

I. Definition

The Background

Since I want to know how to apply ML in insurance, I searched the Competitions in the Kaggle with keyword “insurance”, I find this topic _“Allstate Claims Severity”.(<https://www.kaggle.com/c/allstate-claims-severity>)

The Problem Domain and Statement

My motivation is to use ML to reduce redundant tasks in the insurance field, maybe the customer can be provided with services more quickly. The Allstate can also benefit from this point.

This project wants to create an ML model to predict claims severity. They gave us the train data with more than 130 features to predict severity. Since we want to predict a continuous target variable (loss) with many categorical features, I define this problem as supervised learning and regression problem.

Datasets and Inputs

This project contains 2 csv files. They are:

1. train.csv and test.csv features:
 - id: the index of a training set data
 - cat1 to cat116: category variables(The company will not publish the customer’s privacy information, so all column names are not provided.)
 - Cont1 to cont14: continuous variables
 - Loss(The target variable): the amount which the company pay for the customer.
2. In train.csv:
 - a) 188318 rows
 - b) 132 columns
3. In test.csv:
 - a) 125546 rows
 - b) 131 columns(the test.csv don’t have the loss column)

<https://www.kaggle.com/c/learnplatform-covid19-impact-on-digital-learning/data>

Solution Statement

- Exploratory Data Analysis

- We may use some plot method to see the features ,the correlation between different features.
- Preprocess Data
 - Since we have some categorical features, we need to convert them into numbers. So the model can use them.
 - There are many features in the train set, it may result in overfitting. So we may have to reduce the features by using PCA.
- Choose and Train Model
 - First we can use linear regression as the base model
 - We may also use XGBoost
 - To use Grid Search method test hyper parameters

Evaluation metrics

The prediction can be evaluated in several ways. Since the problem is regression type,we can use mean absolute error(MAE), mean squared error(MSE), mean squared log error, etc.

As the competition official evaluation is done by Kaggle using mean absolute error, we will use MAE as evaluation.

Mean Absolute Error

$$\text{MAE} = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j|$$

II. Analysis

Data Exploration and Exploratory Visualization

This content is done in file EDA_1.ipynb.

First we will see how the dataset looks like. They are the first 5 rows in the dataset.

	id	cat1	cat2	cat3	cat4	cat5	cat6	cat7	cat8	cat9	cat10	cat11	cat12	cat13	\
0	1	A	B	A	B	A	A	A	A	B	A	B	A	A	
1	2	A	B	A	A	A	A	A	A	B	B	A	A	A	
2	5	A	B	A	A	B	A	A	A	B	B	B	B	B	
3	10	B	B	A	B	A	A	A	B	A	A	A	A	A	
4	11	A	B	A	B	A	A	A	B	B	B	A	B	A	

	cat14	cat15	cat16	cat17	cat18	cat19	cat20	cat21	cat22	cat23	cat24	cat25	\
0	A	A	A	A	A	A	A	A	A	B	A	A	
1	A	A	A	A	A	A	A	A	A	A	A	A	
2	A	A	A	A	A	A	A	A	A	A	A	A	
3	A	A	A	A	A	A	A	A	A	B	A	A	
4	A	A	A	A	A	A	A	A	A	B	A	A	

	cat26	cat27	cat28	cat29	cat30	cat31	cat32	cat33	cat34	cat35	cat36	cat37	\
0	A	A	A	A	A	A	A	A	A	A	A	A	
1	A	A	A	A	A	A	A	A	A	A	A	A	
2	A	A	A	A	A	A	A	A	A	A	A	B	
3	A	A	A	A	A	A	A	A	A	A	A	A	
4	A	A	A	A	A	A	A	A	A	A	A	A	

	cat38	cat39	cat40	cat41	cat42	cat43	cat44	cat45	cat46	cat47	cat48	cat49	\
0	A	A	A	A	A	A	A	A	A	A	A	A	
1	A	A	A	A	A	A	A	A	A	A	A	A	
2	A	A	A	A	A	A	A	A	A	A	A	A	
3	A	A	A	A	A	A	A	A	A	A	A	A	
4	A	A	A	A	A	A	A	A	A	A	A	A	

	cat50	cat51	cat52	cat53	cat54	cat55	cat56	cat57	cat58	cat59	cat60	cat61	\
0	A	A	A	A	A	A	A	A	A	A	A	A	
1	A	A	A	A	A	A	A	A	A	A	A	A	
2	A	A	A	A	A	A	A	A	A	A	A	A	
3	A	A	A	A	A	A	A	A	A	A	A	A	
4	A	A	A	A	A	A	A	A	A	A	A	A	

	cat62	cat63	cat64	cat65	cat66	cat67	cat68	cat69	cat70	cat71	cat72	cat73	\
0	A	A	A	A	A	A	A	A	A	A	A	A	
1	A	A	A	A	A	A	A	A	A	A	A	A	
2	A	A	A	A	A	A	A	A	A	A	A	A	
3	A	A	A	A	A	A	A	A	A	A	A	B	
4	A	A	A	A	A	A	A	A	A	A	B	A	

	cat74	cat75	cat76	cat77	cat78	cat79	cat80	cat81	cat82	cat83	cat84	cat85	\
0	A	B	A	D	B	B	D	D	B	D	C	B	
1	A	A	A	D	B	B	D	D	A	B	C	B	
2	A	A	A	D	B	B	B	D	B	D	C	B	
3	A	A	A	D	B	B	D	D	D	B	C	B	
4	A	A	A	D	B	D	B	D	B	B	C	B	

	cat86	cat87	cat88	cat89	cat90	cat91	cat92	cat93	cat94	cat95	cat96	cat97	\
0	D	B	A	A	A	A	A	D	B	C	E	A	
1	D	B	A	A	A	A	A	D	D	C	E	E	
2	B	B	A	A	A	A	A	D	D	C	E	E	
3	D	B	A	A	A	A	A	D	D	C	E	E	
4	B	C	A	A	A	B	H	D	B	D	E	E	

	cat98	cat99	cat100	cat101	cat102	cat103	cat104	cat105	cat106	cat107	cat108	\
0	C	T	B	G	A	A	I	E	G	J	G	
1	D	T	L	F	A	A	E	E	I	K	K	
2	A	D	L	O	A	B	E	F	H	F	A	
3	D	T	I	D	A	A	E	E	I	K	K	
4	A	P	F	J	A	A	D	E	K	G	B	

	cat109	cat110	cat111	cat112	cat113	cat114	cat115	cat116	cont1	cont2	\
0	BU	BC	C	AS	S	A	O	LB	0.726300	0.245921	
1	BI	CQ	A	AV	BM	A	O	DP	0.330514	0.737068	
2	AB	DK	A	C	AF	A	I	GK	0.261841	0.358319	
3	BI	CS	C	N	AE	A	O	DJ	0.321594	0.555782	
4	H	C	C	Y	BM	A	K	CK	0.273204	0.159990	

	cont3	cont4	cont5	cont6	cont7	cont8	cont9	\
0	0.187583	0.789639	0.310061	0.718367	0.335060	0.30260	0.67135	
1	0.592681	0.614134	0.885834	0.438917	0.436585	0.60087	0.35127	
2	0.484196	0.236924	0.397069	0.289648	0.315545	0.27320	0.26076	
3	0.527991	0.373816	0.422268	0.440945	0.391128	0.31796	0.32128	
4	0.527991	0.473202	0.704268	0.178193	0.247408	0.24564	0.22089	

	cont10	cont11	cont12	cont13	cont14	loss
0	0.83510	0.569745	0.594646	0.822493	0.714843	2213.18
1	0.43919	0.338312	0.366307	0.611431	0.304496	1283.60
2	0.32446	0.381398	0.373424	0.195709	0.774425	3005.09
3	0.44467	0.327915	0.321570	0.605077	0.602642	939.85
4	0.21230	0.204687	0.202213	0.246011	0.432606	2763.85

We can see the continuous data's values are already between 0 and 1, so we can use them directly.

	id	cont1	cont2	cont3	\
count	150000.000000	150000.000000	150000.000000	150000.000000	
mean	234490.288887	0.493660	0.507320	0.498990	
std	134937.378713	0.187537	0.207229	0.202252	
min	1.000000	0.000016	0.001149	0.002634	
25%	117537.750000	0.346090	0.358319	0.336963	
50%	234497.500000	0.475784	0.555782	0.527991	
75%	351272.000000	0.623912	0.681761	0.634224	
max	467728.000000	0.984975	0.862654	0.944251	

	cont4	cont5	cont6	cont7	\
count	150000.000000	150000.000000	150000.000000	150000.000000	
mean	0.491802	0.487722	0.490615	0.484894	
std	0.211373	0.209136	0.205168	0.178525	
min	0.176921	0.281143	0.012683	0.069503	
25%	0.327354	0.281143	0.335580	0.350175	
50%	0.452887	0.422268	0.440945	0.438285	
75%	0.652072	0.643315	0.653958	0.590687	
max	0.952482	0.983674	0.997162	1.000000	

	cont8	cont9	cont10	cont11	\
count	150000.000000	150000.000000	150000.000000	150000.000000	
mean	0.485975	0.485215	0.497761	0.493302	
std	0.199114	0.181607	0.185666	0.209816	
min	0.236880	0.000080	0.000000	0.035321	
25%	0.312800	0.358970	0.364580	0.310961	
50%	0.441060	0.437310	0.461190	0.457203	
75%	0.623580	0.558550	0.614590	0.678924	
max	0.980200	0.993790	0.994980	0.998742	

	cont12	cont13	cont14	loss
count	150000.000000	150000.000000	150000.000000	150000.000000
mean	0.492966	0.492626	0.496251	3040.378682
std	0.209489	0.212733	0.222599	2917.478617
min	0.036232	0.000228	0.179722	0.670000
25%	0.308395	0.315758	0.294772	1202.187500
50%	0.462286	0.363547	0.409813	2115.875000
75%	0.675759	0.689974	0.724733	3869.207500
max	0.998484	0.988494	0.844848	121012.250000

We check the skewness of continuous values. We can see the loss value is positively skewed and needs to be rectified.

```

id          -0.004008
cont1       0.517909
cont2      -0.312079
cont3      -0.010250
cont4       0.415957
cont5       0.680679
cont6       0.464061
cont7       0.828456
cont8       0.679399
cont9       1.074892
cont10      0.356928
cont11      0.282640
cont12      0.293526
cont13      0.385175
cont14      0.243783
loss        3.901837

```

Next, we see the correlation between different features, we know the top 5 features have close correlation, we could apply dimensionality reduction method like PCA.

```
cont11 and cont12 = 0.99
cont1 and cont9 = 0.93
cont6 and cont10 = 0.88
cont6 and cont13 = 0.81
cont1 and cont10 = 0.81
cont6 and cont9 = 0.80
cont9 and cont10 = 0.79
cont6 and cont12 = 0.79
cont6 and cont11 = 0.77
cont1 and cont6 = 0.76
cont7 and cont11 = 0.75
cont7 and cont12 = 0.74
cont10 and cont12 = 0.71
cont10 and cont13 = 0.71
cont10 and cont11 = 0.70
cont6 and cont7 = 0.66
cont9 and cont13 = 0.64
cont9 and cont12 = 0.63
cont1 and cont12 = 0.61
cont9 and cont11 = 0.61
cont1 and cont11 = 0.60
cont1 and cont13 = 0.53
cont4 and cont8 = 0.53
```

We can also see the value distribution with the categorical features.(The visualization plot can check the EDA_1.ipynb file)

Cat1 to 72 have only two labels like A and B.

Cat73 to 108 have more than two labels, but they don't have too many.

Cat109 to 116 have many labels.

Algorithms and Techniques

Since we want to predict a continuous target variable (loss) with many categorical features, I can use the linear regression to solve this problem.

Linear Regression

We can fit a curved line passing through feature points, and use it to predict the loss value. But in this problem, we have almost 119 features, it's too many for the linear regression. We will use this algorithm as base classifier and get a base loss value, and we will try to minimize the loss value with other models.

XGBoost

XGBoost is an implementation of gradient boosted decision trees which is for speed and performance. The decision tree takes several weak features and combines them to get a strong model. Since I have this project mostly running on my laptop, I like the XGBoost execution speed and model performance.

Benchmark Model

We can see leaderboard in the Kaggle project page, the Top solution's score is 1109.70772 MAE.

Then we can use part of training data as testing data to see the linear regression model's accuracy as base. Then we compare next model to see how it works. We will choose a better model to run the test.csv, and upload the submission file to check the score.

III. Methodology

Data Preprocessing

As we just say, loss column is positively skewed. We will take log of the loss + shift. Shift is another hyper-parameter. It gives better performance. I found it from the discussion(<https://www.kaggle.com/c/allstate-claims-severity/discussion/24611>)

After log the loss value, it is normalized.

Since there are 114 columns which are categorical features, we will make them numbers. Also I use 150000 entries as train data and the other 38318 entries as test data from train.csv.

Implementation

There are my steps with linear regression algorithms:

1. We import linear regressor from sklearn and also mean absolute error.
2. We train the model
3. We predict the results of the test set
Since we transform target value into $\log(\text{loss} + \text{shift})$, we have to convert the predict result back.
4. We calculate the MAE between the predict values and actual values.

Here are my steps to implement the XGBoost model.

1. We preprocess the data and divide it into training and testing data.
2. We implement the XGBoost model.

I have to notice the tuning processing. Since I have used out my AWS account's budget, I can't tune the hyper-parameters on AWS, I have tuned it manually.

Refinement

We got a base result(1267.48) with linear regression algorithms, and obtained result(1149.75) with naive XGBoost algorithms, which is better than the base result.

Next we need to find the optimal parameters for the model. Since I have used out my AWS account's budget, I can't tune the hyper-parameters on AWS, I have tuned it manually.

The first two parameters which I use are max_depth and min_child_weight. And I run the program on my friend's cloud computer for once. Finally, I got the better result with the parameters as follow:

Max_depth = 6 and min_child_weight = 8

At last, we use these parameters and reduce the learning rate to 0.01.

IV. Results

Model Evaluation and Validation

At last, we got the result 1132.3 by tuning the parameters. It's better than linear and naive XGBoost algorithms.

Then I check the kaggle score, and submit the file. Finally we got the result 1120.234.

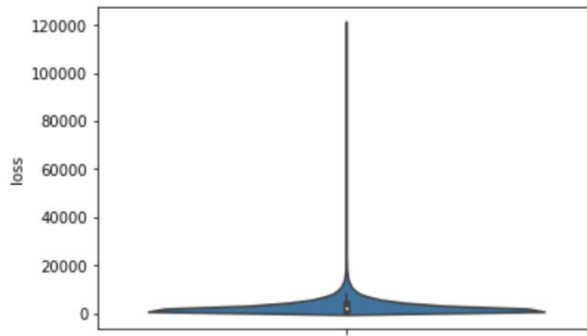
Justification

First the score is better than our base algorithms performance. And the result almost is Rank 500 of 3000(total teams), about first top 20% level. I am satisfied about the result.

V. Conclusion

Free-Form Visualization

We can see the loss value's violin plot , it can be clearly seen that it is skewed.



Reflection

Since all the features have no meanings and I don't have enough domain knowledge, however for sake of academic purposes, I am satisfied with the result.

Improvement

1. I could use the AWS SageMaker to train the model automatically.
2. PCA could be used to reduce the features. But we don't know the features' meaning, it may bring uncertainty.

References

1. Hands-on Machine Learning with scikit-Learn, Keras & TensorFlow, chapter 2
2. Comparing different metric

<https://medium.com/usf-msds/choosing-the-right-metric-for-evaluating-machine-learning-models-part-2-86d5649a5428>

3. Tuning XGBoost

<https://www.kaggle.com/c/allstate-claims-severity/discussion/24611>