352.97
Bagging Linear Regression	344290.80
Linear Regression	346286.53

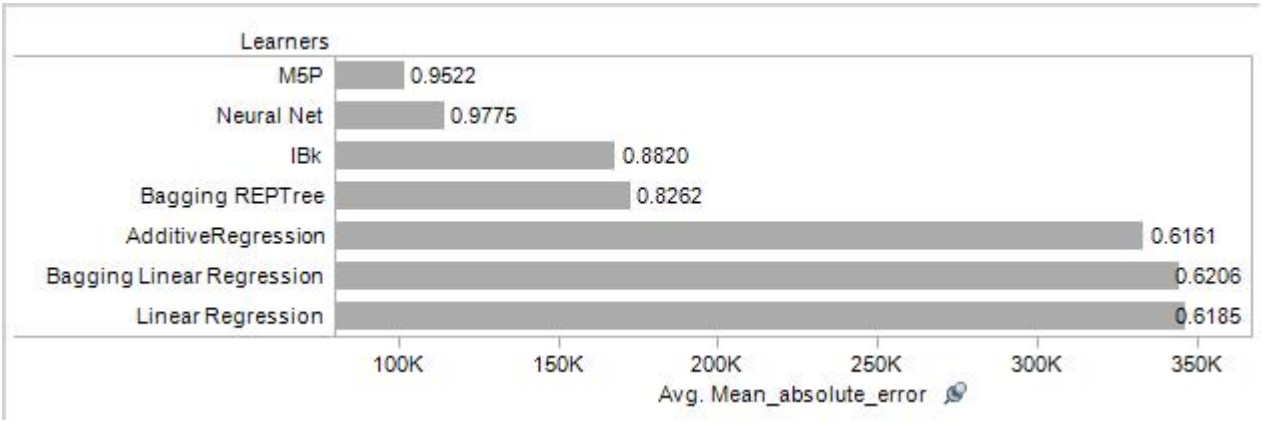


Figure 3: Mean Absolute Errors and Correlation Coefficients, which is the text on the right

Table 4: Correlation Coefficients Obtained by Learners

Learner	Correlation coefficient
AdditiveRegression	0.62
Linear Regression	0.62
Bagging Linear Regression	0.62
Bagging REPTree	0.83
IBk	0.88
M5P	0.95
Neural net	0.98

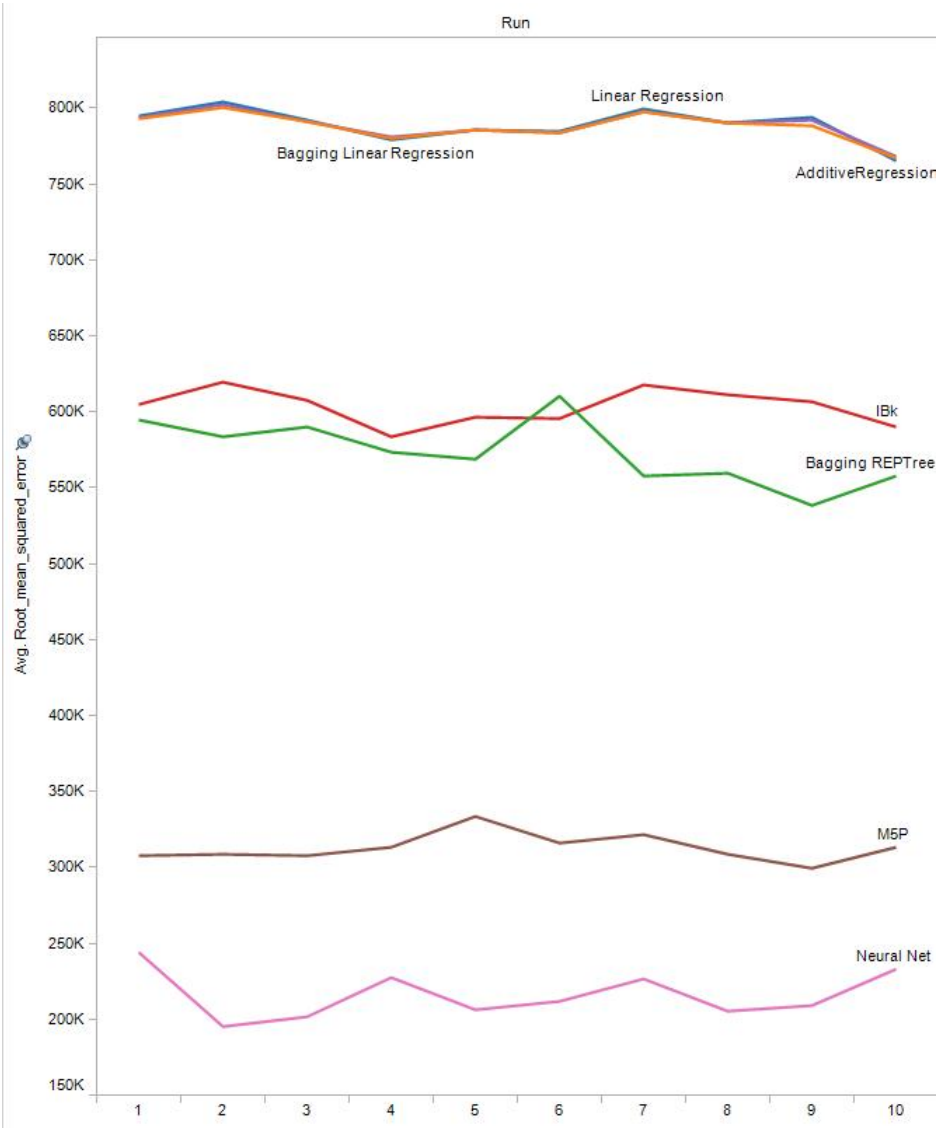


Figure 4: Average of Root Mean Squared Error Over 10 runs

5 Conclusion

Sweeden data (Hallin and Ingenbleek 1983; Andrews and Herzberg 1985) was analyzed using data mining techniques and linear regression and generalized linear models. In the state-of-the art techniques M5P, a model tree generator, and neural network performed the best; however, GLM with Poisson distribution and logit link function was the top performer with a mean absolute error of 75,173 and a correlation coefficient of 0.985.

Although this study covered some of the well-known data mining techniques, this study was not an all inclusive exercise. In addition, I ran some learners by discretizing the output (*Payment*) variable, however, their performance was very poor. Some detailed investigation is needed to find machine learning learners (after discretizing the class variable) that will work well with this kind of data. One novel approach specifically for optimizing premium pricing using mathematical programming and data mining is given by Yeo et al. (2002).

References

Aha, D., Kibler, D., and Albert, M. (1991). Instance-based learning algorithms. *Machine learning*, 6(1):37–66.

Andrews, D. and Herzberg, A. (1985). *Data: a collection of problems from many fields for the student and research worker*. Springer Verlag.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

Christmann, A. (2006). Empirical Risk Minimization For Car Insurance Data.

Friedman, J. (2002). Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 38(4):367–378.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11. /urlhttp://www.cs.waikato.ac.nz/ml/weka/.

Hallin, M. and Ingenbleek, J. (1983). The Swedish automobile portfolio in 1977: A statistical study. *Scandinavian actuarial journal*, 83:49–64.

Loh, W. (2008). Regression by Parts: m. 7 Fitting Visually Interpretable Models with GUIDE. *Handbook of Data Visualization*, page 447.

Marin-Galiano, M. and Christmann, A. (2004). Insurance: an R-Program to Model Insurance Data.

Quinlan, J. (1992). Learning with continuous classes. In *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*, pages 343–348. Citeseer.

Wang, Y. and Witten, I. (1997). Induction of model trees for predicting continuous classes. In *Proceedings of the poster papers of the European Conference on Machine Learning*, pages 128–137.

Yeo, A., Smith, K., Willis, R., and Brooks, M. (2002). A mathematical programming approach to optimise insurance premium pricing within a data mining framework. *The Journal of the Operational Research Society*, 53(11):1197–1203.

A Appendix

```
=== Run information ===

Scheme:       weka.classifiers.rules.M5Rules -M 4.0
Relation:     motorins-weka.filters.unsupervised.attribute.NumericToNominal-R1-4-weka.filte
Instances:    2182
Attributes:   5
              Kilometres
              Zone
              Bonus
              Make
              Payment
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

M5 pruned model rules
(using smoothed linear models) :
Number of Rules : 20

Rule: 1
IF
Make=1,9 <= 0.5
Bonus=1,7 <= 0.5
THEN

Payment =
2062.2942 * Kilometres=3,1,2
+ 879.1224 * Kilometres=1,2
+ 965.6121 * Kilometres=2
+ 960.8203 * Zone=5,6,3,2,1,4
+ 1133.1997 * Zone=6,3,2,1,4
+ 1544.2568 * Zone=3,2,1,4
+ 2558.4693 * Zone=4
+ 661.0802 * Bonus=2,6,1,7
+ 1047.1891 * Bonus=1,7
+ 6380.3711 * Bonus=7
+ 132.4676 * Make=7,3,2,5,6,1,9
+ 198.6624 * Make=2,5,6,1,9
+ 454.8036 * Make=6,1,9
+ 1782.3811 * Make=1,9
+ 12542.9287 * Make=9
+ 22216.9042 [1210/4.383%]

Rule: 2
```

```
IF
Make=9 <= 0.5
Bonus=7 <= 0.5
Kilometres=3,1,2 <= 0.5
THEN

Payment =
17369.829 * Kilometres=3,1,2
+ 6168.2674 * Kilometres=1,2
+ 10946.147 * Kilometres=2
+ 7100.0847 * Zone=5,6,3,2,1,4
+ 10146.439 * Zone=6,3,2,1,4
+ 11182.034 * Zone=3,2,1,4
+ 21572.0506 * Zone=4
+ 8241.2843 * Bonus=2,6,1,7
+ 6171.4447 * Bonus=6,1,7
+ 30572.7495 * Bonus=1,7
+ 17972.7481 * Bonus=7
+ 5428.5548 * Make=4,7,3,2,5,6,1,9
- 3997.7826 * Make=7,3,2,5,6,1,9
+ 1646.247 * Make=2,5,6,1,9
+ 3459.199 * Make=5,6,1,9
+ 5240.9045 * Make=6,1,9
+ 43124.6256 * Make=1,9
+ 28855.4977 * Make=9
- 62781.6493 [175/3.012%]

Rule: 3
IF
Make=9 <= 0.5
Make=6,1,9 > 0.5
Bonus=7 <= 0.5
Zone=6,3,2,1,4 > 0.5
THEN

Payment =
83564.9722 * Kilometres=3,1,2
+ 5100.5057 * Kilometres=1,2
+ 164448.1307 * Kilometres=2
+ 19973.5569 * Zone=5,6,3,2,1,4
+ 36082.0501 * Zone=6,3,2,1,4
+ 85373.8316 * Zone=3,2,1,4
+ 228597.1964 * Zone=4
+ 87568.7132 * Bonus=2,6,1,7
+ 46994.1342 * Bonus=6,1,7
+ 135272.8005 * Bonus=1,7
```

```
+ 90448.5076 * Bonus=7
+ 3505.1909 * Make=2,5,6,1,9
+ 5768.8806 * Make=6,1,9
+ 243006.852 * Make=1,9
+ 48540.3176 * Make=9
- 369429.7342 [105/6.361%]

Rule: 4
IF
Make=1,9 <= 0.5
Zone=6,3,2,1,4 <= 0.5
THEN

Payment =
49371.3525 * Kilometres=3,1,2
+ 12181.7521 * Kilometres=1,2
+ 4868.3924 * Kilometres=2
+ 10038.8468 * Zone=5,6,3,2,1,4
+ 9413.014 * Zone=6,3,2,1,4
+ 24233.9366 * Zone=3,2,1,4
+ 43126.9666 * Zone=4
+ 15100.6121 * Bonus=2,6,1,7
+ 53969.3973 * Bonus=1,7
+ 52878.0107 * Bonus=7
+ 3839.4033 * Make=7,3,2,5,6,1,9
+ 4373.0692 * Make=2,5,6,1,9
+ 10164.6661 * Make=6,1,9
+ 50785.7597 * Make=1,9
+ 76970.2504 * Make=9
- 141985.3333 [111/2.829%]

Rule: 5
IF
Make=1,9 <= 0.5
Bonus=7 > 0.5
Kilometres=3,1,2 > 0.5
Make=7,3,2,5,6,1,9 > 0.5
THEN

Payment =
83668.6414 * Kilometres=3,1,2
- 11547.4021 * Kilometres=1,2
+ 131362.3151 * Kilometres=2
+ 19971.2863 * Zone=5,6,3,2,1,4
+ 36860.5857 * Zone=6,3,2,1,4
+ 121092.4411 * Zone=3,2,1,4
```

```
- 44560.9272 * Zone=1,4
+ 363404.3909 * Zone=4
+ 21086.9262 * Bonus=2,6,1,7
+ 65495.3231 * Bonus=1,7
+ 71668.6882 * Bonus=7
- 7853.6803 * Make=4,7,3,2,5,6,1,9
+ 41841.3076 * Make=7,3,2,5,6,1,9
+ 43475.0866 * Make=2,5,6,1,9
+ 182027.4453 * Make=6,1,9
+ 100066.2695 * Make=1,9
+ 95623.5933 * Make=9
- 193571.8791 [75/6.122%]

Rule: 6
IF
Make=1,9 <= 0.5
THEN

Payment =
95620.7445 * Kilometres=3,1,2
+ 80423.2915 * Kilometres=1,2
+ 27277.8545 * Zone=5,6,3,2,1,4
+ 43695.0537 * Zone=6,3,2,1,4
+ 37302.8174 * Zone=3,2,1,4
+ 139610.1911 * Zone=4
+ 28801.6553 * Bonus=2,6,1,7
+ 70676.8275 * Bonus=1,7
+ 176374.8644 * Bonus=7
+ 1554.7772 * Make=7,3,2,5,6,1,9
+ 61986.0128 * Make=3,2,5,6,1,9
+ 91678.5381 * Make=1,9
+ 143343.8052 * Make=9
- 340687.9025 [190/4.75%]

Rule: 7
IF
Bonus=1,7 <= 0.5
Zone=6,3,2,1,4 > 0.5
Kilometres=3,1,2 > 0.5
THEN

Payment =
291190.1812 * Kilometres=3,1,2
+ 241423.2941 * Kilometres=1,2
+ 480375.2686 * Kilometres=2
+ 49219.8575 * Zone=5,6,3,2,1,4
```

```
+ 25847.1227 * Zone=6,3,2,1,4
+ 784684.3875 * Zone=3,2,1,4
+ 246864.2942 * Zone=1,4
+ 602288.5238 * Zone=4
+ 259154.1713 * Bonus=3,2,6,1,7
+ 553633.2289 * Bonus=2,6,1,7
+ 53262.2639 * Bonus=6,1,7
+ 43907.4521 * Bonus=1,7
+ 146884.6189 * Bonus=7
+ 198336.3639 * Make=9
- 411484.1892 [75/15.283%]

Rule: 8
IF
Bonus=1,7 <= 0.5
Zone=5,6,3,2,1,4 <= 0.5
THEN

Payment =
220292.4246 * Kilometres=3,1,2
+ 121350.2008 * Kilometres=1,2
+ 51980.7404 * Zone=5,6,3,2,1,4
+ 98791.1876 * Zone=6,3,2,1,4
+ 143008.1719 * Zone=3,2,1,4
+ 21732.8172 * Zone=2,1,4
+ 232987.3234 * Zone=4
+ 47901.1763 * Bonus=6,1,7
+ 245205.7696 * Bonus=7
+ 295219.4281 * Make=9
- 409061.2887 [40/1.135%]

Rule: 9
IF
Bonus=1,7 <= 0.5
Zone=2,1,4 <= 0.5
Make=9 > 0.5
Kilometres=1,2 <= 0.5
Zone=3,2,1,4 <= 0.5
THEN

Payment =
321891.0225 * Kilometres=3,1,2
+ 305480.0235 * Kilometres=1,2
+ 145994.4868 * Zone=5,6,3,2,1,4
+ 102237.9602 * Zone=6,3,2,1,4
+ 257780.0819 * Zone=3,2,1,4
```

```
+ 15236.8543 * Zone=2,1,4
+ 270768.3076 * Zone=4
+ 12505.52 * Bonus=2,6,1,7
+ 132364.1597 * Bonus=6,1,7
+ 376389.8399 * Bonus=7
+ 531624.545 * Make=9
- 683069.4256 [25/3.033%]

Rule: 10
IF
Bonus=1,7 <= 0.5
Make=9 > 0.5
Bonus=2,6,1,7 <= 0.5
THEN

Payment =
546514.2348 * Kilometres=3,1,2
+ 184936.2655 * Kilometres=1,2
+ 227768.9955 * Zone=5,6,3,2,1,4
+ 343705.1327 * Zone=3,2,1,4
+ 35182.3198 * Zone=2,1,4
+ 75498.2176 * Zone=1,4
+ 407769.9664 * Zone=4
+ 149278.7299 * Bonus=6,1,7
+ 128958.4302 * Bonus=1,7
+ 442659.1996 * Bonus=7
+ 630549.5677 * Make=9
- 868432.514 [30/4.176%]

Rule: 11
IF
Zone=6,3,2,1,4 <= 0.5
Make=9 <= 0.5
THEN

Payment =
705552.7989 * Kilometres=3,1,2
+ 234901.1868 * Kilometres=1,2
+ 436979.7017 * Zone=5,6,3,2,1,4
+ 356059.5159 * Zone=3,2,1,4
+ 413913.1955 * Zone=4
+ 62509.5862 * Bonus=1,7
+ 684132.6975 * Bonus=7
+ 857870.4501 * Make=9
- 1398042.2829 [31/4.212%]
```

```
Rule: 12
IF
Kilometres=3,1,2 > 0.5
Zone=3,2,1,4 <= 0.5
Zone=5,6,3,2,1,4 > 0.5
THEN

Payment =
378733.2412 * Kilometres=3,1,2
+ 1341777.2839 * Kilometres=1,2
+ 1088700.6233 * Zone=5,6,3,2,1,4
+ 1978183.4277 * Zone=6,3,2,1,4
+ 978382.3307 * Zone=3,2,1,4
+ 1298489.5047 * Zone=4
+ 177104.8392 * Bonus=1,7
+ 2944698.9838 * Bonus=7
+ 5008960.1526 * Make=9
- 7048862.8556 [19/13.531%]

Rule: 13
IF
Kilometres=3,1,2 <= 0.5
Bonus=7 <= 0.5
Bonus=1,7 <= 0.5
THEN

Payment =
527978.2447 * Kilometres=3,1,2
+ 455806.8245 * Kilometres=1,2
+ 548883.7635 * Zone=5,6,3,2,1,4
+ 659874.5072 * Zone=6,3,2,1,4
+ 207998.4132 * Zone=3,2,1,4
+ 879973.9605 * Zone=4
+ 361981.1628 * Bonus=6,1,7
- 120931.6619 * Bonus=1,7
+ 1338911.9546 * Bonus=7
+ 1783243.2664 * Make=9
- 2877057.2802 [16/3.576%]

Rule: 14
IF
Kilometres=3,1,2 > 0.5
Make=9 > 0.5
Bonus=7 > 0.5
THEN
```

```
Payment =
601155.9701 * Kilometres=3,1,2
+ 2134779.4835 * Kilometres=1,2
+ 302994.5332 * Kilometres=2
+ 5950351.5592 * Zone=5,6,3,2,1,4
+ 526222.2226 * Zone=6,3,2,1,4
+ 3976698.5511 * Zone=4
+ 3339054.1975 * Bonus=7
+ 3467950.034 * Make=9
- 6653504.0595 [15/18.703%]

Rule: 15
IF
Zone=3,2,1,4 > 0.5
Make=9 <= 0.5
Kilometres=3,1,2 > 0.5
THEN

Payment =
89815.4498 * Kilometres=4,3,1,2
+ 771191.8904 * Kilometres=3,1,2
+ 1255613.7087 * Kilometres=1,2
+ 359771.5372 * Kilometres=2
+ 516948.1011 * Zone=6,3,2,1,4
+ 1227969.7483 * Zone=4
+ 1101582.4769 * Bonus=7
+ 1973893.4149 * Make=9
- 1648150.4233 [12/12.851%]

Rule: 16
IF
Zone=3,2,1,4 > 0.5
Kilometres=1,2 <= 0.5
Make=9 > 0.5
Bonus=7 <= 0.5
THEN

Payment =
1096332.5416 * Kilometres=3,1,2
+ 2127406.1324 * Kilometres=1,2
+ 370098.2095 * Zone=5,6,3,2,1,4
+ 382552.8368 * Zone=6,3,2,1,4
+ 45061.0001 * Zone=1,4
+ 1085388.9973 * Zone=4
+ 2397290.5325 * Bonus=7
+ 2010404.8137 * Make=9
```

```
- 2475663.8431 [12/7.632%]

Rule: 17
IF
Zone=3,2,1,4 <= 0.5
Bonus=7 <= 0.5
THEN

Payment =
954125.9041 * Kilometres=3,1,2
+ 549902.7647 * Zone=5,6,3,2,1,4
+ 804419.9922 * Zone=6,3,2,1,4
+ 715485.8964 * Zone=3,2,1,4
+ 759931.0415 * Zone=4
+ 234659.1482 * Bonus=7
+ 1586003.3005 * Make=9
- 1966780.8893 [9/2.06%]

Rule: 18
IF
Make=9 > 0.5
Kilometres=3,1,2 <= 0.5
THEN

Payment =
160106.6094 * Kilometres=4,3,1,2
+ 1889967.8232 * Kilometres=3,1,2
+ 1727007.0956 * Zone=6,3,2,1,4
+ 577108.0306 * Zone=3,2,1,4
+ 206361.9902 * Zone=2,1,4
+ 2191353.4254 * Zone=4
+ 1065028.8645 * Make=9
- 765297.1686 [14/15.598%]

Rule: 19
IF
Kilometres=3,1,2 <= 0.5
THEN

Payment =
122676.1599 * Kilometres=4,3,1,2
+ 3252994.0754 * Kilometres=3,1,2
+ 500217.0272 * Zone=1,4
+ 242070.3101 [10/8.352%]

Rule: 20
```

```
Payment =
1588386.4977 * Zone=1,4
+ 5214328.2511 [8/52.373%]



Time taken to build model: 11.02 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.9533
Mean absolute error                  96630.6704
Root mean squared error             307252.6011
Relative absolute error                 26.2058 %
Root relative squared error             30.2006 %
Total Number of Instances             2182
```