Udacity MLND: Capstone Proposal

Eugene Kwak

June 30, 2018

1. Domain Background

Balancing losses and premiums is the crux of managing a successful insurance business. This balancing act consists of being able to quantify and understand what losses (risk) will be incurred in order to set premiums at rates that are both competive in the marketplace and attractive to consumers while guaranteeing profitability. Here I focus on methods for quantifying risk. Insurers invest heavily on teams of analysts, domain experts and actuaries to perform this task. Countless studies are also regularly published exploring machine learning approaches to solve this problem and evaluate commercial solutions (see comparative analysis of risk models in healthcare by the Society of Actuaries from 2016)[1]. In short, risk prediction is and always has been one of the biggest challenges in insurance.

Both insurers and consumers benefit when insurers can assess their financial risk as accurately as possible. This is no trivial task. Traditional approaches often rely on stringent assumptions and sometimes arbitrary methods. With the advent of machine learning and big data in commercial insurance, a world of possibilities opened up. Companies are eagerly seeking new ways to apply novel techniques to improve their business, but are often challenged with the complexities that come with supporting these technologies. For instance, entire infrastructures may need to be replaced as traditional relational database management systems (RDBMS) are not suitable for distributed computing, which is the backbone of many big data architectures like Hadoop. It is also often unclear whether incorporating these new technologies is even worth the upfront investment. In order to provide clarity, research is needed on the applicability of cutting edge approaches to this classic problem.

2. Problem Statement

Can deep neural networks be a viable solution for an automated risk assessment tool in insurance claims?

Risk prediction often includes cost as a dependent variable making this a regression problem. I believe that makes this investigation a great extension to the Deep Learning section of this Nanodegree as the program focused more on classification tasks.

In this project, I will explore the viability of deep neural networks in risk prediction within the domain of auto insurance. I will use the problem proposed by Allstate in a Kaggle competition[2]. There are plenty of solutions applied that have performed very well using traditional machine learning techniques. Even linear regression tends to perform quite well at this task. My examination will cover two comparisons. The first will be to compare a deep neural network against a reasonably performant and sophisticated benchmark model. The second will be to compare against the median performance attained from Kaggle submissions, which evaluated models on the mean absolute error (MAE). The median MAE is **1125.5644**.

The two performance metrics I will use are R^2 and MAE.

A proper deep learning architecture requires substantial investment in both hardware and talent. If I find reasonable performance above the above mentioned comparisons, then I believe it establishes the groundwork for building a case to support such an investment.

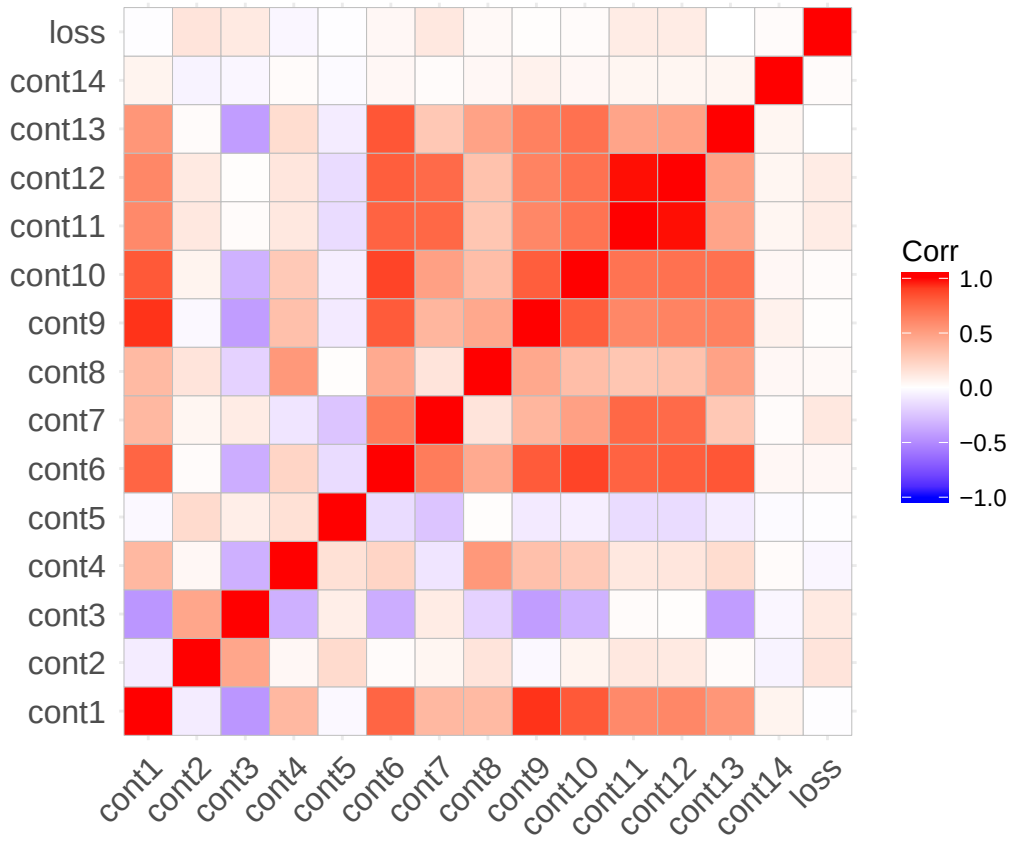
3. Datasets and Input

For their Kaggle competition, Allstate provided both train and test data. The model is to be trained on the train data and used to make predictions on the test data for submission to Kaggle. All data are completely anonymized by Allstate.

The train data consists of 200,867 claims and 130 predictors. The dependent variable "loss" is a measure of cost for the claim. Of the predictors, 116 are categorical and 14 are continuous. The 14 continuous variables have been scaled with a mean of 0.5 and standard deviation of 0.2. It is unclear as to whether any of the features provided were dates. Claims typically are transactional, so the lack of a clear time variable slightly limits the breadth of the analysis. However, the unit of analysis appears to be at the claim level which is often standard practice for these sorts of analyses and thus appropriate for this study.

Continuous variables are prefixed with "cont" and categorical variables are prefixed with "cat". **Figure 1** shows a correlation heatmap of the continuous variables including the target variable (loss).

Figure 1: Correlation matrix of continuous variables



4. Solution Statement

Since the problem comes in two parts, then so should the solution. Again, the goal is to create a deep learning regression model to determine whether it is a viable solution.

1. Can I beat a benchmark model such as an ensemble of regression models with deep learning? In this case, I first establish a benchmark model using traditional machine learning approaches that performs reasonably well. I then build a competing deep learning model and try to beat it. This step only uses the train data.
2. Can a quickly developed deep learning model be competitive against other solutions? Here I use the supplied test data to make predictions using the deep learning model and submit my output to Kaggle for scoring on their platform. My goal here is to determine if I can build a competitive model with limited time and resources. Competitive is defined arbitrarily as being better than the median of all results from the Kaggle competition.

5. Benchmark Model

The benchmark model will be an ensemble of various simple regression models. Although a simple linear regression could serve as a suitable benchmark for this task, I believe that a good comparison will be against a stronger candidate. Ensembling has been known to perform quite well in real world settings, including insurance. With the democratization of machine learning technologies and tools, it is getting easier for companies to apply more advanced methods like ensembling. The ensemble will be kept relatively simple and include 3 models: OLS regression, decision tree, and an elastic net. The individual model outputs will be stacked in order to create a single prediction.

The benchmark model will be self-contained within its own fitted pipeline. The pipeline will include a preliminary feature selection step using feature importance in order to reduce dimensionality. The goal is to simulate a real world data science model development and deployment process as much as possible.

6. Evaluation Metrics

I will use two metrics to evaluate my model. The first is the R^2 score to compare the deep learning model against the benchmark model. R^2 is a commonly reported metric particularly to senior management in business. The second metric I will use is the mean absolute error, which is simply the average of the absolute values of errors. Allstate used MAE for their Kaggle competition and using MAE allows for a direct comparison against the solutions offered by other data enthusiasts. I will use MAE only once on the test dataset for the Kaggle submission once I am satisfied with the final deep learning model.

Eq 1.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}$$

7. Project Design

A key benefit of using this Kaggle competition is that the data has been anonymized and is relatively clean so more effort can be focused on the application of techniques. This is a luxury that is often not available in real world data science projects where the majority of the time is often spent in

data cleansing. Although this is a significant departure from a real world data science exercise, data cleansing is not the focus of this project. I will assume this was already done.

That said, I will follow the old adage of "trust, but verify" and perform basic validations to ensure cleanliness of the data. The following outlines the proposed workflow of the study.

1. **Exploratory Data Analysis**
   Adhering to the "trust, but verify" philosophy, this step will first examine the dataset for common issues such as missing data and row index uniqueness. Then, I will move forward taking a closer look at the feature space to understand correlations and distributions for the categorical and continuous features. I will also look at the distribution of the target variable to check for outliers.

2. **Feature Engineering**
   This step includes feature engineering such as encoding categorical variables to binaries. Given the level of anonymity of the data provided, my ability to sensibly engineer new features is rather limited. Thus, only simple transformations will be considered.

3. **Feature Selection**
   Given the relatively high dimensionality of the data provided, intuition dictates that some form of dimensionality reduction or feature selection will be needed. I will apply a few common techniques, such as removing variables with low or no variance and highly correlated variables. Once the obvious are removed, I will fit a tree-based model to allow an algorithm to rank order features by importance.

4. **Build Benchmark Model Pipeline**
   As mentioned, the benchmark model will be an ensemble of simple regression models. Here I will use cross validation and search a hyperparameter space with varying numbers of predictors to determine an optimal model. Once all the simpler models are tuned and fit, a final ensemble model will be fit to the underlying model outputs to complete the benchmark model. The ensemble's cross-validation R^2 score and MAE will be calculated.

5. **Build Deep Learning Model Pipeline**
   The proposed deep learning model will be built using Keras on a dual GPU personal computer. I will try various network architectures this way to [1] learn more about deep neural networks in regression tasks, [2] use this project as a live test of a deep learning machine I am building, and [3] attempt a comparison of a full-blown deep learning architecture against a traditional CPU-based machine learning implementation (the benchmark model). The third point is to help determine if investing in deep learning is worth it.

6. **Compare Deep Learning Model to Benchmark Model**
   The deep learning model will then be compared against the benchmark model based on the final R^2 score achieved during learning. The reason for R^2 is that it is a very intuitive measure for comparing models and is often reported in business settings. The MAE will also be observed for both models as supplemental information.

7. **Predict with Test Data**
   Up to this point, the test data has not been touched in any capacity beyond downloading it from Kaggle. A suitable candidate for a deep learning model will be used to make predictions on this test set.

8. **Submit Predictions to Kaggle**
   The predictions will be formatted for submission to the Kaggle competition site. Kaggle has the actual true values of the test data labels, so this will allow me to get a measure of final generalizability of my model. More importantly, I can directly compare how my model does

against the competition. My goal is to build a competitive model, here defined as surpassing the median ranking MAE of 1125.5644 achieved during the competition. This will provide what I need to answer the ultimate question of this investigation.
Can deep neural networks be a viable solution for an automated risk assessment tool in insurance claims?

Sources

[1] https://www.soa.org/research-reports/2016/2016-accuracy-claims-based-risk-scoring-models/

[2] https://www.kaggle.com/c/allstate-claims-severity