**Capstone 1**

**Allstate Claims Severity**

**Problem**

Insurance companies are constantly seeking new ways to improve with the goals of encouraging growth and gaining greater market share. One of the many key aspects an insurer must manage is how satisfied its existing customers are with the products and services as this directly relates to retaining customers and the likelihood of a customer referring new clients. Reducing the amount of time a customer spends in the claims process is an important metric for customer satisfaction as it allows customers to focus more time to focus on what matters more to them. To this effect I will explore if it is possible to predict the cost of a claim based on claim details provided by Allstate Insurance.

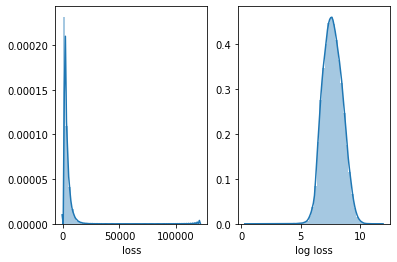
**Clients**

The client is Allstate, a United States personal insurer, who is looking to improve their claims service by developing automated methods of predicting the cost of a claim. A model that can accurately estimate the cost of a claim can decrease the time involved for the claims process, put information in the hands of claimants faster and put them at ease, and also be used in other models (i.e. pricing) to better manage income. The general approach and analysis may be useful for many other personal insurers.

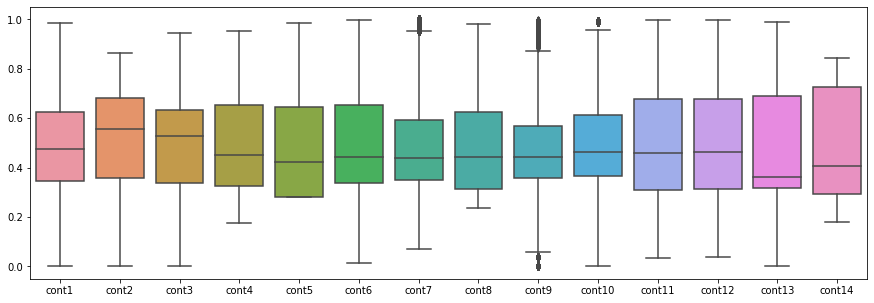
**The Data**

The data for was downloaded from the Allstate Claims Severity Kaggle competition  in CSV format (<https://www.kaggle.com/c/allstate-claims-severity/data>). The training data was then imported into Python 3.7 and analyzed. The data was confirmed to have had no missing values to address. In examining the training data set we see there are over 180,000 claims records. Each record contains 116 categorical features, 14 continuous features, and a continuous target variable ‘Loss’. Here, loss represents the dollar value of how much the claim cost the insurer and can also be considered a measure of severity (with higher costs signifying greater severity.) All of the data has been deidentified: the continuous features have been normalized and each categorical feature is represented by one or more letter of the alphabet depending on the number of unique values (i.e. A to D if there are 4 values for the feature or A to AB if there are 28.) The training data set contains  over 188,000 unique insurance claim records.

The target variable “Loss” appears to be right-skewed. Taking the natural log of each value yields a normal distribution. The log of our target variable will be used for further analysis and in building a predictive model for “Loss”.



The continuous features are all normalized and there do not appear to be many outliers for any of the features. From the histograms below, “cont11” and “cont12” appear to have very similar distributions of their respective data.

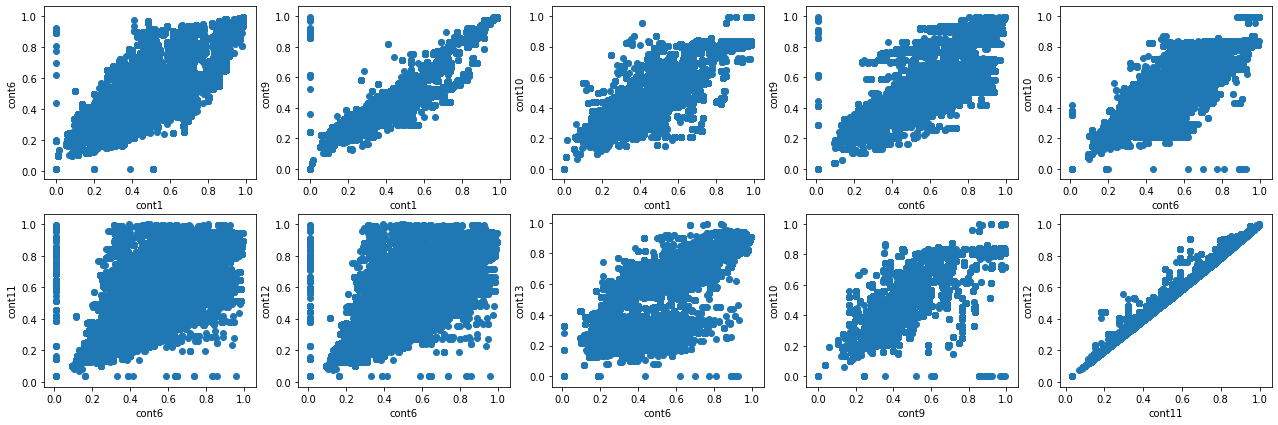


Most of the categorical features are binary (A or B) with A typically outnumbering B by an overwhelming amount. Assuming men and women are insured at similar rates, “Cat2” may represent gender (male/female). Also, “Cat112” contains 51 distinct categories which may correspond to the 50 U.S. states and the District of Columbia.

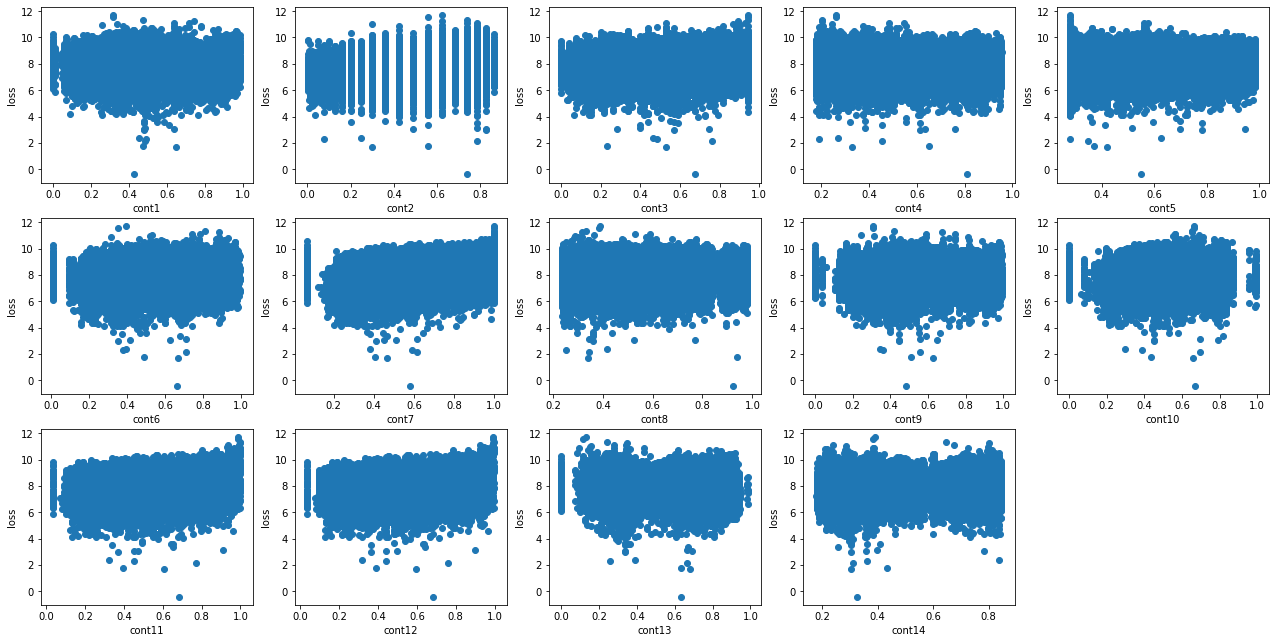
A screenshot of a cell phone

Description automatically generated

High correlation can be observed among several of the continuous features. In particular the correlation between “Cont11” and “Cont12” and the correlation between “Cont1” and “Cont9” are almost equal to one; therefore, we may be able to drop one from each pair without impacting our model.



There is little correlation between any of the continuous features and the target “Loss”. In examining “Cont2” plotted against “Loss” it appears that “cont2” takes on discrete numerical values. Further analysis reveals that the over 180,000 records take on only 33 distinct values in “Cont2” which indicates that though this feature consists of numbers it may be categorical (i.e. internal codes to classify a feature of a claim ; therefore, I will treat this “Cont2” as a categorical feature when building the predictive model.



**The Approach**

Because the target variable is continuous and there is no ambiguity within I aimed to construct a regression model using supervised learning algorithms. All of the categorical features were one-hot encoded and merged into a dataframe with the continuous features. There is a significant trade off here as I avoid any ordinality and allow for machine learning, but in exchange I increase my feature space from 130 to 1153. Due to the size of the training data I chose to construct several models using the default parameters or parameters that limited the training time of several algorithms (i.e. the node length of tree algorithms was limited) then perform feature selection and tune the model that performed best. Performance was measured by the mean absolute error on the test data; therefore, the lower the score the better. The algorithms taken into consideration were linear regression, ridge regression, lasso regression, Decision Tree, Random Forest, and, the highly favored on Kaggle, gradient boosted decision trees (XGBoost).

|  |  |  |
| --- | --- | --- |
| **Model** | **MAE Train** | **MAE Test** |
| Linear Regression | 1231.55 | N/A |
| Ridge Regression | 1236.55 | 1258.93 |
| Lasso Regression | 1810.30 | 1807.93 |
| Decision Tree | 1240.38 | 1298.71 |
| Random Forest | 491.62 | 1209.42 |
| XGBoost | 1083.91 | 1170.48 |

XGBoost performs the best with lasso regression and linear regression trailing the pack. The linear regression algorithm predicted the training data fairly well; however, when predicting the test data the linear regression model predicted values that were not within reason. The ridge regression algorithm performed fairly well and also appears to suffer from less over fitting than the competing tree-based algorithms. Because XGBoost performs the best and also does not require a great deal of computation time I will use it to determine my final model for predicting the cost of a claim.

I again one-hot encoded the categorical features and included “Cont2” (the continuous feature that appears to be categorical data represented numerically.) This increases my feature space from 1153 to 1185. To reduce the time required to fit and tune the model I performed feature selection based on each feature’s F-Score. There was very little loss reducing the feature space to 125 features (112 categorical and 13 continuous.)

|  |  |  |
| --- | --- | --- |
| **# of Features** | **MAE Training** | **MAE Test** |
| 1185 | 1087.21 | 1175.86 |
| 125 | 1094.72 | 1185.92 |

To ensure that a generalized model was created, 5 - fold cross validation was utilized. A grid search was used in conjunction with XGBoost to find an optimal range for several parameters, then a grid search was used with all of the parameters to find the optimal combination for the model.

|  |  |
| --- | --- |
| **Parameter** | **Values** |
| max\_depth | 5, 6, 7 |
| gamma | 1, 2, 3 |
| eta | 0.25, 0.3, 0.35 |
| alpha | 0, 0.2, 0.4 |

**Conclusion**