ITCS/DSBA 6100: Big Data for Competitive Advantage

Project and Teams

Dr. Gabriel Terejanu

Fall 2019

Important Deadlines

- Teams have been created – 17 teams & training and testing datasets available in Canvas
- The Kaggle competition has been lunched – deadline for submissions is Sun, Dec 1
- Your overall project grade will be given by 90% video presentation grade and 15% performance based on ranking
- Final project 5 min video presentations are due on Tue, Nov 26
- This video presentation will be peer evaluated and the evaluations will be due on Mon, Dec 2. Each presentation will have 8 or 9 reviewers.
- Discussions of the results and best presentations will be aired in the last day of class on Wed, Dec 4.
- Note, that each team member will also be evaluated by his/her team members and the overall project grade will be adjusted based on the contributions. These evaluations will be due on Fri, Dec 6.

Grading

- **Overall Project Grade = 90%** peer evaluations + **15%** ranking based in the private Kaggle leaderboard
  - 8-9 team members will evaluate your video presentations
  - 5 min video presentations due on Tue, Nov 26
  - Peer evaluations due on Mon, Dec 2
  - I will release a list with the presentations to be evaluated by each one of you
  - I will also release an evaluation rubric ahead of time such that you can make the presentation based on that
  - The ranking score will be linearly determined based on the minimum set by the Kaggle baseline (all portfolios has ln_LR=0) and the best performer in the class in the private leaderboard.
- Individual Project Grade = **contrib%** x Overall Project Grade
  - Contribution percentage will be obtained by averaging the scores provided by your team members with respect to your overall contribution to the project. These will be due Fri, Dec 6.
  - I will also release some guidelines on what to think when you provide the score.

Teams & First Things TO DO

- 17 teams have been created by combining two homework groups (i.e. group 1 + group 2 = team 1)
- Get organized – schedule the first meeting asap
- Discuss about each one strengths and weaknesses
- Establish a plan of action and the best way to communicate
- Set expectations and deadlines for each member
- Check/report work status frequently
- Learn more about the topic associated with your dataset and brainstorm about what can be done with the data
- Start exploring the dataset (careful with big files/datasets)
- Identify early technical challenges and propose ways to address them

Kaggle competition

https://www.kaggle.com/t/0e4caf836bfd4d9289a801cbc54bba34

You will need to create a Kaggle account and a Team, which is the same as the project team (i.e. Team 5)

Collaboration Tools

1. Agree agree beforehand on what everyone on the team expects from everyone else.
2. Communicate! Communicate ! Communicate!
   - Both onsite whenever possible and often online (your team may have one student that takes the class remotely)
   - If for some (personal) reason you are not able to accomplish your tasks please let the team know asap and find a solution
3. Take peer-evaluation feedback on mid-Nov seriously and adjust.

Dataset - DO NOT DISTRIBUTE

This dataset has been provided to enhance the learning experience that you have in this class and create a framework where you can work on an industry relevant problem. While, the dataset is anonymized to avoid information compromise, it is still proprietary material and it is restricted to this class. Thus, this dataset is for you and your teammates to develop predictive models as required by the class project. PLEASE DO NOT DISTRIBUTE THIS DATASET OUTSIDE THIS CLASS! Any such attempt will jeopardize future collaborations with industry partners, and your colleagues planning to take this class in the future will be deprived of this type of learning experience.

Training Dataset - Policy level, auto insurance

Not available in testing datasets

	Policy#	Various Attrs	Annual Premium	Claim Count	Loss Amount	Freq.	Severity	Loss Ratio
N	i	…						

$$Total\_Premium = \sum_{i=1}^{N} AnnualPremium(i)$$

$$Total\_Losses = \sum_{i=1}^{N} LossAmount(i)$$

Target: natural log of portfolio loss ratio

$$\ln\_LR = \ln\left(\frac{Total_{Losses}}{Total_{Premium}}\right)$$

Importance of Loss Ratio

1. 50% of policies are are mispriced by more than ±10% up to 50%
   - Loss Ratio can be used to reduce or increase rates
   - E.g. Permissible loss ratio = 1 – expense ratio = 0.7
   - Portfolio loss ratio = 0.666
   - Rate Change Factor = 0.666/0.7 -1 = -0.04762
   - Results in a rate reduction of 4.762%

2. Loss reserving – determines the present liability associated with future claim payments
   - Expected Loss Ratio can be used to determine the estimated losses

Testing Dataset & Objective

- A set of 330 policy portfolios, each having at least 1,000 policies with the following attributes
- Various attributes are the same as in the training dataset

Policy#	Various Attrs	Annual Premium

- Project Objective: Predict the ln_LR (the natural logarithm of portfolio loss ratio) for each portfolio in the testing dataset and submit your predictions in Kaggle.

How I have created the testing portfolios

- Decide how many policies should be in a portfolio (1000, 3000, or 5000).
- Decide the percentage of policies that have losses (1%, 2%, 3%, 4%, 5%, 6%, 7%, 8%, 9%, 10%, 20%).
- From the initial training dataset randomly draw a portfolio using the above parameters. I have 10 random portfolio samples for all the combination of the above parameters.
- Now, with a portfolio available, one can easily compute **ln_LR** (the natural logarithm of the loss ratio of the portfolio) as we have access to all premiums and losses from the training data.

Some ideas to get started …

1. **Naïve approach**
   1. Use the training dataset to build a model that given policy attributes predicts the loss of that individual policy
   2. Use the above model to predict the loss of all policies in a testing portfolio
   3. Once the losses are available, now you can compute the natural logarithm of the loss ratio of the entire test portfolio

2. **Probably a better approach**
   1. Create training portfolios similarly as I have created testing portfolios.
   2. For each training portfolio engineer a set of features that summarizes the data in that portfolio i.e. mean driver age, mean miles to work etc.
   3. Create a new training dataset using these new features and ln_LR and train a model
   4. Use these features and the model to make prediction on the testing portfolios

Dataset particularities

- The dataset contains policies for one year
- Most policies have no claims
- Watch out for outliers (e.g. claims of $1m)