Oregon State University

AI_534_001_F2021 MACHINE LEARNING

Implementation Assignment 2

Professor: Xiaoli Fern
Student: Chengxu Xu

Introduction

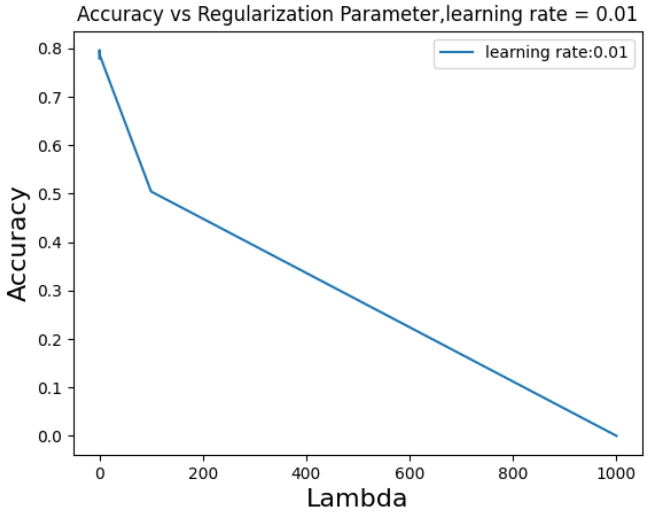
This report is for IA2 of CS534 and is roughly divided into three parts, Part1, Part2 and Kaggle. For Part1 and Part2, each trivia problem comes with corresponding run results or generated images, and the kaggle part describes my thoughts and attempts in conducting the kaggle competition.

Part 1

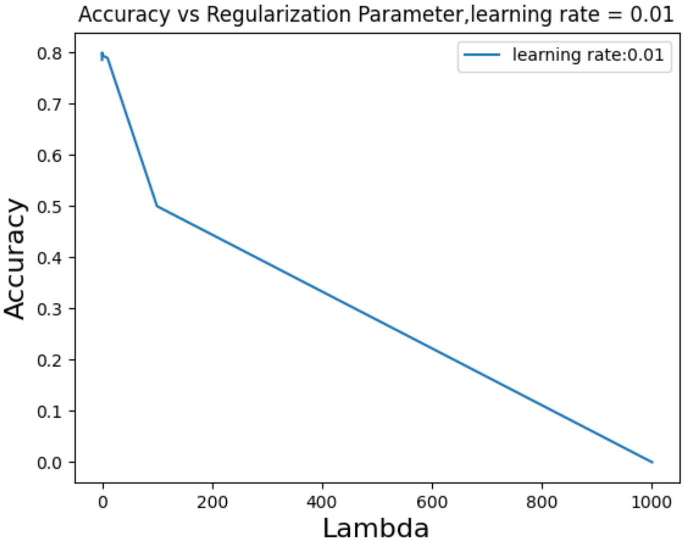
(a)

The following figure shows the images of accuracy and λ, where $\lambda \in \{10^i : i \in [-3,3]\}$.

Training accuracy VS λ:



Validation accuracy VS λ:



What trend do you observe for the training accuracy as we increase λ? Why is this the case?

In the interval $\lambda \in \{10^i : i \in [-3,3]\}$, the observed training accuracy first increases and then decreases as λ increases, and the training accuracy reaches its highest value of about 79.683% when λ takes the value of 0.1. After testing, I set the step size of the algorithm convergence to 0.001, and the minimum iteration time was 50, so that the function does not converge too fast and the algorithm can perform a certain number of iterations.

After observing the number of iterations to reach convergence under this condition, I found that the algorithm reached the highest number of iterations when λ was 0.1, which was 1771. The number of iterations for the other λ values does not exceed 100. I think one possible reason is that $\lambda = 0.1$ provides a more suitable loss value in each calculation of the loss function, so the effect of the left and right parts of the loss function plus sign on the loss is basically balanced, which brings a more suitable loss value and more iterations.

What trend do you observe for the validation accuracy? What is the best λ value based on the validation accuracy?

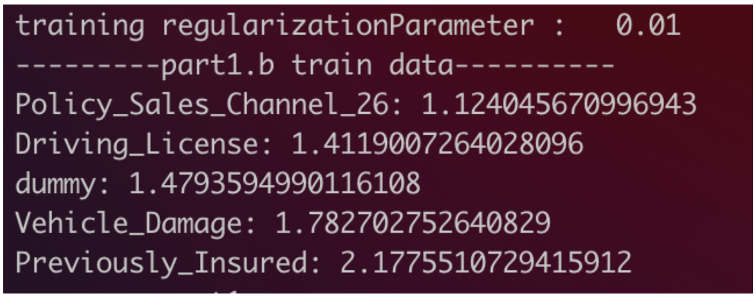
The trend of validation accuracy under different λ conditions is basically the same as that of training accuracy, which is increasing and then decreasing in the interval $\lambda \in \{10^i : i \in [-3,3]\}$. Based on the validation accuracy, the best λ is 0.1, and the validation accuracy reaches 80% under this condition.

(b)

Top 5 features with largest $|w_j|$ for λ^*, λ_- and λ_+:

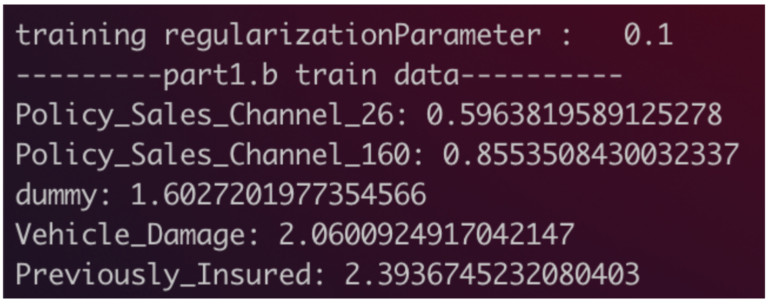
Training data:

For λ_-



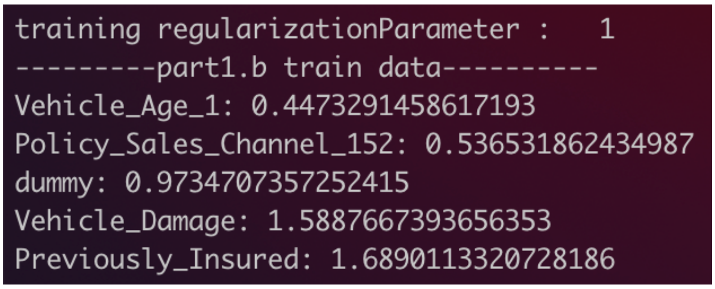
```
training regularizationParameter :   0.01
---------part1.b train data----------
Policy_Sales_Channel_26: 1.124045670996943
Driving_License: 1.4119007264028096
dummy: 1.4793594990116108
Vehicle_Damage: 1.782702752640829
Previously_Insured: 2.1775510729415912
```

For λ^*



```
training regularizationParameter :   0.1
---------part1.b train data----------
Policy_Sales_Channel_26: 0.5963819589125278
Policy_Sales_Channel_160: 0.8553508430032337
dummy: 1.6027201977354566
Vehicle_Damage: 2.0600924917042147
Previously_Insured: 2.3936745232080403
```

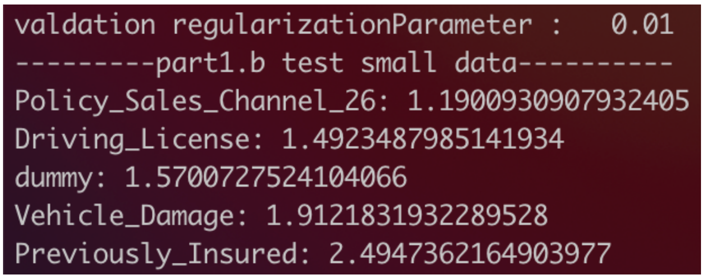
For λ_{+}



```
training regularizationParameter :   1
---------part1.b train data----------
Vehicle_Age_1: 0.4473291458617193
Policy_Sales_Channel_152: 0.536531862434987
dummy: 0.9734707357252415
Vehicle_Damage: 1.5887667393656353
Previously_Insured: 1.6890113320728186
```

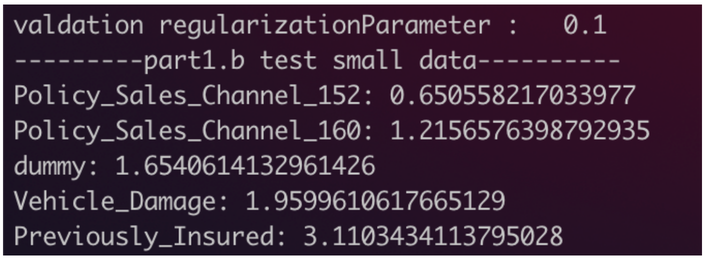
Validation data:

For λ_{-}



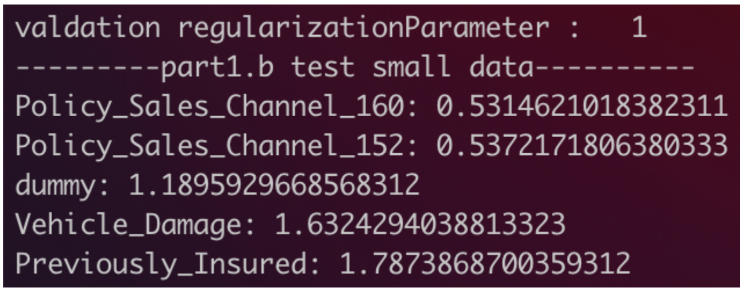
```
valdation regularizationParameter :   0.01
---------part1.b test small data----------
Policy_Sales_Channel_26: 1.1900930907932405
Driving_License: 1.4923487985141934
dummy: 1.5700727524104066
Vehicle_Damage: 1.9121831932289528
Previously_Insured: 2.4947362164903977
```

For λ^{*}



```
valdation regularizationParameter :   0.1
---------part1.b test small data----------
Policy_Sales_Channel_152: 0.650558217033977
Policy_Sales_Channel_160: 1.2156576398792935
dummy: 1.6540614132961426
Vehicle_Damage: 1.9599610617665129
Previously_Insured: 3.1103434113795028
```

For λ_{+}



```
valdation regularizationParameter :   1
---------part1.b test small data----------
Policy_Sales_Channel_160: 0.5314621018382311
Policy_Sales_Channel_152: 0.5372171806380333
dummy: 1.1895929668568312
Vehicle_Damage: 1.6324294038813323
Previously_Insured: 1.7873868700359312
```

Do you see differences in the selected top features with different λ values? What is your explanation for this behavior?

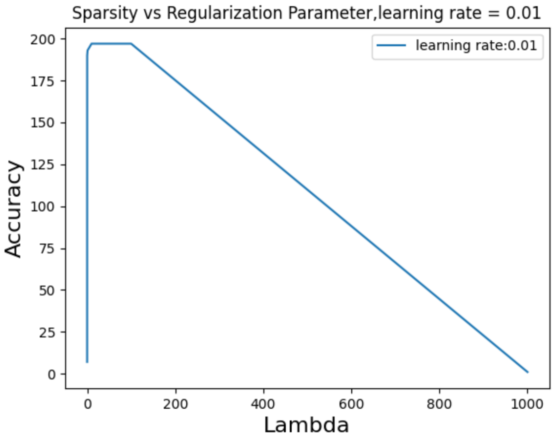
When the value of λ changes, the top 5 features change, but in general the change is not significant, while the dummy feature remains among the top 5 features.

I think one possible reason for this is that the size of λ directly affects the value of the loss function and the gradient process, so that the final generated w will have different weight distributions for different features. Meanwhile, since we excluded w0 for the dummy feature when calculating the L2 norm contribution, the dummy feature has been kept with a larger weight.

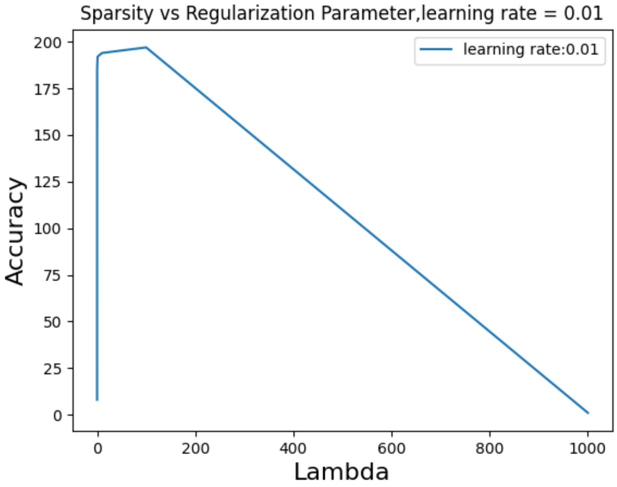
(c)

Sparisity vs λ :

For training data:



For validation data:



What trend do you observe for the sparsity of the model as we change λ? If we further increase λ, what do you expect? Why?

After observation, I found that the sparsity of the model increases and then decreases as λ increases, and then jumps up to about 190 when λ reaches its optimal value, i.e., when λ changes from 0.01 to 0.1, and then decreases to almost 0 when λ = 1000 times.

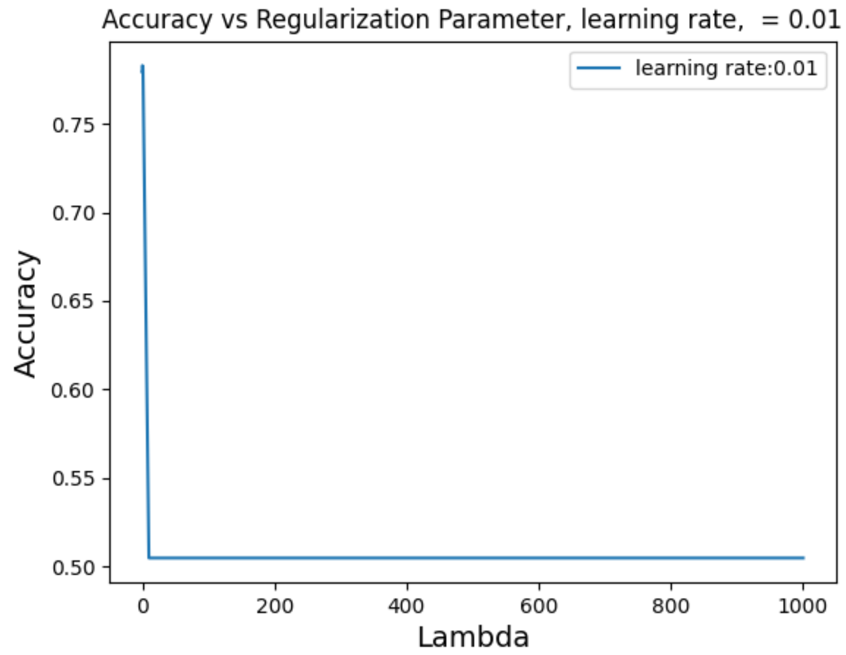
Based on the observations, I speculate that if we further increase λ, the sparsity of the model will continue to remain at 0. I think one possible reason is that the sparsity of the model is too large, and the function converges too quickly, and the weights are not yet ready to be reasonably assigned before the iterations are terminated.

Part 2

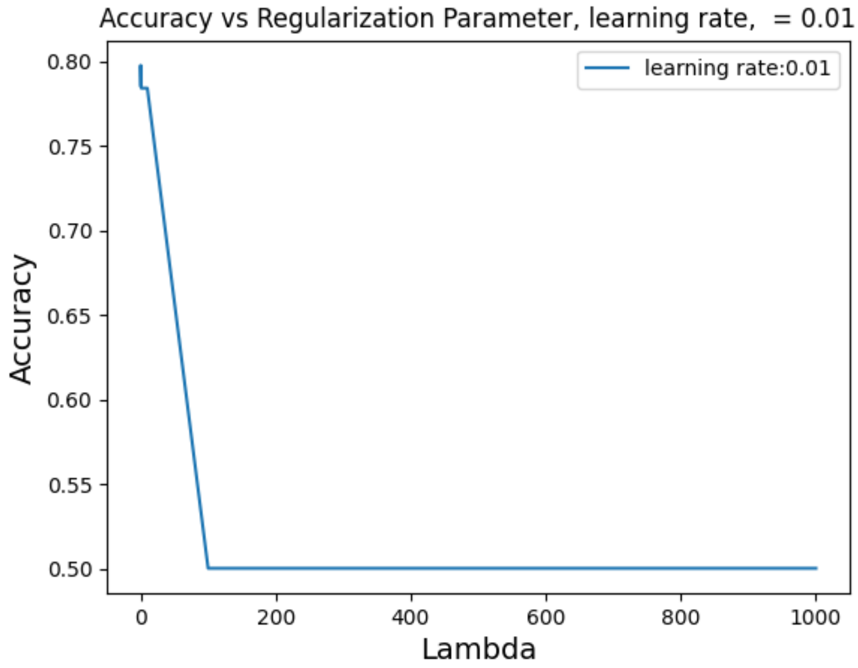
(a)

The following figure shows the images of accuracy and λ, where λ ∈ {10^i : i ∈ [-3,3]}.

Training accuracy VS λ:



Validation accuracy VS λ:



What trend do you observe for the training accuracy as we increase λ? Why is this the case?

Observing the generated plots, I found that when λ increases, the tracing accuracy of L1 regression, which follows the same trend as L2 regression, increases first and then decreases, reaching the highest accuracy of about 78.3% when λ takes the value of 0.1. I think the reason for this phenomenon is that the input λ value will affect the loss function, and when λ is too low or too high, it will cause the weight of the left and right sides of the plus sign of the loss function to change, resulting in a negative impact.

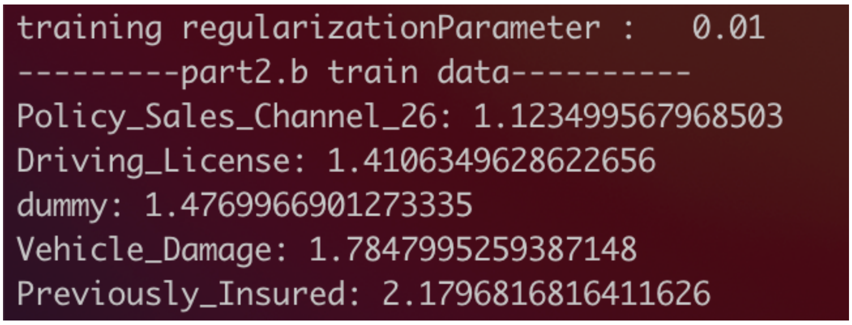
At the same time, the increase of λ is accompanied by the increase and then decrease of the validation accuracy. For L1 regression, the best λ value is 0.1, and the validation accuracy reaches 79.7%.

(b)

Top 5 features with largest $|w_j|$ for λ^*, λ_- and λ_+:

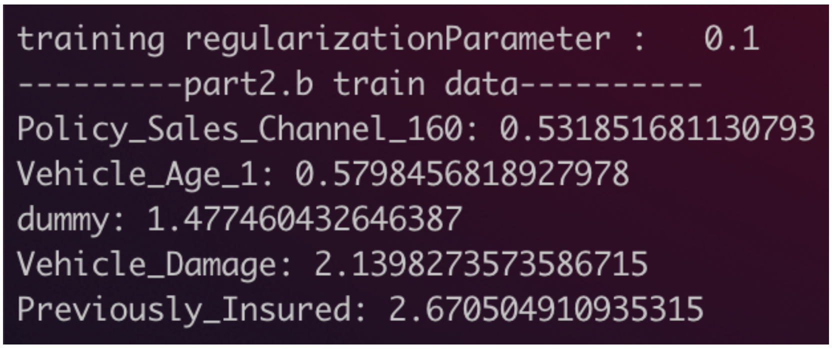
Training data:

For λ_-



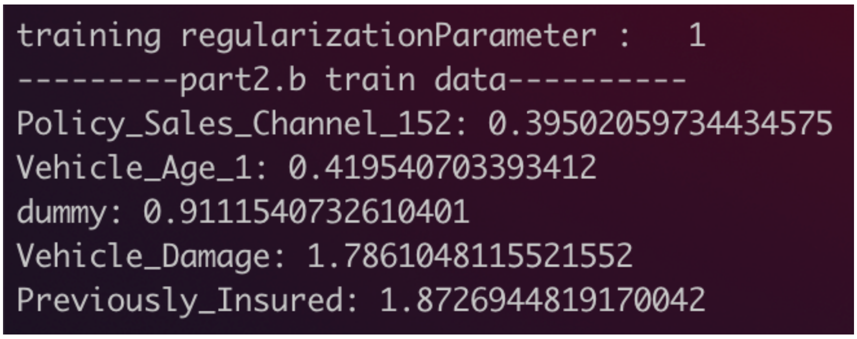
```
training regularizationParameter :   0.01
---------part2.b train data----------
Policy_Sales_Channel_26: 1.123499567968503
Driving_License: 1.4106349628622656
dummy: 1.4769966901273335
Vehicle_Damage: 1.7847995259387148
Previously_Insured: 2.1796816816411626
```

For λ^*



```
training regularizationParameter :   0.1
---------part2.b train data----------
Policy_Sales_Channel_160: 0.531851681130793
Vehicle_Age_1: 0.5798456818927978
dummy: 1.477460432646387
Vehicle_Damage: 2.1398273573586715
Previously_Insured: 2.670504910935315
```

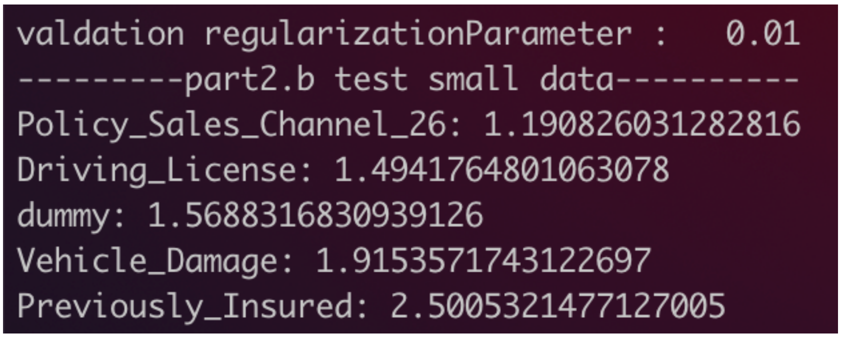
For λ_+



```
training regularizationParameter :   1
---------part2.b train data----------
Policy_Sales_Channel_152: 0.39502059734434575
Vehicle_Age_1: 0.419540703393412
dummy: 0.9111540732610401
Vehicle_Damage: 1.7861048115521552
Previously_Insured: 1.8726944819170042
```

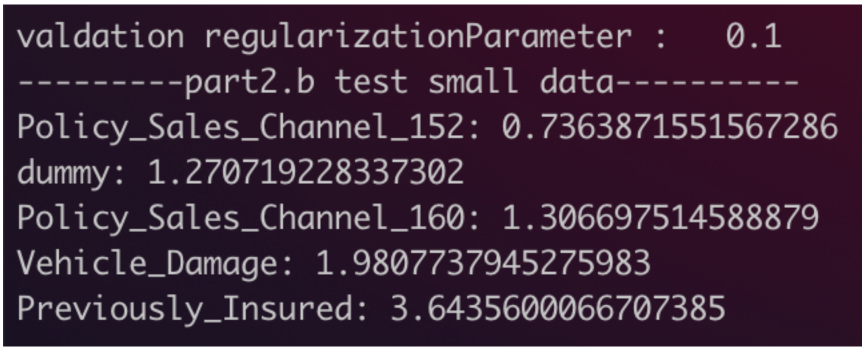
Validation data:

For λ_-



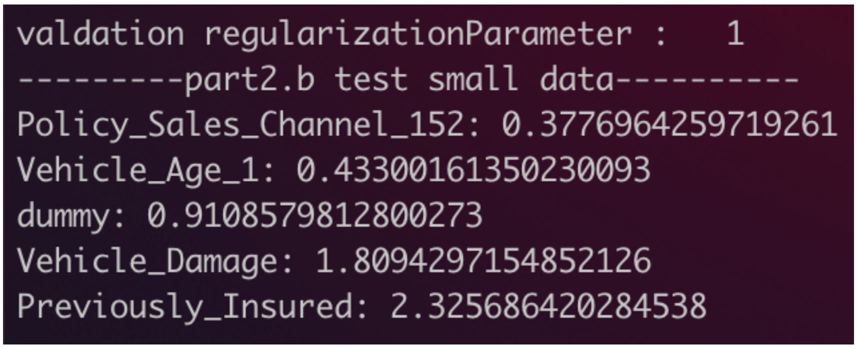
```
valdation regularizationParameter :   0.01
---------part2.b test small data----------
Policy_Sales_Channel_26: 1.190826031282816
Driving_License: 1.4941764801063078
dummy: 1.5688316830939126
Vehicle_Damage: 1.9153571743122697
Previously_Insured: 2.5005321477127005
```

For λ^*



```
valdation regularizationParameter :   0.1
---------part2.b test small data----------
Policy_Sales_Channel_152: 0.7363871551567286
dummy: 1.270719228337302
Policy_Sales_Channel_160: 1.306697514588879
Vehicle_Damage: 1.9807737945275983
Previously_Insured: 3.6435600066707385
```

For λ_+



```
valdation regularizationParameter :   1
---------part2.b test small data----------
Policy_Sales_Channel_152: 0.3776964259719261
Vehicle_Age_1: 0.43300161350230093
dummy: 0.9108579812800273
Vehicle_Damage: 1.8094297154852126
Previously_Insured: 2.325686420284538
```

Do you see differences in the selected top features with different λ values? What is your explanation for this behavior?

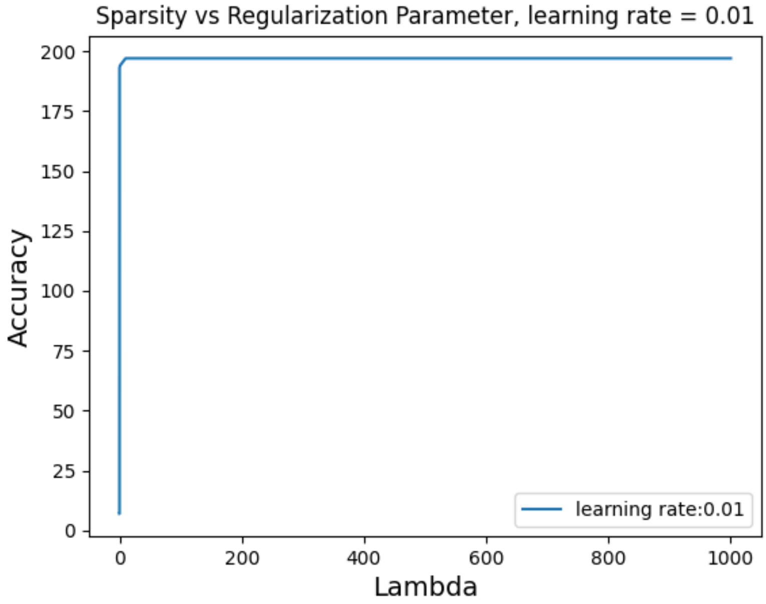
Different values of λ lead to different maximum weight features, but there are signs that features such as “dummy”, “Vehicle_Damage”, “Previously_Insured” are always important features that affect the prediction, while the weight of other "Policy_Sales_Channeles" will change with λ.

I think one possible reason is that the size of λ directly affects the value of the loss function and the gradient process, thus affecting factors such as the number of iterations, so that the final generated w will have different weight distributions for different λ. Meanwhile, the dummy feature is retained with larger weights because we exclude dummy in the calculation of L1 norn.

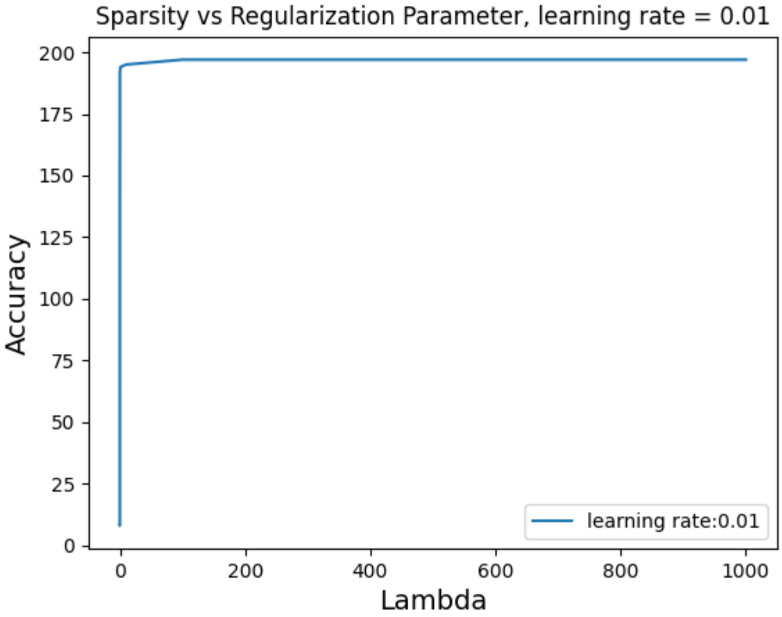
(c)

Sparisity vs λ :

For training data:



For validation data:



What trend do you observe for the sparsity of the model as we change λ? If we further increase λ, what do you expect? Is this trend different from what you observed in 1(c)? Provide your explanation for your observation.

I observe an increasing trend for the sparsity of the model as we change λ, and it increases significantly when λ takes the value of 0.1, reaching over 190

If we further increase λ, I expect that the sparsity of the model will stabilize at 197 and start decreasing when λ reaches a large value.

Compared with 1(c), the biggest difference is that the sparsity of the model remains at 197 when λ takes the value of 1000 and does not drop to 0. I think one possible reason is that for L1 regression, λ does not affect the loss function and gradient as much as L2 regression.

Kaggle

For this kaggle competition, I chose L2 logistic regression as my regression algorithm for the competition, which I tested on the public leaderboard with an accuracy of about 0.7919. The reason for choosing L2 logistic regression is that with the same learning rate and regularization parameter, L2 logistic regression shows a higher accuracy rate. Also, after the experiments in part1, I have concluded that L2 logistic regression has the best performance when the regularization parameter is about 0.1, so I set the regularization parameter of the algorithm to 0.1 and the learning rate is set to 0.01. During the test, I noticed the influence of the convergence parameter 'stopping_threshold ' on the accuracy and set the 'stopping_threshold ' to 0.001 to ensure that the convergence parameter is appropriate, and the algorithm can be iterated enough times. In addition, I normalized the input data by processing the three features 'Age', 'Annual_Premium' and 'Vintage' based on the experience of the last assignment, which also gave good results. Finally, I chose IA2-dev.csv, which has a larger amount of data, to train my model, which also brings some accuracy improvement.

I think there are many factors that affect the performance, the algorithm is the main influence, and the impact of the number of training data can not be ignored, a large number of training data can bring better accuracy performance, and finally, if the input data is properly pre-processed, it can also improve the accuracy performance.