Predicting Next-Year Written Premium with Insurance Agency Performance Data

Anthony Winney

July 2026

Abstract

This project creates and selects predictive models for a publicly available dataset of property and casualty insurance data concerning agency performance. Written premium under a logarithmic transform was selected as the target variable, and the other 48 columns in the raw dataset were considered as potential predictors, with careful consideration of leakage – the scenario is predicting written premium for the upcoming year. Data cleaning and predictor analysis were performed to select a reasonable initial set of predictors. The models considered were OLS (Ordinary Least Squares), RF (Random Forests), and LMM (Linear Mixed Models). While an RF was selected as the leading predictive model, OLS served a useful role in rounding out the feature analysis phase of the project, and LMM quantified variance attributable to agency and parent agency identity which could not be measured using the other models. The leading RF model used the predictors identified to be valuable via predictor analysis and OLS and was tuned on a small hyperparameter grid. The RF had a test R^2 of 0.8839, substantially better than the leading OLS model's test R^2 of 0.7552 and the leading LMM model's test R^2 of 0.7607. Further, the LMM revealed that agency identity accounts for 24.22% of the remaining variance after controlling for the same fixed-effect predictors used in the selected OLS and RF models. Motivated by the LMM's result, k-means clustering was performed on the agencies, and an agency cluster predictor was added to the RF, but this showed no improvement. In conclusion, the overall leading model was an RF model tuned for selected hyperparameters using 10 selected predictors.

---

1 Introduction and Roadmap

Predictive modeling comes in many different shapes and sizes, and it is one of the cornerstones of data science and actuarial science. A full modeling project involves much more than just training a specified model to a dataset. Before the modeling starts, the dataset must be understood at a high level so that the modeler knows what he's trying to model and how the data can help him make predictions. A few examples of what the modeler must understand before modeling include what each column and row of the dataset represent, what the possible target variables versus the possible explanatory variables are, and whether there are any problems with the raw dataset that need to

be cleaned first before feeding it into the model. In practice, preparing the raw data for modeling almost always takes substantially more time than the modeling itself. The dataset for this project was pulled from Kaggle, so it is already relatively prepared and cleaned, but not perfect. There were still many items which needed to be resolved before modeling could take place.

The dataset is concerned with property and casualty insurance agency performance, and the main goal of this project is to specify the most accurate model to predict annual written premium and to understand the main sources of its variance. Written premium was selected because it is a reasonable measure of an insurance agency's performance. Once the dataset was cleaned, preliminary analysis was performed on the predictors to gain a deeper understanding of them and create an initial set of predictors that would be reasonable to feed to the models. A few engineered predictors were also created. The project took careful consideration to avoid leakage. In other words, the model can only consider data that is available at the time of prediction in the specified context.

Once a final modeling population and predictor set were approved, the models considered were ordinary least squares (OLS), random forests (RF), and linear mixed models (LMM). The R^2 on the holdout set, or test R^2, which measures reduction in squared error compared to a mean-only null model, was used as the primary metric for selecting the most accurate model. Various other metrics were analyzed to prune the candidate predictor set for each model specification, tune model hyperparameters, and explain variance attributable to agency identity.

Throughout the entire modeling process, there were many instances where judgment calls needed to be made. Apart from communicating results, this report serves to provide the reasoning behind these judgment calls, which hopefully improves its credibility.

2 Dataset Overview and Problem Definition

This project starts with the Kaggle Agency Performance raw dataset. This project called for a dataset with many data points, a moderate amount of predictors, and ample modeling opportunities. This dataset was found by browsing Kaggle for a dataset that meets these criteria. In particular, this raw dataset has 213,328 rows and 49 columns, where each row represents annual data for a unique agency $\times$ product $\times$ state $\times$ year combination, and the columns can be grouped into the following categories: agency and parent agency identity; product, geography, and time; premium and policy volume; loss and profitability; retention and growth; agency characteristics; vendor information; operational history information; and quote and bind counts. All of these columns were considered, but most of them did not survive the predictor analysis stage of the project. The data spans 11 years, from 2005 to 2015. Since the dataset is centered around agency performance, the target variable chosen for prediction should reflect that. Annual written premium seemed like the most logical choice, and it is the sole target variable for the entirety of this project. With only information available at the start of the year, the models sought to predict the annual written premium for that year.

3 Data Cleaning and Final Modeling Population

Although the data was very well-prepared since it was taken from Kaggle, there were a few miscellaneous items that needed investigation and cleaning. This didn't necessarily mean that the

data was faulty, but rather that it is just good practice to ensure the modeling population was well-understood and fits within the scope of the project. In this phase of the project, the train and test split was defined, unusual predictor values were handled, and it was confirmed that the data behaves according to common sense assumptions. Then, a final modeling population was selected, which persisted as the baseline modeling population throughout the entirety of the project.

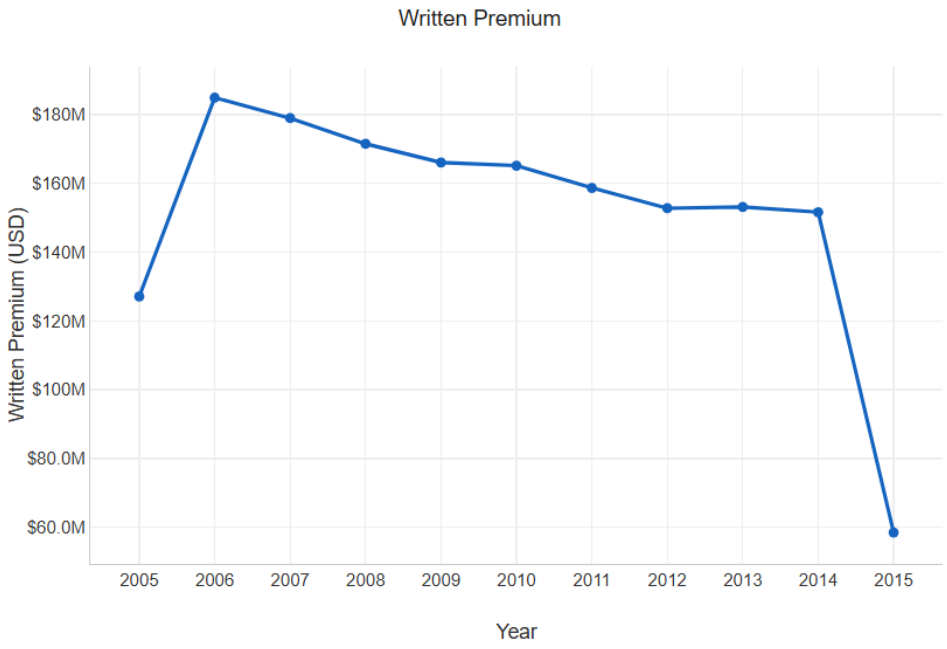

Figure 1: Line Chart Showing Dropoff in Written Premium for Years 2005 and 2015

First, a train and test split was selected. The original dataset spans from years 2005 to 2015. During an initial written premium trend analysis, it was discovered that the total written premium for years 2005 and 2015 were substantially lower than the middle years, as shown in Figure 1. Upon further investigation, it was revealed that the maximum `MONTHS` value, which appears to mean the number of months represented in the observation, was 8 for 2005 and 5 for 2015, which explains the lower written premium values for these years. Since these are incomplete years, they were excluded from all training and testing. A train/test split of 2006 through 2012 for training and 2013 through 2014 for testing was settled upon. By having the two latest years reserved for testing, this mimics real forecasting and helps avoid information leakage.

Agency identity, loss ratio, and written premium all warranted further analysis before they could be safely admitted into the modeling stages. There were two columns relating to agency identity: `AGENCY_ID` and `PRIMARY_AGENCY_ID`. It was confirmed that each value of `AGENCY_ID` maps to one and only one `PRIMARY_AGENCY_ID` which verifies the intuitive conclusion that each agency is nested within a parent agency. There were about 1,600 unique agencies and about 600 unique parent agencies. It is worth noting now that there were many instances where an `AGENCY_ID` value would match its `PRIMARY_AGENCY_ID` value, and there were many instances where the `PRIMARY_AGENCY_ID` value would be the reserved sentinel value of 99999. Since the agency identity columns were not considered until the LMM section, they will be discussed in more detail in that section of this report. Loss ratio appeared to be stable across the years when aggregating portfolio-wide rather

than averaging individual loss ratios. Additionally, 67% of the usable rows contained a loss ratio value which was exactly zero. This initially raised concern, but was determined to be a genuine business phenomenon rather than a data-quality problem because all zero loss ratios were paired with zero incurred losses, 99.63% of zero loss ratios were paired with positive earned premium, and the median policy count for zero loss ratios was 24 compared to a 421 median policy count for nonzero loss ratios.

For written premium, however, non-positive values were determined to be noisy, uninterpretable data points and were excluded from the modeling population. Investigation of non-positive written premium revealed four categories. First, there were about 13,000 rows with a `PROD_ABBR` value of `COMMPOL` which always had zero written premium despite having millions of policies in force. This suggests that `COMMPOL` is a reporting artifact and perhaps an umbrella category which is broken down into more specific `PROD_ABBR` values in other rows, but this is just conjecture. Second, there were structural zeros, where zero premium coincided with zero policies in force. Third, there were zero premium rows with nonzero policy counts, suggesting accounting adjustments or some other ambiguous meaning. Fourth, there were negative premium rows, which similarly suggests ambiguous meaning, perhaps accounting for refunds, credits, or reinsurance activity. In conclusion, only rows with `WRTN_PREM_AMT` > 0 were included in the modeling population as the modeling problem is focused on predicting the written premium amount for active agencies, not predicting whether any premium will be written at all.

After these exclusions, the final candidate modeling population consisted of 103,923 observations in the training set and 31,144 observations in the testing set.

4 Predictor Engineering and Analysis

First, the target variable, `WRTN_PREM_AMT`, was analyzed. Since its values are large (tens to hundreds of millions) compared to the other columns, a log transform was considered alongside raw written premium, namely `log(WRTN_PREM_AMT + 1)`, henceforth called `log_wp`, and `WRTN_PREM_AMT`. Analysis revealed a raw premium skewness of 6.53 compared to a log-transformed skewness of 0.05 and a raw premium p99/p50 ratio of 129 compared to a log-transformed p99/p50 ratio of 1.61. Since one of the assumptions for OLS is homoskedasticity, or constant variance across output values, and the log-transform eliminated the right-skewness in written premium, the log-transformed written premium was selected as the target variable.

To avoid leakage, a few potentially useful previous-year engineered columns were created, namely `PREV_RETENTION_RATIO`, `AVG_PREMIUM_PER_POLICY_LAST_YEAR`, and `PREV_LOSS_RATIO`. Each of these had substantial missingness in the dataset, which was considered in the modeling stage. The previous-year variables `PREV_WRTN_PREM_AMT` and `PREV_POLY_INFORCE_QTY` were already present as columns in the raw dataset. `AVG_PREMIUM_PER_POLICY_LAST_YEAR` was found to have exact linear dependence on `PREV_WRTN_PREM_AMT` and `PREV_POLY_INFORCE_QTY`, so it provides no additional information and was immediately rejected as a candidate predictor. `log_prev_wp` and `log_prev_poly` were created as log-transformed columns for the same reasons discussed in the previous paragraph, and they were considered as candidate predictors instead of their raw counterparts. There was one more small issue that deserves a brief mention relating to the engineered predictors. When using predictors based on the previous year's data, the row is only valid if the previous year's data is also valid, for example with regard to the requirement that written premium be positive. 546 training

rows and 163 testing rows with invalid `PREV_WRTN_PREM_AMT` or `PREV_POLY_INFORCE_QTY` values were dropped, resulting in a final modeling population of 103,377 training rows and 30,981 testing rows.

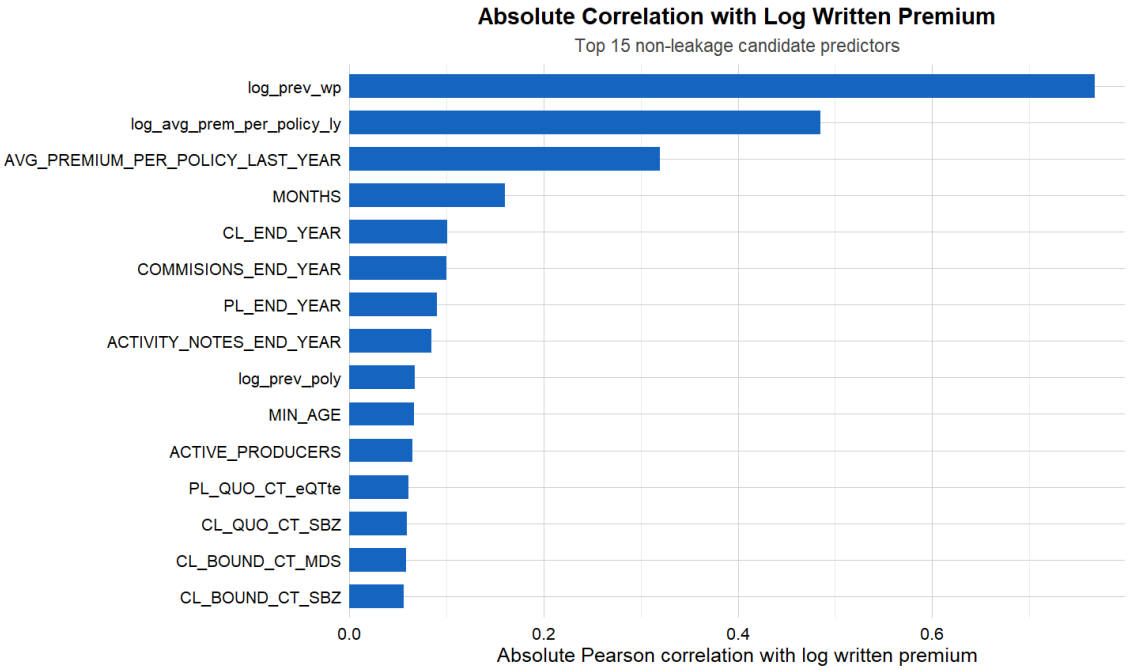

Figure 2: Top 15 Correlations with log_wp

After excluding the aforementioned rows and applying the log transform to written premium, the predictor analysis phase kicked off by considering all pairwise correlations to get an initial feel for which might be the strongest predictors. Figure 2 shows the top 15 highest absolute Pearson correlations with `log_wp`. This does not give a final list of predictors, and there are some predictors such as `PREV_LOSS_RATIO` which have a negligible correlation with `log_wp` but were still considered as potential predictors because they make conceptual sense and could have predictive value in a multivariable model. This initial, loose result for predictor importance served as a benchmark for an OLS progression roadmap, and the predictors were further pruned after analyzing each of the OLS models. This will be discussed further in the OLS section.

Analysis of the product, quote/bind, and operational history variables yielded cause for immediate elimination of certain columns. First, the columns `PROD_ABBR` and `PROD_LINE` are redundant, the former taking on more specific product categories such as `HOMEOWNERS`, `WORKCOMP`, or `SNOWMOBILE`, and the latter taking on one of two values, `PL` or `CL`, corresponding to personal or commercial. Each `PROD_ABBR` value maps to one and only one `PROD_LINE` value. So, the column `PROD_LINE` was excluded from the set of candidate predictors. There were 16 quote/bind columns and 8 operational history columns that were initially considered as candidate predictors. Analysis of these columns showed that they are static over time, i.e. for each agency $\times$ product $\times$ state combination, the values of these columns did not change over time. Other issues were discovered with the operational history variables, such as invalid values, but the static nature of these two families of columns was sufficient reason to exclude them from the candidate predictor set.

5 Ordinary Least Squares

At this point, both the final modeling population and candidate predictor set were finalized, and modeling could begin, starting with OLS. Even though it was never expected that OLS would produce the most accurate predictive model, it serves to round out the predictor analysis phase of the project by providing additional useful information, such as statistical significance of predictors in the form of the p-value. The strategy was to incrementally build upon a baseline OLS by adding predictors and then analyzing the change in test R^2 and test RMSE (root mean squared error). There were 7 additive OLS models tested, additive meaning they contain no interaction or higher-power terms, and then there were 3 OLS models with interaction terms tested. The additive models were labeled `OLS 0` through `OLS 5` plus `OLS_ADDITIVE_FINAL`, while the interaction models tested were labeled `OLS_INTERACTION_FULL`, `OLS_INTERACTION_SUPPORTED`, and `OLS_INTERACTION_NUMERIC_ONLY`. This report will now define each model one-by-one and discuss their results because they each reveal key insights into the dataset's relation to written premium that will be useful later.

The 7 additive models, `OLS 0` through `OLS 5` plus `OLS_ADDITIVE_FINAL`, were defined by starting with the simplest model possible in `OLS 0`, progressively adding predictors to create more complex models, and intermediately analyzing the results to draw conclusions about the predictive power of the added predictors. Intuition and preliminary correlation analysis agreed that `log_prev_wp` (log of the previous year's written premium) was the most powerful predictor for `log_wp`, so `OLS 0` was defined as having `log_prev_wp` as its only predictor. `OLS 1` is `OLS 0` plus `log_prev_poly` (log of the previous year's number of policies in force); `OLS 2` is `OLS 1` plus agency characteristics predictors, namely `ACTIVE_PRODUCERS`, `AGENCY_APPOINTMENT_YEAR`, `MAX_AGE`, and `MIN_AGE`; `OLS 3` is `OLS 2` plus geography, product mix, vendor, and time-based predictors, namely `STATE_ABBR`, `PROD_LINE`, `PROD_ABBR`, `VENDOR`, and `STAT_PROFILE_DATE_YEAR`; `OLS 4` is `OLS 3` plus `PREV_LOSS_RATIO`; and `OLS 5` is `OLS 3` plus `PREV_RETENTION_RATIO`. Note that some of the models include `PROD_LINE` which earlier in the report was discussed as being redundant with `PROD_ABBR`. This is because that redundancy was discovered through OLS modeling, even though conceptually it falls under the predictor analysis phase of the project.

Model	Test R^2	Test RMSE
OLS 0	0.6035	1.3919
OLS 1	0.6976	1.2156
OLS 2	0.7003	1.2102
OLS 3	0.7552	1.0937

Figure 3: OLS 0 through 3 Results Progression Table

Results of the additive OLS modeling will now be discussed. Refer to Figure 3 for a table of the test R^2 and test RMSE results for each of OLS models 0 through 3. With a test R^2 of 0.6035, the

results of OLS 0 show that log_prev_wp alone already explains a substantial portion of the variance of log_wp. OLS 1 showed a substantial increase in test R^2 over OLS 0 (+0.0941); OLS 2 showed a very small increase in R^2 over OLS 1 (+0.0027); and OLS 3 showed a substantial increase in test R^2 over OLS 2 (+0.0549). Therefore, it was concluded that the agency characteristics added in OLS 2 provide minimal additional predictive power while the rest of the predictors added in OLS 1 and OLS 3 add meaningful signal.

OLS 4 and OLS 5 were concerned with testing whether PREV_LOSS_RATIO and PREV_RETENTION_RATIO provide additional predictive signal. OLS 4 added PREV_LOSS_RATIO to OLS 3. As discussed in the data cleaning section, a large portion of the loss ratio data was missing in the raw dataset. To accurately test whether previous loss ratio results in a more accurate model, OLS 3 must be refit to the same population as OLS 4, i.e. the population for which previous loss ratio data is available. Call these complete case, population restricted models OLS 3CC and OLS 4CC. The results of the comparison are a +0.0000 test R^2 and a +0.0001 test RMSE going from OLS 3CC to OLS 4CC. So, PREV_LOSS_RATIO was excluded from the set of valuable predictors for the final OLS model. As a side note, PREV_LOSS_RATIO remained statistically significant with a p-value of 0.000019, demonstrating that statistical significance does not necessarily imply practical predictive importance. A very similar process was taken for OLS 5. Retention ratio data also exhibited a substantial amount of missingness in the raw dataset, so OLS 3 needed to be refit again. The models compared in this step were called OLS_3_RET_CC and OLS_5_RET_CC. However, adding PREV_RETENTION_RATIO only contributed +0.0009 to the test R^2, so it was also excluded from final OLS model consideration, even though it was also found to be statistically significant.

This leaves the final additive OLS model needing to be specified. OLS 3 represented the most complex model which showed improvement over simpler models, so it serves as OLS_ADDITIVE_FINAL with one final pruning step for removing the redundant PROD_LINE predictor. Since the predictors added in OLS 2 did improve both test R^2 and test RMSE, albeit marginally, and since overfitting was not a concern at this point, those predictors were kept in the final additive model. PROD_LINE was actually discovered to be redundant in the OLS 3CC and OLS 4CC comparison because the coefficient of PROD_LINE_PL was unstable, going from 1.85 to -0.21 between the two models, prompting further investigation and confirmation of redundancy. This leaves the final OLS_ADDITIVE_FINAL as identical to OLS 3 except with PROD_LINE removed. OLS_ADDITIVE_FINAL resulted in a final test R^2 of 0.7552 and test RMSE of 1.0937, exactly matching OLS 3.

With an additive OLS model being finalized, there were 3 OLS models with interaction terms tested next: OLS_INTERACTION_FULL, OLS_INTERACTION_SUPPORTED, and OLS_INTERACTION_NUMERIC_ONLY. The first model, OLS_INTERACTION_FULL, added every possible combination of interaction terms to OLS_ADDITIVE_FINAL. This resulted in 628 interaction terms being added, including 82 terms which R identified as aliased due to a rank-deficient design matrix, and a catastrophic test R^2 value of -42.1610. This was likely due to categorical $\times$ categorical interaction terms being very sparsely represented in the training data. The total failure of OLS_INTERACTION_FULL prompted an investigation into which interaction families are feasible. A feasibility review indicated that categorical $\times$ categorical interaction terms were not suitable for OLS due to sparse training data, but numeric $\times$ categorical families were feasible. OLS_INTERACTION_SUPPORTED included such numeric $\times$ categorical interaction terms, but it still failed catastrophically with a test R^2 of -5.7008, and there were still aliased terms recognized by R. Finally, OLS_INTERACTION_NUMERIC_ONLY was fitted using only numeric $\times$ numeric interaction terms. This resulted in a reasonable model, but it still showed a decline in predictive quality, exhibiting a 0.0243 decrease in test R^2 and a 0.0530 increase in test RMSE over OLS_ADDITIVE_FINAL, which

suggested that adding interaction terms resulted in overfitting, despite 15 of the 21 numeric × numeric interaction terms being statistically significant at $p < 0.05$. Therefore, `OLS_ADDITIVE_FINAL` is the benchmark OLS model moving forward, and any other models should be judged against it on the same training and testing population.

6 Random Forests

After the conclusion of OLS, which provided a baseline predictive model for comparison and offered additional insights into the data, the project shifted to a more powerful model, random forests, henceforth referred to as RF. RF is similar to standard decision trees, with a few modifications. Instead of considering all predictors at each split and choosing the split with the most information gain, a small, random subset of predictors is chosen as candidates for each split. Plus, it incorporates bagging, which involves training many decision trees and averaging all of their output values to reach a final prediction, which reduces variance without increasing bias.

Parameter	Values
num.trees	500
mtry	3, 5, 7, 10
min.node.size	5, 20, 50, 100
sample.fraction	0.6, 0.8, 1.0

Figure 4: Hyperparameter Tuning Grid for `RF_1_SAFE_TUNED`

The target variable, modeling population, train/test split, and predictor set were held constant to the specifications used for `OLS_ADDITIVE_FINAL` to ensure comparative validity. Two RF models were trained: `RF_0_SAFE_DEFAULT` and `RF_1_SAFE_TUNED`, where the former is a default RF model using reasonable hyperparameter values and the latter is a lightly tuned RF model to minimize out-of-bag error on a small hyperparameter grid. See Figure 4 for the hyperparameters that were tuned. `num.trees` is the number of individual trees grown in the random forest; `mtry` is the number of predictors randomly considered for each split; `min.node.size` is the minimum number of observations allowed in each terminal node; and `sample.fraction` is the fraction of the training data used to build each individual tree. Note that for `sample.fraction` < 1.0, the default R behavior is to perform bootstrap sampling without replacement, as opposed to sampling with replacement for `sample.fraction = 1.0`, and this behavior was kept. `RF_0_SAFE_DEFAULT` had pre-decided default hyperparameter values of `num_trees = 500`, `mtry = 3`, and `sample_fraction = 1.0`.

The results of RF modeling were quite good. `RF_0_SAFE_DEFAULT` showed a substantial improvement on the test metrics over `OLS_ADDITIVE_FINAL` with a substantial increase in test R^2 (+0.1281) and a substantial decrease in test RMSE (-0.3385). Tuning resulted in the hyperparameter val-

ues of num.trees = 500, mtry = 3, min.node.size = 5, and sample.fraction = 0.6 for the RF_1_SAFE_TUNED model. However, tuning resulted in a tiny improvement over RF_0_SAFE_DEFAULT with a minimal increase in test R^2 (+0.0006) and a minimal decrease in test RMSE (-0.0020). Still, since these are improvements nonetheless on a large testing population, RF_1_SAFE_TUNED was selected as the benchmark RF model. These results showed that for this dataset, RF provides a large performance gain over OLS due to model specification itself, not from aggressive tuning. RF was likely able to capture nonlinear and interaction structure that OLS could not.

Permutation importance was used to gauge predictor importance in the RF model, which is measured by the drop in model performance when a predictor is randomly shuffled. Permutation importance was chosen over impurity-based importance metrics because some of the categorical variables have many categories, which could bias the impurity importance measure toward those categorical variables. The top 5 normalized permutation importance results were as follows: log_prev_wp = 60.77%, log_prev_poly = 16.88%, PROD_ABBR = 10.54%, AGENCY_APPOINTMENT_YEAR = 2.69%, and ACTIVE_PRODUCERS = 2.32%. This matches the conclusion drawn from OLS and gives even more insight. As a reminder, log_prev_wp was the only predictor in the baseline model OLS 0 and offered the most accuracy gain, followed by log_prev_poly which was introduced in OLS 1, followed by geography, product, time, and vendor information, including PROD_ABBR, in OLS 3. Adding agency characteristic information, including AGENCY_APPOINTMENT_YEAR and ACTIVE_PRODUCERS, in OLS 2 provided little performance gain. These feature importance conclusions match, and the results of RF analysis additionally revealed that PROD_ABBR was the main driver of performance gain out of the group of predictors that were added to OLS 3.

There was one small miscellaneous data quality concern that was addressed at this point in the project, so it will be briefly discussed now. The concern was that since 2005 represents partial-year data, using 2006 as a training year with previous-year predictors could be a data quality issue. A sensitivity test was performed to gauge the impact of using potentially corrupt previous-year predictors. OLS_ADDITIVE_FINAL and RF_1_SAFE_TUNED were refit using training data from 2007 through 2012, notably excluding 2006. Thankfully, there was a minimal decline in test R^2 (< 0.001) and a minimal increase in test RMSE (< 0.002) for OLS, and the changes in these metrics were near-zero for RF. Additionally, the coefficient changes for each of the predictors were small for OLS and the feature importance changes were small for RF. Therefore, the conclusion was to keep the original 2006 through 2012 training window and 2013 through 2014 testing window, and that previous-year predictor values from 2006 did not materially distort the models.

7 Linear Mixed Models

Models up until this point could not consider two important columns in the raw dataset: AGENCY_ID and PRIMARY_AGENCY_ID. That's where a linear mixed model, henceforth referred to as LMM, can help. LMM is like OLS, except it also has random effect variables, which will be the agency and parent agency identities in this case. The assumption for LMM is that the random effect variables account for variance attributable to group-level differences not otherwise explained by the fixed effect variables, where the groups seen in the data are only a subset of all possible groups in the world's population. In this case, the groups are agency identity and parent agency identity, and the motivation for LMM modeling is to discover how much variance in written premium is attributable to those. This is especially useful in this case, and in general, when there are too many groups to treat the random effect variable as an ordinary categorical variable. In this dataset, there are

1,623 unique values for AGENCY_ID and 588 unique values for PRIMARY_AGENCY_ID, so treating these columns as categorical variables to be fed into OLS or RF would have almost certainly resulted in overfitting because the data for each category is sparse. However, conditional predictions can be made which do consider the historical data corresponding to agency and parent agency identity as a credibility weighted average, where credibility is assigned depending on the number of observations in the training data for the particular agency or parent agency identity value. This is a perfect use case for LMMs.

Recall from earlier in this report that it has been verified that each value of AGENCY_ID corresponds to one and only one value of PRIMARY_AGENCY_ID. In LMM terms, it can be said that agency identity is nested within parent agency identity, which allows for the use of nested LMMs. So, the main cases considered in this section are LMM modeling with agency identity only and nested LMM modeling with agency identity nested within parent agency identity. For each LMM, the set of fixed effect predictors matches the set of predictors used in OLS_ADDITIVE_FINAL and RF_1_SAFE_TUNED. All other modeling characteristics also match, so comparative validity is maintained. There were 3 main branches of the LMM modeling phase: considering agency identity only; considering agency identity nested within parent agency identity; and considering agency identity nested within parent agency identity, excluding all data where agency identity equals parent agency identity. For each of these three phases, two different prediction methods were used: marginal predictions, which only consider the fixed effects, and conditional predictions, which consider the fixed effects and historical data from the random effects. Comparing results from marginal predictions versus conditional predictions gauged the predictive gain from considering historical data associated with the agency and parent agency identities. Note that 100% of the rows in the dataset had a valid value for agency identity, but many of the rows had sentinel values for parent agency identity. A model label will either have the suffix *_FULL to reflect that it was trained on all the data or *_CC to reflect that it was trained on data excluding certain offending rows.

The LMM modeling phase started with the agency-only LMM, called AGENCY_LMM_FULL. To give a better idea of the LMM structure, the formula for this model is $\texttt{log\_wp}_{ij} = X_{ij}\beta + u_j + \varepsilon_{ij}$ where $X_{ij}\beta$ is the fixed-effect component from OLS_ADDITIVE_FINAL, u_j is the agency-level random intercept for agency j, and ε_{ij} is the residual error. Marginal predictions for this model yielded a test R^2 of 0.7345 and a test RMSE of 1.1389, and conditional predictions for this model yielded a test R^2 of 0.7607 and a test RMSE of 1.0814. The conditional predictions resulted in a notable improvement on both test R^2 (+0.0262) and test RMSE (-0.0575) over marginal predictions which shows that agency-specific credibility carries valuable predictive information. This is an important result, but this LMM only slightly beats out OLS_ADDITIVE_FINAL and pales in comparison to RF_1_SAFE_TUNED when it comes to predictive accuracy. The intraclass correlation coefficient, or ICC, which measures the proportion of variance attributable to group-level differences, is an additional powerful metric given by LMMs. The agency ICC for AGENCY_LMM_FULL was 0.2422, which means that after controlling for fixed effects, agency identity still accounts for 24.22% of the remaining unexplained variance. This further illustrated that agency identity explains a substantial amount of variance.

Then, it was time to consider the parent agency identity. This next LMM was called NESTED_LMM_PARENT_CC, and its formula is $\texttt{log\_wp}_{ijk} = X_{ijk}\beta + v_k + u_{j(k)} + \varepsilon_{ijk}$, where $X_{ijk}\beta$ is the fixed-effect component from OLS_ADDITIVE_FINAL, v_k is the random intercept for parent agency k, $u_{j(k)}$ is the random intercept for agency j nested within parent agency k, and ε_{ijk} is the residual error. As mentioned previously, many rows in the data had sentinel values for PRIMARY_AGENCY_ID, hence the *_CC suffix in NESTED_LMM_PARENT_CC. To ensure valid comparisons, the agency-only

LMM was refit to the same restricted population, called `AGENCY_LMM_PARENT_CC`. The numerical results using conditional predictions for `NESTED_LMM_PARENT_CC` were not as promising, showing negligible improvement over `AGENCY_LMM_PARENT_CC` in both test R^2 (+0.0004) and test RMSE (-0.0007). However, the parent ICC for the nested model was 0.0438, meaning that after accounting for fixed effects, parent agency accounts for about 4.38% of the remaining residual variance. This is not insignificant, but still, the relevant statistics showed that most of the hierarchical variance remains at the agency level, and incorporating parent agency credibility does not substantially improve predictive accuracy.

At this point, there was still one more item that needed consideration. It was briefly mentioned earlier that many agencies self-parent, i.e. its `AGENCY_ID` value equals its `PRIMARY_AGENCY_ID` value. The concern was that this structural characteristic could be muddying the results when quantifying variance attributable to the agency identity versus the parent agency identity. To remedy this, one final model was fit: `NESTED_LMM_TRUE_PARENT_CC`, which restricts the modeling population to only data in which the agency does not self-parent, so that the effects of parent agency identity could be fully isolated. Again, the agency-only LMM was refit to this modeling population so that comparative validity could be preserved, and this model was called `AGENCY_LMM_TRUE_PARENT_CC`. Similarly to the previous nested model, `NESTED_LMM_TRUE_PARENT_CC` showed negligible difference (this time actually a decline in quality) over `AGENCY_LMM_TRUE_PARENT_CC` in both test R^2 (-0.0002) and test RMSE (+0.0005). The parent ICC for this model was 0.0352.

The results of the LMM modeling section are clear. Agency identity did explain a substantial amount of variance and added predictive information. Parent agency identity explained a small amount of variance but did not materially improve predictive accuracy, even in the case where agency identity and parent agency identity are distinct. So, `AGENCY_LMM_FULL` survived as the leading LMM predictive model.

8 Random Forests with Agency Clustering

Even though the leading LMM could not beat out the predictive accuracy of the leading RF, the LMM modeling section still provided a useful result: there is a substantial amount of variance attributable to agency identity not explained by the set of predictors which have been used as fixed effects for OLS, RF, and LMM. This begs the question: would incorporating agency identity, somehow, into the leading RF model improve its predictive accuracy? To answer this question, cluster analysis was performed on the agency identities to create a new engineered predictor, called `AGENCY_CLUSTER`, which would have a manageable number of categories, in the double digits, compared to the large number of categories of `AGENCY_ID`, which was over 1000.

To do this, k-means clustering was performed on the agencies, using only information in the 2006 through 2012 training period, to avoid leakage and stay consistent with the train and test split that has been used throughout this project. An agency profile dataset was created, where each row represents one agency, on which k-means clustering was to be performed. So, there were 1,254 rows in this dataset, equal to the number of unique agencies seen in the training data. After considering an initial 27 columns and running diagnostics to eliminate redundancy and other issues, the columns for this dataset finally included 15 engineered predictors which aimed to best capture the spirit of agency identity. These columns were constructed by aggregating their values across all rows belonging to each agency in the raw dataset for the 2006 through 2012 training period. All of the

predictors were also standardized to avoid giving unequal weight to predictors with differing ranges of values.

The 15 predictors were grouped into 8 families: support / credibility, which captured the amount of historical information for each agency; premium / policy scale, which captured historical written premium and policy volume; volatility / stability, which captured whether an agency is historically stable or volatile; agency characteristic, which captured the number of active producers in an agency; breadth, which captured whether an agency was narrow or diversified across products and states; concentration, which captured concentration instead of raw breadth counts; product-line mix, which captured the share of personal versus commercial lines; and trend, which captured the historical growth or shrinkage in an agency's written premium.

With the agency profile dataset being finalized, the agencies were ready to be clustered, and k-means clustering was performed for the following values of k, or the number of clusters: 5, 10, 20, 30, and 50. That is, `AGENCY_CLUSTER` was an engineered predictor appended to the original dataset, where each row took a value between 1 and k, which corresponded to the categorical predictor value associated with a particular cluster. `RF_1_SAFE_TUNED` was retrained to include the `AGENCY_CLUSTER` predictor, once for each value of k. Unfortunately, no retrained RF beat `RF_1_SAFE_TUNED` in terms of test R^2. That being said, agency clustering didn't result in a notable decline in predictive accuracy. For each k, the test R^2 of the retrained RF was within 0.005 of the test R^2 of `RF_1_SAFE_TUNED`.

Interestingly, permutation importance on the leading RF with agency clustering ranks `AGENCY_CLUSTER` as the 4th most important feature at 3.84%. However, the results of agency clustering suggest that incorporating agency identity does not materially change the predictive accuracy of the leading RF model, and that at least for this context, `AGENCY_CLUSTER` does not provide additional information beyond the existing predictor set.

9 Conclusion

The modeling phase of this project was then completed, and `RF_1_SAFE_TUNED` survived as the ultimate leading predictive model for `log_wp`. Staying consistent with the rest of this report, test R^2 and test RMSE were used to compare the predictive accuracies of each model.

Model	Test R^2	Test RMSE
OLS_ADDITIVE_FINAL	0.7552	1.0937
RF_1_SAFE_TUNED	0.8839	0.7532
AGENCY_LMM_FULL *	0.7607	1.0814

* Conditional predictions.

Figure 5: Test Statistics for Final Benchmark Models

See Figure 5 for a key summary of each benchmark model's predictive accuracy performance. `AGENCY_LMM_FULL` edges out the `OLS_ADDITIVE_FINAL` by a modest $\Delta R^2 = +0.0055$ thanks to the

agency credibility incorporated via conditional predictions, and `RF_1_SAFE_TUNED` dominated both with a very substantial $\Delta R^2 = +0.1232$ over `AGENCY_LMM_FULL`. Even though the other models could not top RF in terms of predictive accuracy, they each provided useful information. The OLS models gave insight into predictor importance beyond correlation measures, which solidified a consistent set of predictors to be used for subsequent modeling. The LMM models provided a key insight into the proportion of variance explained by agency identity and parent agency identity, and that accounting for agency-specific historical data via conditional predictions allows the leading LMM model to beat out the leading OLS model.

	Original (2006–2012)		Refit (2006–2014)		Δ = Refit metric - Original metric	
Model	Train R^2	Train RMSE	Train R^2	Train RMSE	Train R^2	Train RMSE
OLS_ADDITIVE_FINAL	0.7810	1.0238	0.7809	1.0267	-0.0001	+0.0029
RF_1_SAFE_TUNED †	0.8989	0.6955	0.9000	0.6935	+0.0011	-0.0020
AGENCY_LMM_FULL *	0.8040	0.9685	0.8033	0.9727	-0.0007	+0.0042

No holdout available after all-data refit.
* Conditional predictions.
† OOB metrics.

Figure 6: Original vs. Refit In-Sample Diagnostics

The standard final-model step of refitting the leading models on all available data, in this case to additionally include 2013 and 2014 that was thus far reserved for testing, was completed. Figure 6 shows the training set diagnostics for each leading model on the original training set of 2006 through 2012 compared to the refit models on the new training set of 2006 through 2014. Since all data was used for training, there was no holdout set, so test statistics could not be reported. This step was to simulate a real-life business environment, where the models should be retrained on all available data before they are deployed for actual usage. This serves to increase stability, and the small deltas indicate that the models were already quite stable before refitting.

For future work, other nonlinear models could be explored. Since the RF was so effective, boosting could be tried next to see if there is even more improvement. There is a large enough data pool that a neural network model could lead to the strongest predictive model. Additionally, given the LMM result, it could be fruitful to experiment with other methods of incorporating agency identity into the model. For now, it is not surprising that RF was the leading model compared to the other linear models. That being said, a final test R^2 value of 0.8839 for the `RF_1_SAFE_TUNED` is satisfactory for this project.

10 References and Reproducibility

All scripts were written in R, and JSON outputs were generated to summarize results at important milestones in the project. For reproducibility purposes, all the code, outputs, and figures can be found in the public GitHub repository here: `https://github.com/anthonywinney/insurance-modeling-backend`. The Kaggle dataset can be found here: `https://www.kaggle.com/datasets/moneystore/agencyperformance`.