Allied World Presentation: Motor Third-Party Liability Claims Analysis and Prediction

Yi-Pei Chan

5 Feb. 2021

Link to complete code and analysis :
https://yipeichan.github.io/claims.html

Project Concept

- Problem to solve :
  How can we predict the number of claims a policyholder would file, given his age, his car brand, and so on ?
- My approach to solve the problem :
  1. Explore the structure and properties of the dataset
  2. Choose the proper models to answer the question
- Methodology :
  After exploring the data with visualizations,
  1. Generalized Poisson Linear Model
  2. Poisson Lasso Regression, Poisson Ridge Regression
  3. Gradient Boosting Model
- Goals achieved by this project :
  1. Explored relationships between the risk factors and ranked the influences of risk factors on claim numbers
  2. Investigated the efficacy of using modern machine learning algorithms to do P&C ratemaking
  3. Make your hiring decision easier !

Data Exploration- The Dataset

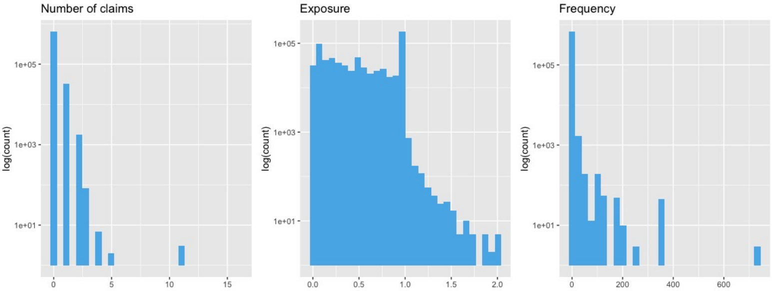
- CASdatasets Package :
  Proposed by Christophe Dutang[1] on OpenML
- Used in this study is freMTPL2freq dataset :
  1. Risk features were collected from motor third-party liability policies in France
  2. 678,013 samples, 12 explanatory variables

Variable Name	Description	Key
IDpol	Policy ID	(link with the claims dataset)
ClaimNb	Number of claims during the exposure period	
Exposure	Period of exposure (in years)	
VehPower	Power of the car	
VehAge	Vehicle age (in years)	
DrivAge	Driver age (in years)	
BonusMalus	Bonus/malus, between 50 and 350	<100: bonus; >100: malus in France
VehBrand	Car brand	Unknown categories
VehGas	Car gas	Diesel or regular
Area	Density value of the city where the car driver lives in	"A" for rural to "F" for urban centre
Density	Density of inhabitants of the city where the car driver lives in	Number of inhabitants per square-kilometer
Region	Policy region in France	

1. https ://www.openml.org/d/41214

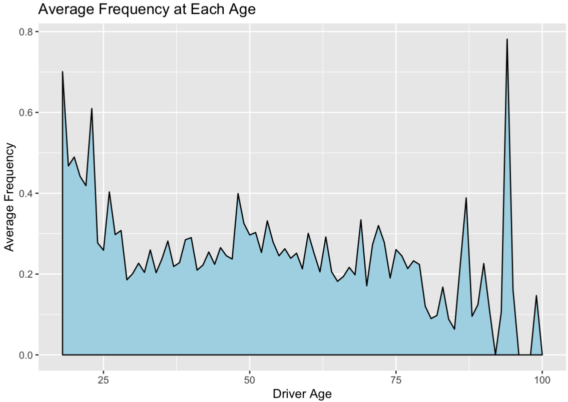
Data Exploration - Visualization

- Among the 678,013 policies, there were 34,060 filed claims, i.e. 5.02% notified claims.
- Potential Problems :
  1. Mean should equal to Variance in Poisson distribution
     $\Rightarrow$ Use Negative binomial if Overdispersed
  2. More 0s than are expected in Poisson regression ?
     $\Rightarrow$ Incorporate the logit model for predicting excess 0s
  3. Varied exposure periods (observations not comparable)
     $\Rightarrow$ Add offset of Exposure term to the model



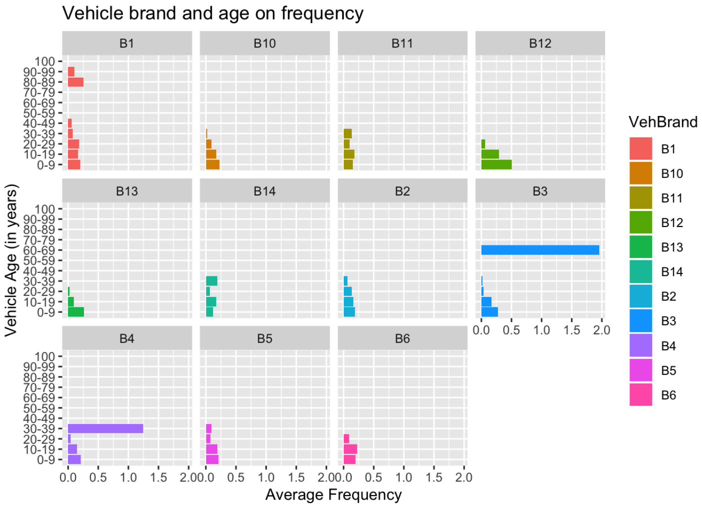
Data Exploration - Visualization

- Exposure : the duration of the insurance coverage
- Claim frequency : claim count per unit of exposure
- Did driver age influence frequency ?
  1. The highest mean frequency happens at age 94
  2. Drivers between age 18 to 23 tends to have higher mean frequency



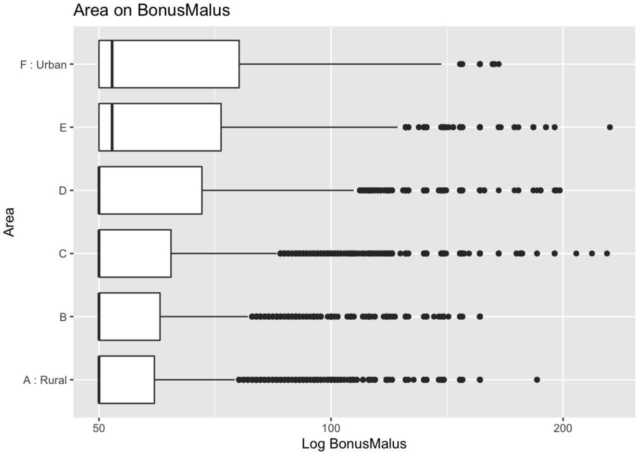
Data Exploration - Visualization

- Did vehicle brand and age influence frequency ?



Data Exploration - Visualization

- What is the relationship between area and bonus-malus ?



Model and Prediction - Poisson GLM

Before training the models, randomly select 30% of the data and set aside as testing set to find the best fitting model

- Poisson GLM Model

```
glm(formula = ClaimNb ~ VehPower + VehAge + DrivAge + BonusMalus +
    VehBrand + VehGas + Density + Region + Area, family = "poisson",
    data = data[(data$data == "train"), ], offset = log(Exposure))
```

- Statistically significant variables (Signif. level 1%) : VehAge, DrivAge, BonusMalus, VehPower, Density, etc.
- Overdispersion Test

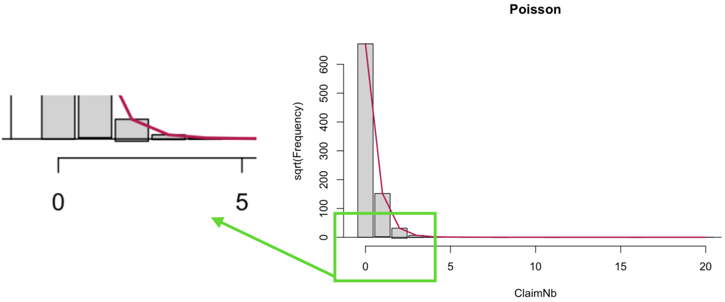
```
 Overdispersion test

data:  poissonglm
z = 3.9191, p-value = 4.444e-05
alternative hypothesis: true alpha is greater than 0
sample estimates:
   alpha
0.243334
```

1. Small p-value :The test confirms the overdispersion
2. The alpha value very close to zero : Overdispersion may not be a serious concern here

Model and Prediction - Poisson GLM

- Hanging rootogram :
  Only 2 count is a little under predicted



Model & Prediction - Poisson Lasso & Ridge Regression

```
glm.ridge$lambda.min; coef(glm.ridge, s = "lambda.min")
```

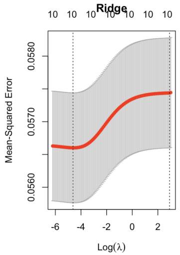
```
## [1] 0.009804138
```

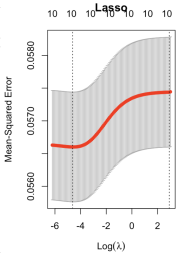
```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                         1
## (Intercept) -2.9270950727
## Exposure    -1.0400993812
## VehPower     0.0061349023
## VehAge      -0.0263678397
## DrivAge      0.0060848768
## BonusMalus   0.0169722817
## VehBrand    -0.0010265539
## VehGas       0.0502432492
## Area         0.0169615264
## Density      0.0194397589
## Region      -0.0009442549
```

```
glm.lasso$lambda.min; coef(glm.lasso, s = "lambda.min")
```

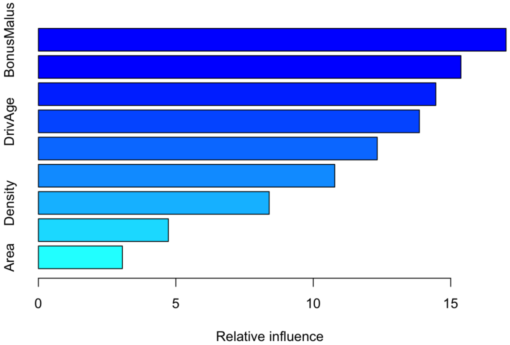
```
## [1] 0.001635429
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                        1
## (Intercept) -2.696642397
## Exposure    -1.193913018
## VehPower     .
## VehAge      -0.024586132
## DrivAge      0.006071144
## BonusMalus   0.017390359
## VehBrand     .
## VehGas       0.004379296
## Area         .
## Density      0.016603390
## Region       .
```





Model & Prediction - Gradient Boosting Model

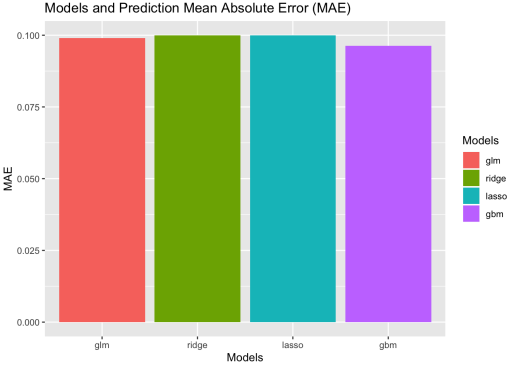



```
       var   rel.inf
BonusMalus 17.014808
    Region 15.372979
    VehAge 14.459134
   DrivAge 13.862481
  VehBrand 12.328304
  VehPower 10.782009
   Density  8.396521
    VehGas  4.728894
      Area  3.054871
```

Final Validation

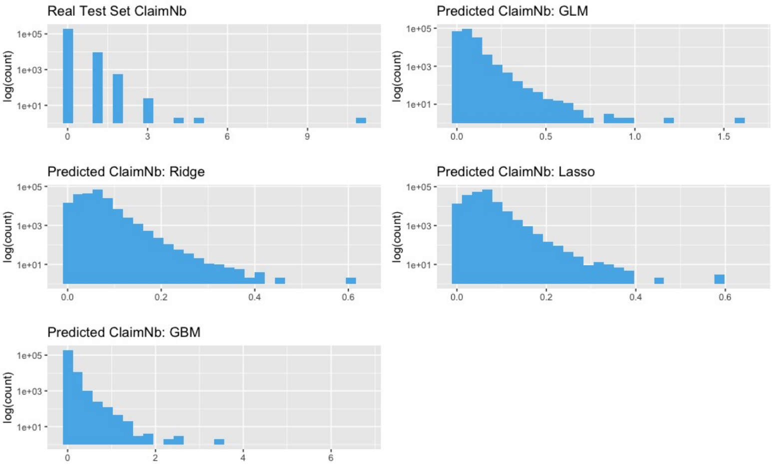
Use the test set to find the best fitting model

- The claim number prediction MAE for test set with
  1. Poisson GLM : 0.09905573
  2. Poisson Ridge GLM : 0.09988506
  3. Poisson Lasso GLM : 0.09996999
  4. Gradient Boosting Model : 0.09630762



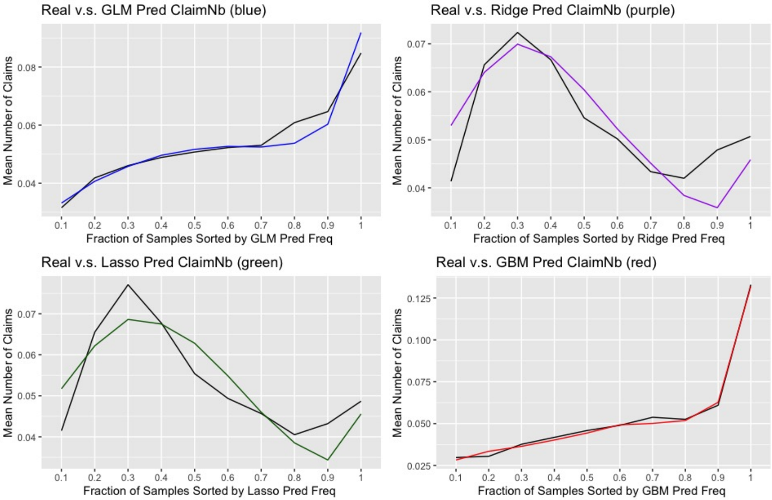
Final Validation

Evaluation of the Predicted Number of Claims in the Test Set



Final Validation

Evaluation of the Predicted Number of Claims in the Test Set



Q & A

Link to complete code :
https://yipeichan.github.io/claims.html