Fei Shen, Dongqi Guo, Yaohua Song

team 7

HEALTH INSURANCE MARKETPLACE

1. Research:

The market place insurance data provide the health insurance form the market perspectives. A simple relationship between the insurance price you pay and the benefit money value you get is the relationship among premium rate (paid in advance), the deductibles, and out-of-pocket money (MOOP). Also, premium rates varies based on the metal level of insurance (the “rating” to define if this insurance is good), age, locations of specific plan, and the dependents information.

Most the health insurance company provide the quota for users to enter demographic information to get the premium rate. However, the base rate of premium rate is not transparent. We want to explore the secrets behind the base individual premium rate.

We used two parts in this dataset: Rate.csv and PlanAttributes.csv:

In the Rate.csv file, it records each users’ information and their insurance premium price. It contains the business year, location (based on states), age, and premium price. Also, it provided the alternative tobacco-user premium price for users consume tobaccos.

In the PlanAttributes, csv file, each plan’s meta data is recorded. All the labels are listed. If one label doesn’t apply to a certain insurance plan, a blank is used. According to health.org’s article, there’s a relationship between insurance premium, insurance plan deductibles, and plan out-of-pocket money. That’s the only numeric values in plan attributes file.

**B. Approach:**

General approach: data cleansing -> data visualization -> using neuron network to train and predict -> apply prediction and relationship weight to user information in website.

**C.Challenge:**

We download this database from Kaggle, whicThe biggest challenge is to analyze the data. The raw data is extremely complicated. All the data are scrambled together. Some attributes indicates other attributes’ values. We refer the source website documentations to analyze the data. (CMS.gov, https://www.cms.gov/cciio/resources/data-resources/marketplace-puf.html)

**d.Solution:**

Data cleansing:

We used the Hadoop Mapreduce, Python and R for data cleansing:

1. Mapreduce for Rate.csv: categorized each rate. Increase the data for each category. Format the output as( BusinessYear, Location, PlanId, Age, MarriageStatus, Depdents, Rate). The rate here ‘s based on each variables. For tobacco use, it has an alternative rate. Here the rate has been altered.
2. MapReduce for PlanAttributes.csv: separate medical plan and dental plan. For medical plan, get the deductibles, including Combined, Tier1, and out of Network deductibles; and get the Most out ofpocket money (MOOP), including combined, tier1, and out of network MOOP. Also export the Tier1 coinsurance and metal level for furture work; For dental plan, get the same information, but include an addition
3. R here’s used for file join and transform to R supported data convert. R inner joined drug csv, medical csv and dental csv with rates, to provide both individual demographic information and plan attributes information. The general format becomes (Location, Age, Rate, CombinedMOOP, CombinedDedutibles, Tier1MOOP, Tier1CombinedDeductibles, OutofNetworkMOOP, OutOfNetworkDeductibles, MetalLevel)
4. Python is used for Python supported data type convert.

Data visualization:

A series Hadoop Mapreduce to visualize data for use case and future analysis:

1. Handle data in Mapper choose year, state and individual rate column, in Reducer sort the individual rate and get the median. Use google chart(geochart) to show the median individual rate of each state in 2014,2015 and 2016.
2. Handle data in Mapper, choose age and individual rate column and set up the age range. In Reducer, sort the rate and get the median. Use Barchart to show the median individual rate of each age range.
3. Handle data in Mapper, choose year, state, tobacco and individual rate column, in Reducer sort the individual rate and get the median. Use Geochart to show the median rate of Non preference and Tobacco/Non-tobacco users in each state of 2014, 2015 and 2016.

Neural Network:

R:

This data set contains lots of empty value in numeric field, mainly because one plan doesn’t apply to this attributes. So missing values here are set to 0. Dental plan has approximate 11,000,000 observation, and the medical has 12,000,000+ observations. They both contains 10 variables.

1. R is used for PCA analysis, Linear Regression and ANN analysis. The PCA analysis the numeric attributes, which includes the deductibles and MOOP. The PCA does show the only the combined deductibles and MOOP, but some of the plan doesn’t contains the combined values for them, they contains either In-network or out-of-network files. There for in ANN, I still added all three categories for deductibles and MOOP.

#Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7

#Standard deviation 1.7975767 0.9896331 0.9697439 0.8951678 0.76690642 0.5161894 0.43933878

#Proportion of Variance 0.4616117 0.1399105 0.1343433 0.1144751 0.08402078 0.0380645 0.02757408

#Cumulative Proportion 0.4616117 0.6015222 0.7358656 0.8503406 0.93436142 0.9724259 1.00000000

1. The Linear Regression is used for deductibles, MOOP and Location, age and metal level. Age is transferred to numeric value, and location and metal level are still categorical value. The liner regression has an accuracy to predict around 70%, using min\_max accuracy method, every time I run this part.

Call:

lm(formula = Rate ~ ., data = train)

Residuals:

Min 1Q Median 3Q Max

-430.3 -87.8 -12.2 65.5 5102.7

CombInnOonIndividualMOOP -2.775e-04 2.365e-04 -1.173 0.240783

DedCombInnOonIndividual 2.302e-04 4.895e-04 0.470 0.638186

DedInnTier1Coinsurance 9.755e+00 5.849e+00 1.668 0.095322 .

DedInnTier1Individual -2.830e-03 6.959e-04 -4.067 4.78e-05 \*\*\*

DedOutOfNetIndividual 1.182e-03 3.249e-04 3.640 0.000273 \*\*\*

InnTier1IndividualMOOP -9.210e-04 4.052e-04 -2.273 0.023025 \*

OutOfNetIndividualMOOP 1.490e-03 1.215e-04 12.262 < 2e-16 \*\*\*

MetalLevelCatastrophic -6.544e+01 6.484e+00 -10.093 < 2e-16 \*\*\*

MetalLevelGold 1.844e+02 3.124e+00 59.025 < 2e-16 \*\*\*

MetalLevelPlatinum 2.604e+02 4.544e+00 57.307 < 2e-16 \*\*\*

MetalLevelSilver 8.329e+01 2.601e+00 32.027 < 2e-16 \*\*\*

---

Signif. codes: 0 ?\*\*?0.001 ?\*?0.01 ??0.05 ??0.1 ??1

Residual standard error: 156.3 on 38344 degrees of freedom

(1605 observations deleted due to missingness)

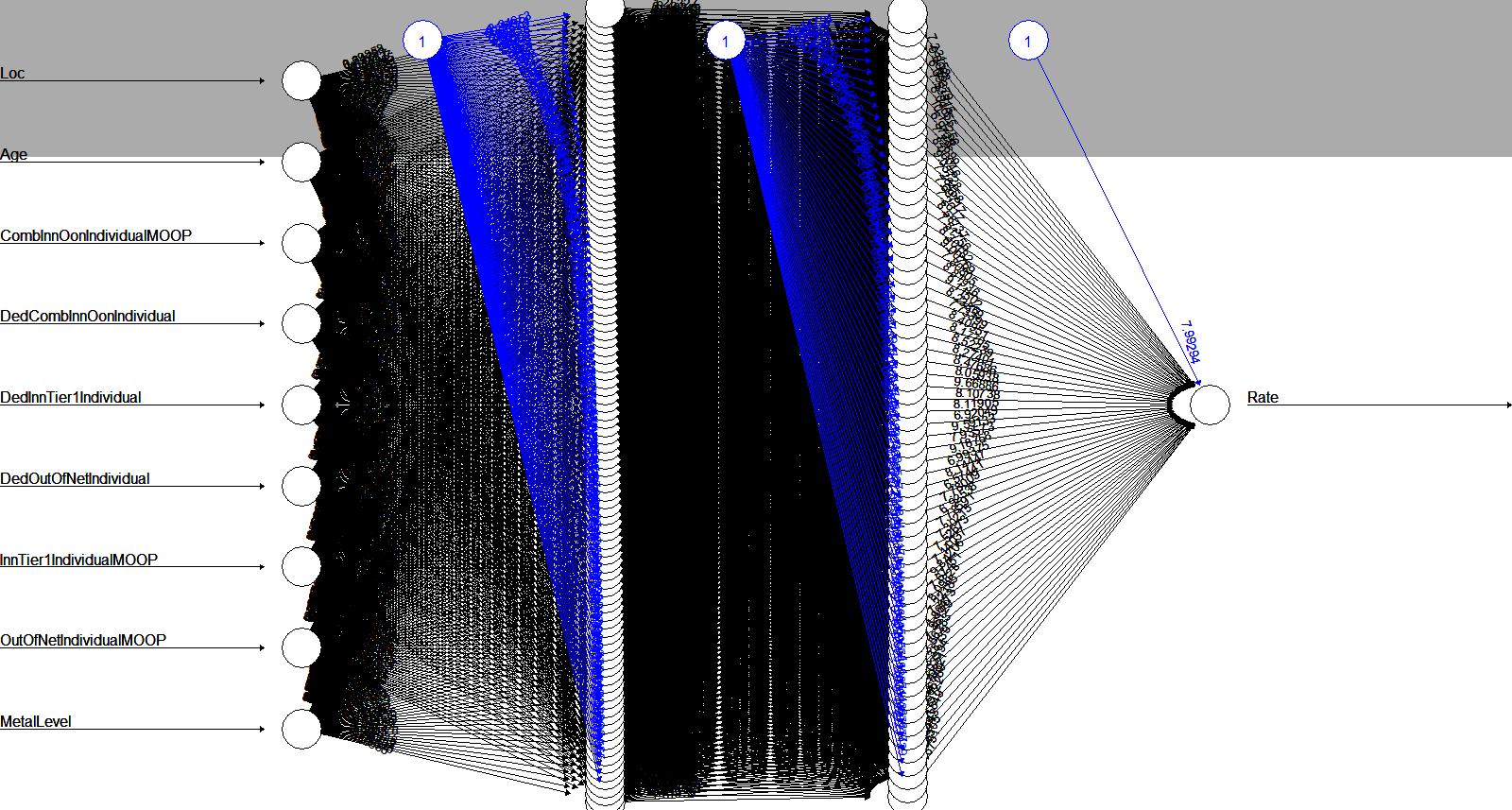
Multiple R-squared: 0.6622, Adjusted R-squared: 0.6617

F-statistic: 1503 on 50 and 38344 DF, p-value: < 2.2e-16

Min\_max\_accuracy:

[1] 0.8136303

1. R uses the neuralnet package for ANN. The first step is to use model.matrix to convert categorical values to numeric. This function uses the dummy encode method to convert categorical. Then is to scale all the data for data normalization. Then use neural network: 9 inputs, 2 hidden layers of 10:60 and 1 outputs. Threshold of 0.1. The accuracy measured in mean-square around 6. The mean-square close to 0 indicating a better result. This is a relatively large dataset, with lots of missing values, 6 is a moderate result for this model.



Python:

We used NeuPy library, which is a Neural Network library built upon theano and panda. (Yurii Shevchuk, <http://neupy.com/2015/07/04/boston_house_prices_dataset.html>)

1. Process data passed from R data cleasing, and remove the first “serial no” column.
2. Create dummy encode to convert the categorical variable( loc and metal level) to quantity result.
3. Scale the numeric attributes to normalize them. Set up training set and test set.
4. Apply the NeuPy library, to create neuron network, using a layer of (9: 30: 1), one hidden layer model to train the model. Iterates around 30 times. Do the same to the dental plan.
5. Python executes much faster than it is in R. The accuracy is around 4 based on root-mean-squared-log error algorithm, which has the results close to 0.
6. The dental plan actually has a bigger error (lower accuracy) because the missing data is more in dental plan, and some dental plan actually doesn’t follow the same relationship patterns for the medical plan.
7. The optimization here is the

Performance for medical plan

Main information

[ALGORITHM] ConjugateGradient

[OPTION] verbose = True

[OPTION] epoch\_end\_signal = None

[OPTION] show\_epoch = 25

[OPTION] shuffle\_data = False

[OPTION] step = 0.1

[OPTION] train\_end\_signal = None

[OPTION] error = mse

[OPTION] addons = ['LinearSearch']

[OPTION] update\_function = fletcher\_reeves

[OPTION] maxiter = 10

[OPTION] search\_method = golden

[OPTION] tol = 0.1

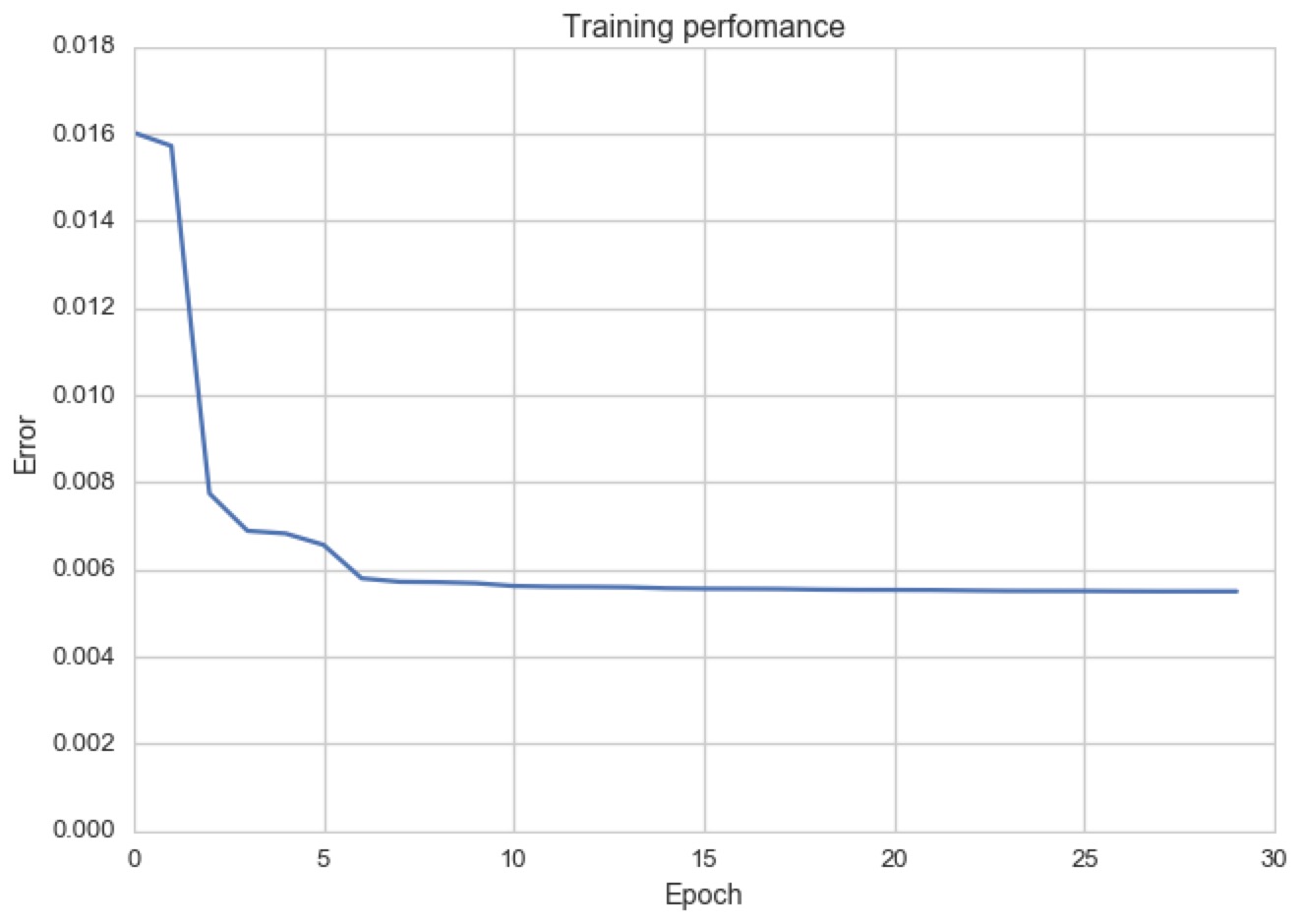
[THEANO] Initializing Theano variables and functions.

[THEANO] Initialization finished successfully. It took 0.38 seconds

Start training

[TRAINING DATA] shapes: (838859, 9)

[TRAINING] Total epochs: 30



[ 518.81 420.84 1105.53 ..., 1040.11 406.66 614.62]

[[ 448.906]

[ 448.906]

[ 448.906]

...,

[ 448.906]

[ 448.906]

[ 448.906]]

Error: 0.4791631316941596

Performance for dental plan

ALGORITHM] ConjugateGradient

[OPTION] verbose = True

[OPTION] epoch\_end\_signal = None

[OPTION] show\_epoch = 25

[OPTION] shuffle\_data = False

[OPTION] step = 0.1

[OPTION] train\_end\_signal = None

[OPTION] error = mse

[OPTION] addons = ['LinearSearch']

[OPTION] update\_function = fletcher\_reeves

[OPTION] maxiter = 10

[OPTION] search\_method = golden

[OPTION] tol = 0.1

[THEANO] Initializing Theano variables and functions.

[THEANO] Initialization finished successfully. It took 0.38 seconds

Start training

[TRAINING DATA] shapes: (838860, 9)

[TRAINING] Total epochs: 10

------------------------------------------------

| Epoch # | Train err | Valid err | Time |

------------------------------------------------

| 1 | 0.02494 | - | 9.5 sec |

| 10 | 0.02426 | - | 00:00:12 |

------------------------------------------------



[ 31.81 26.7 31.13 ..., 30.13 24.61 24.55]

[[ 99999.9]

[ 99999.9]

[ 0. ]

...,

[ 0. ]

[ 0. ]

[ 0. ]]

Error: 4.4866501085037545

[[ 0.48023036]

[-1.5032572 ]

[-1.15054085]

...,

[-0.74071615]

[-0.66144944]

[-0.04406338]]

[-0.94024881]

1. Then we apply the weights from the best performance to the data we collect from users. We will tell users what the premium should be. They will enter the age, location, deductible and MOOP information. Another tab will show them the visualization information in the website. We used a template from bootstrap for format of the website.

**E. Future work:**

We only analyze the data from individual perspectives and the meta data level. In the future, we can apply the benefit table to analyze each individual benefits in plan for each users. This will be a good use in HBASE, and the ANN can be done in parallel. Future use case can also be showing user the trend of health insurance price based on time series.

Works Cited

Shevchuk, Yurii. “ Predict prices for houses in the area of Boston.” *NeuPy Neural Networks in Python.* NeuPy.com*.* 04 Jul 2015. Web. 20 Apr. 2017. http://neupy.com/2015/07/04/boston\_house\_prices\_dataset.html

“Health Insurance Marketplace Public Use Files (Marketplace PUFs)”, *CMS.gov*, <https://www.cms.gov/cciio/resources/data-resources/marketplace-puf.html>. Web. 20 Apr. 2017.

“How insurance companies set health premiums.” *HealthCare.gov.* U.S. Centers for Medicare & Medicaid Services. Web. 20 Apr. 2017. <https://www.healthcare.gov/how-plans-set-your-premiums/>

Michy Alice, “Fitting a neural network in R; neuralnet package”. *R-bloggers*. R-bloggers.com.23 Sep. 2015. Web. 20 Apr. 2017. <https://www.r-bloggers.com/fitting-a-neural-network-in-r-neuralnet-package/>

“Health Insurance MarketPlace.” *Kaggle.com*. Kaggle.com. Web. 20 Apr. 2017. <https://www.kaggle.com/hhs/health-insurance-marketplace>

Bootstrap Greyscale resource <https://startbootstrap.com/template-overviews/grayscale/>

GoogleChars <https://google-developers.appspot.com/chart/>