A Replicative Study of Recommender Systems for Insurance Marketing

Luke Strassburg
University of Minnesota Twin Cities
Minneapolis, United States of America
stras134@umn.edu

ABSTRACT

This is a replication paper of Recommender Systems for Insurance Marketing [1]. This paper describes implements all the association rules and recommendation modeling techniques described in the original paper, expanding the analysis to be from the perspective of a marketing professional in the insurance industry. Addition of the metrics accuracy, PR-AUC, Precision, Recall, and F2, in addition to AUC give an accurate representation of how the various models make their decisions. Usage of the models to recommend cross-selling opportunities of products to existing customers can lead to enhanced profitability for insurance companies.

ACM Reference Format:
Luke Strassburg. 2025. A Replicative Study of Recommender Systems for Insurance Marketing. In . ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

1.1 Impact on the Insurance Industry

The foundation of the insurance industry is managing the balance between selling as many policies to customers as possible, while also managing risk liability. Within existing clientele, there exists a high potential for cross-selling multiple policies across various fields, home, auto, life, etc., to existing clients who only hold one or two of them. This potential extra profit from selling multiple policies to the same user is the foundation of how insurance companies expand their market cap. Discovering who to attempt to sell more policies to, and which ones, is a key issue that separates marketing professionals. Constantly pushing existing users to buy more policies is a good way to annoy and potentially lose existing customers, but not attempting to sell any more policies leads to a stagnant and profitless business.

1.2 Recommender Systems

Recommender systems are the fundamental algorithmic modeling procedures used to generate recommendations and strategically market items to users. In the insurance industry, this relationship is represented as recommending various insurance policies to clients based on their fundamental characteristics and prior purchase history. Using the recommendations from a recommender system simplifies the marketing strategy. Specifically analytically identifying high probability client, policy pairs that are likely to be sold. This minimizes the risk of frustrating and potentially losing clientele, while maximizing the ability to generate more profit.

In Recommender Systems for Insurance Marketing [1] they develop a algorithmic framework to develop such a model. Using actual anonymized data from an insurance company's policy holders from 2 sets of time, separated by a few months, for 10 million policy holders. They treat whether a policyholder holds that specific policy at a given time as an "Implicit Rating" for that policy. They use this data to create 10 algorithmic frameworks split across traditional recommender system algorithms, matrix factorization techniques, supervised learning models, and machine learning models. Each of these algorithmic frameworks is tasked with identifying users probabilities of owning a product and evaluated using AUC. They then choose the model with the best AUC for each policy as what they will present probabilities for. These probabilities can be directly interpreted by marketing professionals to yield actionable insights in selling more policies.

Table 1: Insurance Product Possession: Period One

Policy	A	B	C	D	E	F	G	H	I	L
Owned	0.78	0.22	0.01	0.21	0.02	0.02	0.02	0.01	0.02	0.03

A key issue with Recommender Systems for Insurance Marketing's [1] is the distribution of products within the policyholder dataset. As shown in 1, 7 of the 10 policies found within the shared subsetted 10000 policy holder dataset are held by less than 5 percent of customers. This sparsity for these items indicates that they are potentially undersold within the clientele. This means any strong analytical indicators for actionable insights should be taken by marketing professionals to increase their sales within these policies. This need for actionable insights is promoted by the low policy purchase rates between period 1 and 2 for the policies in the dataset 2.

Table 2: Policy Purchases Between Period 1 and 2

Policy	A	B	C	D	E	F	G	H	I	L
Bought	0.04	0.03	0.0	0.02	0.00	0.00	0.00	0.00	0.00	0.00

In this replication, all the models implemented by Recommender Systems for Insurance Marketing's [1] with an emphasis on providing metrics that focus on various aspects of model predictive

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

power, including finding more true positives within the modeling procedure at the expense of some incorrect predictions, such as accuracy, precision, recall, PR-AUC, and F2-Score.

2 METHODOLOGY

2.1 Dataset

The provided dataset from [1] has information for 10000 policy holders, including ownership for stages 1 and 2 of 10 insurance policies, 15 categorical identifiers, and 14 numeric identifiers. It also includes a set flag which indicates whether each data point was considered to be a part of the train, validation, or test sets throughout modeling.

2.2 Problem Formulation

The fundamental idea behind this research is to identify relationships that could gain marketers an advantage or insight in determining potential marketing strategies. There are multiple ways to go about doing this. This could be done on an item level, looking for associated products that are often purchased together. It could also be done through a model designed to make specific predictions for user item combinations. These models lead to more specificity in recommendations, but are more prone to errors and likelihood of false positives.

2.3 Association Rules

Association rules represent item-item relationships and are used in data mining information from itemsets. In the case of the insurance recommender, a policy holders portfolio acts as a single itemset and association rules are generated from the itemsets considered in aggregate. The Apriori algorithm (arules package, R) was used to identify itemsets within the data, specifically with the following requirements, also used in [1]:

- Minimum Rule Length of 2
- Minimum Support of 0.01 (Percent Occurrence of Rule Across all Itemsets)
- Minimum Confidence of 0.2 (Percent Occurrence of RHS when all LHS Present)

2.4 Evaluation Metrics

The original paper in [1] used AUC as the only metric they used for comparing models accuracy. While AUC is a great metric in classification frameworks to identify model predicting power and lift, in the intended framework of using recommendations to educate potential business opportunities it is lacking the full picture. Including other metrics such as accuracy, precision, recall, pr-auc, and f2-score help give a greater overall picture of the types of recommendations being made by the models. All of these metrics emphasize finding true positives.

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F2 = \frac{5 * Precision * Recall}{4 * (Precision + Recall)} \tag{4}$$

Association rules obviously could not be coerced into a suitable style to fit into these metrics, but all the other models listed in the replication implement all of these metrics. AUC and PR-AUC are computed using probabilities of the positive class which is why there is some variation in them within models seemingly making the same classification decisions.

2.5 Item-Item Collaborative Filtering

Item-Item Collaborative Filtering is a popular fundamental recommender system approach (recommenderlab, R). It chooses to recommend items by generating similarity profiles between items based on users ratings of both items for users who have rated both items. The fundamental idea behind the model is that if a user likes item A for instance and many users who liked A and also purchased item B also like item B, recommending item B to the original user could be a good idea.

2.6 Matrix Factorization Approaches

Matrix Factorization approaches are a common technique to fill in sparse datasets. They follow the general format of multiplying vectors together to yield an approximate value for the missing value.

$$\hat{R} = UV \tag{5}$$

In recommender systems, U is generally considered to be the user vector while V is considered to be the item value. Interactions and relationship values are learnt through training and populate the U and V vectors. Throughout the replication, 4 matrix factorization techniques were used:

- Library for Factorization Machines (LibFM)
- Generalized Low-Rank Model (GLRM)
- Alternating Least Squares (ALS)
- Sparse Linear Methods (SLIM)

LibFM (recosystem, R) was trained using a grid search for its parameters k, loss function, learning rate, and regularization terms. GLRM (h20, R) was also trained using a grid search for its parameters k, x gamma, and y gamma. ALS (implicit, Python) does not use any grid search for its parameters. SLIM (slimrec, R) uses a grid search for its alpha parameter.

2.7 Logistic GLMS

Logistic GLMS (sklearn, Python) are a type of logistic regression model that allows for error terms associated with predictions rather than the straight up predictions typically associated with logistic regression models. A 5 fold cross validation grid search was used to determine the optimal Logistic GLM involving the c and regularization parameters.

2.8 Machine Learning Algorithms

Machine learning models are a popular new method of recommender system modeling. Machine learning models take in attributes about a user and use them to extrapolate predictions. This

is an entirely different method from the matrix factorization techniques or IBCF which rely completely on information from the ratings matrix. 4 machine learning models are implemented within this replication.

- Category Boost (CTB)
- Deep Learning (DL)
- LightGBM (LGB)
- XGBoost (XGB)

Category Boost (catboost, Python) is implemented with set parameters and is a gradient boosted model that specializes in categorical features. Deep Learning (tensorflow, Python) is implemented with embeddings for each of the categorical features which are learnt throughout the neural network. LightGBM (lightgbm, Python) is also a gradient boosted model that specifically specializes in runtime. XGBoost (xgboost, Python) is also a gradient boosted model that involves regularized learning to prevent overfitting which is a common problem in most gradient boosted models.

3 RESULTS

Table 3: Association Rules for Policies in Stage 1

LHS	RHS	Support	Confidence	Lift	Count
(A, D)	(B)	0.04	0.42	1.90	394
(B, D)	(A)	0.04	0.84	1.08	394
(A, B)	(D)	0.04	0.22	1.06	394

Table 4: IBCF Results

Item	Accuracy	AUC	PR-AUC	Precision	Recall	F2
A	0.22	0.47	0.75	0.65	0.01	0.01
B	0.76	0.51	0.25	0.35	0.02	0.02
C	0.98	0.57	0.02	0.00	0.00	0.00
D	0.79	0.51	0.21	0.27	0.02	0.02
E	0.97	0.57	0.03	0.03	0.02	0.02
F	0.97	0.51	0.02	0.00	0.00	0.00
G	0.96	0.52	0.03	0.05	0.03	0.03
H	0.98	0.52	0.01	0.00	0.00	0.00
I	0.97	0.55	0.03	0.09	0.04	0.04
L	0.96	0.52	0.04	0.07	0.01	0.01

Table 5: LibFM Results

Item	Accuracy	AUC	PR-AUC	Precision	Recall	F2
A	0.65	0.51	0.77	0.78	0.78	0.78
B	0.66	0.51	0.21	0.22	0.22	0.22
C	0.99	0.50	0.01	0.00	0.00	0.00
D	0.66	0.51	0.21	0.18	0.17	0.17
E	0.98	0.58	0.01	0.00	0.00	0.00
F	0.98	0.58	0.03	0.00	0.00	0.00
G	0.98	0.53	0.02	0.00	0.00	0.00
H	0.99	0.51	0.00	0.00	0.00	0.00
I	0.98	0.56	0.03	0.00	0.00	0.00
L	0.97	0.49	0.03	0.00	0.00	0.00

Table 6: GLRM Results

Item	Accuracy	AUC	PR-AUC	Precision	Recall	F2
A	0.29	0.51	0.78	0.77	0.13	0.16
B	0.73	0.49	0.23	0.24	0.12	0.14
C	0.89	0.52	0.01	0.02	0.17	0.07
D	0.73	0.50	0.22	0.22	0.10	0.11
E	0.91	0.56	0.02	0.01	0.06	0.03
F	0.92	0.57	0.03	0.05	0.13	0.10
G	0.93	0.55	0.02	0.03	0.06	0.05
H	0.95	0.58	0.00	0.00	0.00	0.00
I	0.94	0.54	0.02	0.00	0.00	0.00
L	0.94	0.49	0.03	0.02	0.02	0.02

Table 7: ALS Results

Item	Accuracy	AUC	PR-AUC	Precision	Recall	F2
A	0.22	0.50	0.78	0.00	0.00	0.00
B	0.78	0.50	0.25	0.00	0.00	0.00
C	0.99	0.50	0.01	0.00	0.00	0.00
D	0.79	0.50	0.21	0.00	0.00	0.00
E	0.98	0.50	0.02	0.00	0.00	0.00
F	0.98	0.50	0.02	0.00	0.00	0.00
G	0.98	0.50	0.02	0.00	0.00	0.00
H	0.99	0.50	0.01	0.00	0.00	0.00
I	0.97	0.50	0.02	0.00	0.00	0.00
L	0.97	0.50	0.03	0.00	0.00	0.00

4 DISCUSSION

4.1 Association Rules

Looking at 3 we can see that three association rules were found within the subset dataset. All three rules come from the same itemset A, B, D. These also happen to be the items with individual probabilities greater than 0.05. Looking at the confidence values themselves we can see that having items B and D yields a probability of 0.84 of also having item A. This is a really strong rule and is definitely an actionable item for marketing policy A. Any confidence above 0.2 is considered significant in the context of this evaluation, so owning two of the policies from itemset A, B, D should result in a recommendation to sell for the third policy.

4.2 Item-Item Collaborative Filtering

Item-Item Collaborative Filtering surprisingly struggled a ton with this predictive evaluation 4. This is probably due to there not being a ton of items available for analysis (Neighborhood value of k-2) and quite a small dataset, only 1000 ratings. None of the items had an AUC value above 0.6 and some of them even had below baseline guessing rates (0.5) with A having a value of 0.47. One thing to point out here though is that the precision and recall rates are non-zero,

Table 8: SLIM Results

Item	Accuracy	AUC	PR-AUC	Precision	Recall	F2
A	0.25	0.70	0.73	0.81	0.22	0.26
B	0.78	0.56	0.30	0.00	0.00	0.00
C	0.99	0.64	0.06	0.00	0.00	0.00
D	0.79	0.66	0.20	0.00	0.00	0.00
E	0.98	0.56	0.05	0.00	0.00	0.00
F	0.98	0.66	0.02	0.00	0.00	0.00
G	0.98	0.52	0.06	0.00	0.00	0.00
H	0.99	0.59	0.03	0.00	0.00	0.00
I	0.98	0.67	0.07	0.00	0.00	0.00
L	0.97	0.46	0.05	0.00	0.00	0.00

Table 9: Logistic GLM Results

Item	Accuracy	AUC	PR-AUC	Precision	Recall	F2
A	0.78	0.59	0.83	0.78	1.00	0.95
B	0.78	0.53	0.23	0.00	0.00	0.00
C	0.99	0.52	0.01	0.00	0.00	0.00
D	0.79	0.67	0.32	0.14	0.00	0.00
E	0.98	0.61	0.02	0.00	0.00	0.00
F	0.98	0.58	0.03	0.00	0.00	0.00
G	0.98	0.69	0.05	0.00	0.00	0.00
H	0.99	0.67	0.01	0.00	0.00	0.00
I	0.98	0.69	0.05	0.00	0.00	0.00
L	0.97	0.50	0.04	0.00	0.00	0.00

Table 10: Category Boost Results

Item	Accuracy	AUC	PR-AUC	Precision	Recall	F2
A	0.78	0.64	0.86	0.78	1.00	0.95
B	0.78	0.58	0.28	0.00	0.00	0.00
C	0.99	0.56	0.02	0.00	0.00	0.00
D	0.79	0.71	0.36	0.50	0.00	0.00
E	0.98	0.72	0.05	0.00	0.00	0.00
F	0.98	0.64	0.04	0.00	0.00	0.00
G	0.98	0.72	0.06	0.00	0.00	0.00
H	0.99	0.66	0.02	0.00	0.00	0.00
I	0.98	0.76	0.07	0.00	0.00	0.00
L	0.97	0.66	0.09	0.00	0.00	0.00

Table 11: Deep Learning Results

Item	Accuracy	AUC	PR-AUC	Precision	Recall	F2
A	0.78	0.62	0.85	0.78	1.00	0.95
B	0.28	0.55	0.25	0.22	0.93	0.57
C	0.96	0.61	0.02	0.04	0.11	0.09
D	0.49	0.67	0.32	0.28	0.88	0.62
E	0.92	0.64	0.03	0.04	0.18	0.11
F	0.71	0.60	0.04	0.03	0.45	0.13
G	0.72	0.54	0.03	0.03	0.37	0.11
H	0.98	0.60	0.02	0.03	0.12	0.08
I	0.91	0.63	0.04	0.07	0.23	0.16
L	0.92	0.68	0.07	0.10	0.21	0.17

Table 12: LightGBM Results

Item	Accuracy	AUC	PR-AUC	Precision	Recall	F2
A	0.78	0.62	0.84	0.78	1.00	0.95
B	0.78	0.58	0.28	0.00	0.00	0.00
C	0.99	0.57	0.01	0.00	0.00	0.00
D	0.79	0.69	0.35	0.00	0.00	0.00
E	0.98	0.68	0.05	0.00	0.00	0.00
F	0.98	0.61	0.03	0.00	0.00	0.00
G	0.98	0.68	0.04	0.00	0.00	0.00
H	0.99	0.48	0.01	0.00	0.00	0.00
I	0.98	0.71	0.07	0.00	0.00	0.00
L	0.97	0.67	0.08	0.00	0.00	0.00

Table 13: XGBoost Results

Item	Accuracy	AUC	PR-AUC	Precision	Recall	F2
A	0.78	0.63	0.84	0.78	0.99	0.94
B	0.78	0.60	0.29	0.41	0.04	0.05
C	0.99	0.55	0.01	0.00	0.00	0.00
D	0.78	0.71	0.36	0.37	0.06	0.07
E	0.98	0.70	0.05	0.00	0.00	0.00
F	0.98	0.69	0.07	0.00	0.00	0.00
G	0.98	0.74	0.05	0.00	0.00	0.00
H	0.99	0.59	0.01	0.00	0.00	0.00
I	0.98	0.77	0.07	0.00	0.00	0.00
L	0.97	0.67	0.10	0.00	0.00	0.00

which is not the case for many of the models. IBCF is not afraid to recommend products to people. This would make this model a good tool for marketers to identify potential sales, even if the accuracy is not as good.

4.3 Matrix Factorization Techniques

LibFM 5 and GLRM 6 get very similar results to IBCF. LibFM gets a really good precision and recall for A but the AUC is still around the baseline. ALS 7 ended up regularizing too much and predicted every policy to not be owned. This is not a good model and is basically guessing. SLIM 8 has good AUC values, 3-4 above 0.6, even if its precision and recall are 0 for most policies. The higher AUC values indicate it is picking up on some trends, due to the class probability foundation of the AUC formula. This would honestly be a good model if the threshold for determining whether a policy is to be recommended or not is set lower at something like 0.3 instead of 0.5.

4.4 Logistic GLMS

Logistic GLMs 9 gets pretty average results. The high precision and recall values for A are intriguing but the AUC value is lower and the basically 0 recall values for all the other policies are discouraging.

4.5 Machine Learning Algorithms

Category Boost 10 and LightBGM 12 get basically the same results and they also mirror the results obtained in 9. XGBoost 13 has non-zero precision and recall values for the three most common policies (A, B, D). With high paired AUC values, most of them greater than 0.6, this is a much better model than its other counterparts. In the context of trying to get the most accurate model with some predictive power for true positives, this is probably the best model. The best model, however, for the context of cross-selling policies to existing customers is the Deep Learning model 11. With accuracies lower than almost all of the other models, it may not seem like this is a great modeling framework. Deep Learning has non-zero precision and recall values for all of the items. This is huge for marketing purposes because the model is actually recommending products to people. Also the accuracy is still quite good for all the policies except for B and D.

5 CONCLUSION AND FUTURE WORK

5.1 Overall Conclusions

The results of the models across the board were quite varied. Some models like ALS 7 managed no learning and recommended "do not sell" for every item. Others like Deep Learning 11 recommended a decent percentage of the true positives to be sold to the customers, which is essential to making and expanding profits as an insurance companies on their existing customer base. Ultimately the IBCF, SLIM, DL, and XGBoost models all have advantages to their prediction methods compared to each other, and there is an argument to select any of them over the other. One potential option is treating the model selection process as an ensemble. For instance, take all the models from Deep Learning except B and D which it struggles with and take them from XGBoost which yielded good results on those policies.

5.2 Future Work

One area of analysis that was not part of the initial replication that could be of potential interest in the future is analyzing the Bought category in between timezone 1 and timezone 2. The entire point of designing models like this is to model policy holder behavior, and drawing information directly from their actions using their characteristics could yield even better ideas about who is a direct audience for the various policies. A classification modeling framework using the 4 machine learning methods used in this paper along with logistic GLMS and random forest would likely be sufficient for this project.

REFERENCES

[1] Giorgio Alfredo Spedicato and Giuseppe Savino. 2022. Recommender Systems for Insurance Marketing. *Variance* 15, 1 (jan 27 2022).