National Research University Higher School of Economics
Faculty of Computer Science
Programme 'Master of Data Science

Master's Thesis

SAINT for Credit Default Risk: Application of SAINT architecture and pre-training for different segments on tabular Credit default data

Student: Bhupendra Dubey
Supervisor: Eldar Ganbarov

Moscow 2021

SAINT for Credit Default Risk: Application of SAINT architecture and pre-training for different segments on tabular Credit default data

A Project Report Submitted by

Bhupendra Dubey

NATIONAL RESEARCH UNIVERSITY

Higher School of Economics

December 23, 2021

Contents

Abstract

Loan default prediction is an important problem in Banking and finance. The financial and social data used for this task is generally heterogeneous tabular data. Deep neural networks generally work very well in Natural Language Processing and computer vision applications or wherever the data is homogeneous but not for tabular data which is heterogeneous. Here we survey how SAINT architecture (Somepalli et al., 2021), which has both self-attention and inter-sample attention and create embedding of categorical as well as continuous features in the data followed by feeding to transformer modules as described in in the paper "Attention is all you need" (Vaswani et al., 2017).

We try to compare the performance of SAINT architecture which is a deep learning architecture for tabular data on loan default prediction with classical algorithms like Logistic regression, Decision trees and powerful ensemble models such as XGBoost.

One of the challenges in banking is the lack of enough data for training in the case of some segments (e.g., when the loan amount is high). Models trained on fewer data may not generalize well. In such scenarios, we propose using pre-training on already present data from other segments. We study the effect of self-supervised contrastive pre-training vs no pre-training. We also compare our SSL pre-training with fully supervised pre-training.

SAINT seems to be overperforming classical models and self-supervised learning-based pre-training does seem to help in giving better results compared to no pre-training scenario.

1 Introduction

1.1 Credit default risk

Lending is the most important vertical in the banking industry and is the primary source of revenue for banks and other lending institutions. However, lending as a business carries serious risks. The borrower is contractually obligated to pay interest on the loan amount and also the principal. In case the borrower defaults on the periodic payments the bank stands to lose money if the loan was unsecured or not sufficiently secured.

The proportion of bad loans is a serious indicator of the health of lending institutions. A high proportion of bad loans create a significant liquidity crisis for banks and can result in the complete collapse of the bank. Therefore it becomes imperative for banks and all other types of lending institutions to accept a loan application only after thorough due diligence on repaying capacity of the borrower.

Key factors in accepting a loan application depend upon the financial history of the borrower like credit card history, banking transactions, assets, liabilities and cash flows. Other factors depend upon loan terms and structure like interest rate, equated monthly installment (EMI), loan tenure etc. In cases where the repaying capacity of the borrower is in question, the lending institution may decide to reject the loan application.

1.2 Self supervised learning

Self-supervised learning learns the representational representation of data from the data without explicit labels. The learning task is set so that supervisory signals are generated from the training data itself.

When the labelled data is scarce and at times present only for a few segments, a common technique is to learn from unlabeled data which is available in abundance and then fine-tune on labelled data. Self-supervised learning can be used to learn the representational representation of data of majority unlabeled

data across data in self-supervised part and then fine-tuning can be applied on target data where data is scarce but labels are available.

1.3 Different client segments and pre-training

One of the critical problems to solve to mitigate credit default risk is to predict loan seekers likelihood of defaulting on the payments in case the loan is approved and granted.

Another important challenge in banking is that sometimes there is not enough data for a particular segment of loan applications. For example, there might be a segment of clients seeking very large loans whereas most of the clients for which data is available to learn are small clients. There could also be a cold start problem when a new loan product is launched or new segment of clients is introduced. Due to lack of any history on these new product/segments there is no data which can be used to train new models which can help analysing the risk a accepted loan application poses. In such cases rather than training the model on smaller segments, it might be beneficial to pre-train the model on segments where data is available in large quantities and fine-tune the model on the target segment.

1.4 Deep learning and tabular data challenges

One of the characteristics of banking data is its heterogeneous tabular nature. The features come from various unrelated sources, each having it's own meaning and unit. The data contains categorical, ordinal as well as continuous features. The features may be very sparse even within a column. This makes it quite different from image data where nearby pixels are highly correlated.

Also, there is no inherent ordering in various features as present in textual data. Thus the features are present in relatively very high dimensions and are neither dense or continuous. These limitations make it difficult to apply powerful deep learning techniques which are already state of the art in Natural Language Processing(NLP) and computer vision in applying for loan default

prediction. Hence most of the popular models in use for this problem in the industry are not based on neural networks.

There has been focus on improving the limitation posed by tabular data for deep learning and quite a few improvements have been made in this regard by introducing techniques like attention and some of the qualities of tree based algorithms (Arik and Pfister, 2020; Popov et al., 2019).

2 Related Work

2.1 Credit default risk

The use of machine learning to predict creditworthiness has been extensively studied and applied. Authors of Addo et al. (2018) explore the importance of features in predicting creditworthiness. The Paper also explores the reliability of metrics to evaluate credit default risk. One of the important conclusions in Addo et al. (2018) is that algorithms based on artificial neural networks may not perform on credit default data. The paper explores algorithms like Logistic Regression, Elastic Net, Random Forest and deep learning-based models.

Galindo and Tamayo (2000) explores CART decision-tree models for credit default risk prediction and also compares the result with neural networks and k-nearest neighbours.

Huang et al. (2004) explores the uses of support vector machines(SVM) for the problem of credit default prediction and compares it with back propagation neural networks (BNN) in the banking domain.

2.2 Deep learning on Tabular data

Deep learning in tabular data remains less popular compared to other modelling approaches due to various limitation we discussed earlier. However there has been quite a few noteworthy researches in recent times that have tried to address some of those limitation and have attempted to harness the power of deep neural networks in modelling tabular data.

Recently TabNet Arik and Pfister (2020) introduces the novel deep artificial neural network-based architecture for tabular data which uses a sequential attention mechanism to focus on the most salient features at each step.

TabTransformer Huang et al. (2020) is another transformer-based model where categorical features are converted to embeddings using transformer blocks. Continuous features are not encoded and are just appended with encoded categorical features and fed to a neural network. So the drawback is continuous features are not considered while calculating attention.

NODE(Neural Oblivious Decision Ensembles) Popov et al. (2019) is another end to end deep learning method for tabular data. NODE is made up of oblivious decision trees(ODT) each with same depth. The salient feature of these oblivious decision trees compared to other trees is that they are differentiable and hence the gradient can be back propagated.

Value Imputation and Mask Estimation(VIME) Yoon et al. (2020) is a frame work for both self and semi supervised learning for tabular data. The authors propose denoising based pretext task for both semi and self supervised learning.

SAINT Somepalli et al. (2021) is another deep tabular architecture that has attention over both rows and columns of tabular data. One of the issues with TabTransformer Huang et al. (2020) where attention is not calculated for continuous features is addressed in this paper by converting both categorical and continuous features to higher dimension embeddings and then passing over transformer blocks.

2.3 Self supervised learning

To get around for training data scarcity, a common technique is to learn from unlabeled data which is available in abundance and then fine-tune on labelled data. The rationale is that the network learns a useful representation from unlabeled data first and then improved upon by fine-tuning on final data.

Self-supervised learning is sort of something in between supervised and unsupervised learning. There are no labels that are used for learning but instead,

some supervisory signals are generated from the data itself. SubTab Ucar et al. (2021) introduces a novel method where instead of using all features, only subset of features are used to reconstruct either the subset or the whole set of features. Author demonstrate use of this technique in both constrastive(by denoising) and in non contrastive fashion.

Self-supervised pre-training has been shown to improve robustness and uncertainty in downstream tasks (Hendrycks et al., 2019) other than solving the problem of lack of enough labelled data for training.

As such self-supervised learning generally lags behind fully supervised learning as far as accuracy is concerned but as studied in Lan et al. (2020), looking at accuracy alone is not correct to discount self-supervised learning in comparison to fully supervised learning with labels. Self-supervised learning has been shown to greatly improve robustness and uncertainty against label corruption, adversarial inputs, out of distribution detection. So rather than pitching supervised learning against self-supervised learning both could be used in conjunction to yield more desirable results.

2.3.1 Contrastive Self supervised Learning

For the unsupervised part generally, contrastive pre-training is used which is based on the idea that a similar input pair(positive) should have output indicating similarity and a dissimilar(negative) pair should have output indicating dissimilarity. This means in a well-trained model distance in the embedded space is maximized for dissimilar output and minimized for similar outputs.

Input data is augmented and paired to generate training samples. For augmentation authors of Somepalli et al. (2021) which explores neural networks for tabular data also uses contrastive loss for self-supervised learning. In the paper, the author described the use of CutMix introduced in Yun et al. (2019) on input data and again use MixUp as in Zhang et al. (2018) after passing through an embedding layer. VIME Yoon et al. (2020) also proposes denoising based contrastive self supervised learning for the pretext task.

2.3.2 Non-Contrastive Self supervised Learning

Non-contrastive self-supervised Learning differs from contrastive self-supervised Learning in the fact it uses only positive pairs and minimizes the distance between them (Tian et al., 2021).

Recent work on this BYOL Grill et al. (2020) is have shown even without contrastive pairs significant learning can be achieved. In this paper author present a pair of neural network which interact with each other and learn. The two networks are called online and target network

2.4 Self Attention

Self-attention is a concept which dictates that some part of the sequential input should be given more weight compared to others while predicting the next outcome in sequence. Transformer models which were introduced by Vaswani et al. (2017) employs this concept of self-attention and encoder-decoder architecture to sequential input and now form basis of many state of the art developments that have superseded recurring neural network based model.

Transformer models are now proven to be giving significantly better results than Recurrent neural network models(RNNs), which is another popular modelling technique for sequential inputs.

As of now most states of the art models in natural language processing tasks are based on transformers eg. BERT and GPT-3 (Devlin et al., 2019; Brown et al., 2020).

3 Architecture

Here we will give a brief overview of SAINT architecture as described in Somepalli et al. (2021). SAINT is composed of stacked together attention blocks as described in Vaswani et al. (2017). There are two types of attention blocks one is self-attention and other being inter-sample attention. The inter-samle attention is simply the encoder described in Vaswani et al. (2017). Inter-sample attention

block is also similar with the only difference being the attention layer in it is inter-sample. Which simply means for the attention the samples in a batch are used. Each stage consists of both self and inter-sample attention blocks and there are many such stages.

Figure 1: SAINT Architecture (Source: Somepalli et al. (2021)

(a) Attention block based on (Vaswani et al., 2017). Out of two attention blocks one is self-attention blocks which computes attention within the sample and other is inter sample attention introduced in (Somepalli et al., 2021).

(b) For un-supervised pre-training contrastive and denoising loss is minimized between sample and it's views by CutMix (Yun et al., 2019) and mixup (Zhang et al., 2018). In the supervised fine-tuning step embeddings of the sample are passed through SAINT.

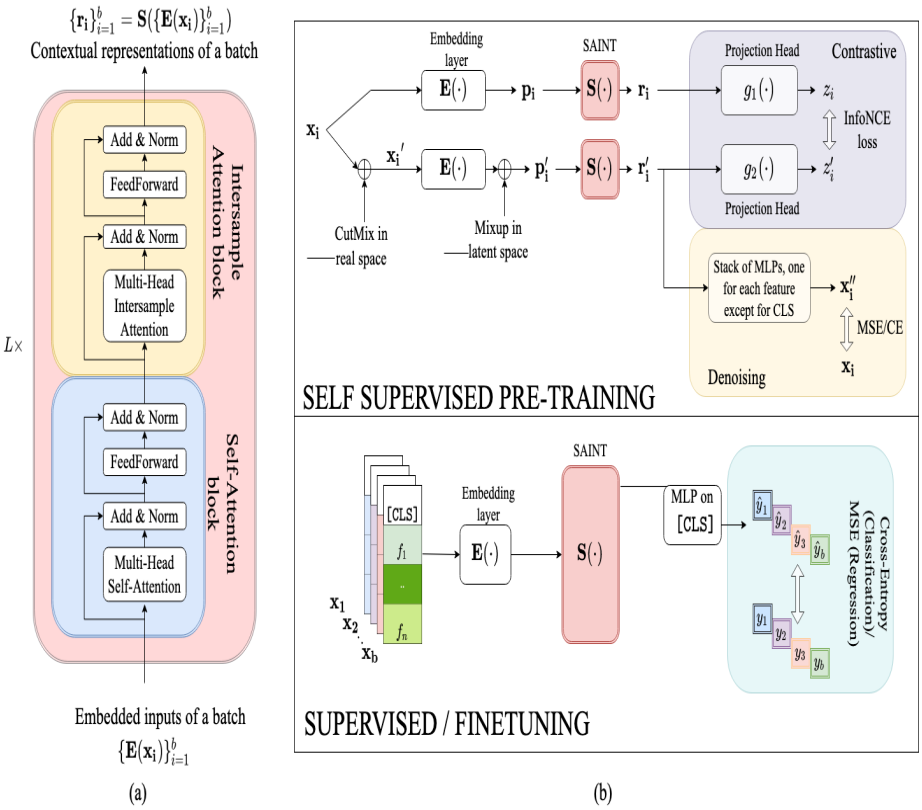
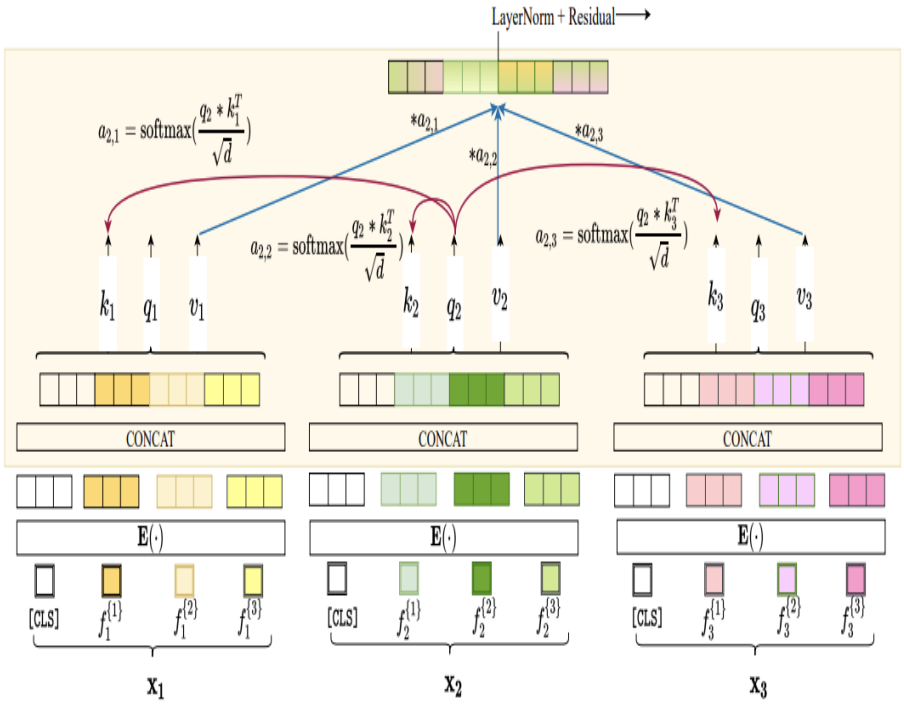


Figure 2: Inter-Sample attention (source Somepalli et al. (2021))

Intersample attention on a batch (batch size = 3)



4 Experiments

4.1 Dataset

For these experiments, we used a subset of LendingClub data. Lending club is a platform where individuals can lend money to other users on the platform. Various features specific to loans and the individuals like interest rate, principal, past credit statistics, financial status etc are included. The data set contains both accepted and rejected loans.

For our experiments, we will use only accepted loans. The data contains a binary field 'default' indicating whether the borrower eventually defaulted on the loan.

Table 1: Lending club data

	#samples	#features	#default percentage
train	1128702	38	0.2
val	240864	38	0.26

One important observation is that the for higher value loan amount we have lot less data. Since the dataset is huge we will use subsets of the data described in the following sections of our experiments.

Figure 3: Histogram for loan amount

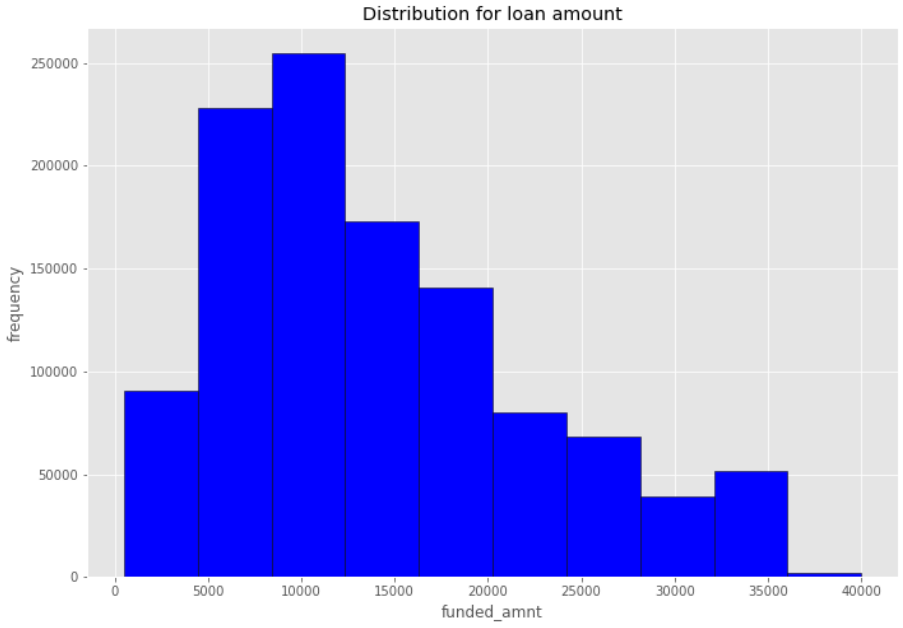


Figure 4: Histogram for annual income

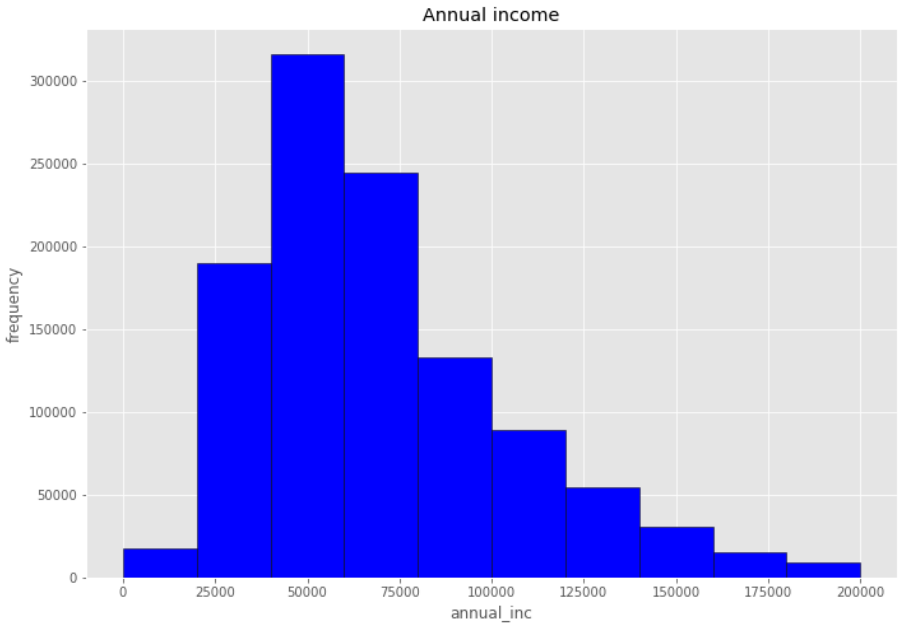
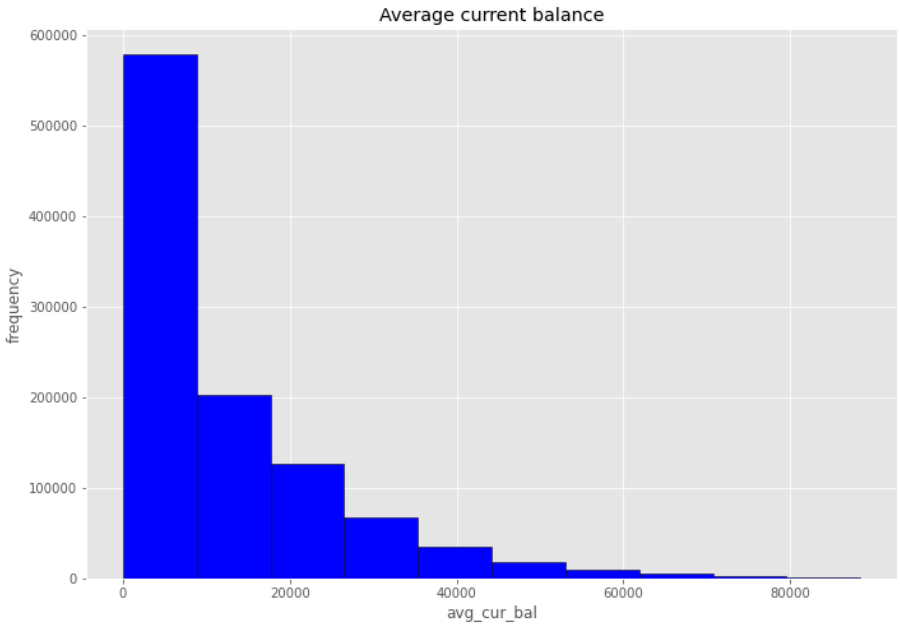


Figure 5: Histogram for annual income



4.2 Research goals

Our goal is to effectively learn whether a prospective borrower will default on their loan payment based on customers financial history and loan features. Here we will survey SAINT Somepalli et al. (2021) which is a neural network-based architecture on tabular data.

We will set up the problem as binary classification with cross-entropy as loss between the true value and predicted values as loss function. We will pursue two goals as part of our research. One where we evaluate the performance of SAINT against classical logistic regression, Decision trees and powerful ensemble models like XGBoost.

In other set of experiments, we explore self-supervised learning as proposed in (Somepalli et al., 2021) using contrastive pretraining in cases where we do not have enough training data for the segment we want to develop a model for. Lack of data due to segment being thin or due to cold start problems on launch of new loan products or introduction of new segments is big challenge in banking industry.

4.2.1 Performance of SAINT

The deep learning model for tabular heterogeneous data is still not very popular compared to classical models. Here we compare performance using selected metrics for SAINT on the data-set with the existing popular method. We are dealing with a binary classification problem here. We compare SAINT being applied to tabular data in banking against the performance of Logistic regression, Decision trees and XGBoost.

4.2.2 Self-supervised pre-training on Loan default data

Here we explore how self-supervised learning can help in learning on segments where data is scarce. We do this by pretraining on majority data and then fine-tuning on smaller target segment data.

Here we compare evaluate the performance of contrastive self-supervised pre-training with no pre-training. We also evaluate how contrastive un-supervised pre-training as introduced in (Somepalli et al., 2021) for tabular data stack up against regularly supervised pre-training for cases where we already have labels for some sections of customers and we want to apply our models on a different section.

4.3 Metrics

We have set up our problem here as binary classification where we will be predicting whether a loan applicant will default or not based on the client banking data we have. To compare different approaches we evaluate based on AUROC(Area Under the Receiver Operating Characteristics).

The area under ROC curve (AUROC) of 1 means the model has excellent differentiating power between two binary classes. An AUC of 0.5 means the model is random and has no predicting power. For inference, we also calculate and observe AUPRC(Area Under the Precision-Recall curve).

5 Results

5.1 Saint comparison

5.1.1 Training

We will use SAINT architecture and train for about 300 epochs selecting the best model based on the best AUROC. We use AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.9$ and $decay = 0.01$. We use batch size of 256 and a learning rate of 10^{-3} for our training.

5.1.2 Data subset

We randomly sample 10% samples of LendingClub data from both the train and test set and use that for our experiments here. We use *pandas.sample* with

random seed of 101 in python for sampling.

Table 2: Data points for SAINT evaluation

	#samples	#features	#default percentage
train	11287	38	0.2
val	2408	38	0.26

5.1.3 Experiment Results

We compared results of SAINT against other popular models Logistic regression, Decision trees and XGBoost and recorded various metrics to compare. What we observe is SAINT over performs Logistic regression, Decision Trees and XGBoost in both AUROC and AUPRC.

Table 3: SAINT vs non neural network model

Model	AUROC	AUPRC
SAINT	0.7	0.42
LR	0.68	0.39
Decision tree	0.65	0.37
XGBoost	0.68	0.40

5.2 Self-supervised learning and pre-training

Here we attempt to first train our model on segments of clients which are in the majority category with a large set of data and try to fine-tune the resulting model on a smaller segment of clients with a lot less data. The idea is that the representational representation of data learned in the pre-training step will help us to start with better weights before final fine-tuning on the target segment.

Lack of training data is a common problem faced in the banking industry that all segments of clients are not equally represented and thus not enough data is available for training them. It can also be valuable in solving the cold start problem when some new segments of loan products are launched and hence there is no past data for them available to train any model.

5.2.1 Training

For pre-training SAINT in self-supervised mode with contrastive loss, we train for 50 epochs. For fine-tuning we train for about 150 epochs selecting the best model based on AUROC.

We use $AdamW$ optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.9$ and $decay = 0.01$. We use batch size of 256 and a learning rate of 10^{-3} for our training.

5.2.2 Data subset

Here we tried to explore how pre-training can help when we are interested in client segments who are extreme based on some parameter such as loan amount or cash flows of the client. Naturally for these extreme values loan application data is in scarcity so we will create the pre-training set from non-extreme value data which is available in larger quantities.

For our target set of extreme loans, we used a threshold of over 80 percentile and 90 percentile for the loan amount, annual income and average current balance of all accounts. Threshold on each of these features will make separate sets of experiments. To generate a subset for pre-training experiments we did the following steps to select a subset for experiments from whole LendingClub data.

1. **Train:** Randomly sampled 70% of all points above 80 percentile threshold of the selected feature from the train set in first threshold experiment. For other threshold experiment we randomly sampled all points above 90 percentile threshold of the selected feature from the train set, We use *pandas.sample* with random seed of 101 in python for sampling.

2. **Test:** Randomly sampled 70% of all points above 80 percentile threshold of the selected feature from the test set in first threshold experiment. For other threshold experiment we randomly sampled all points above 90 percentile threshold of the selected feature from the test set, We use *pandas.sample* with random seed of 101 in python for sampling.

3. **Pre-train:** Randomly sampled 50% of all points between 50-80 percentile of the selected feature from the train set in first threshold experiment. Randomly sampled 50% of all points between 60-90 percentile of the selected feature from the train set in first threshold experiment. We use *pandas.sample* with random seed of 101 in python for sampling.

Table 4: Data points for Pre-training evaluation(threshold 80 percentile on loan amount for train-test).

	#samples	#features	#default percentage
pre-train	200486	38	0.22
train	164416	38	0.23
val	37734	38	0.32

Table 5: Data points for Pre-training evaluation(threshold 90 percentile on loan amount for train-test).

	#samples	#features	#default percentage
pre-train	186187	38	0.22
train	113413	38	0.21
val	31603	38	0.26

Table 6: Data points for Pre-training evaluation(threshold 80 percentile on annual income for train-test).

	#samples	#features	#default percentage
pre-train	182202	38	0.19
train	158973	38	0.20
val	38392	38	0.26

Table 7: Data points for Pre-training evaluation(threshold 90 percentile on annual income for train-test).

	#samples	#features	#default percentage
pre-train	173342	38	0.18
train	116156	38	0.20
val	29422	38	0.26

Table 8: Data points for Pre-training evaluation(threshold 80 percentile on average current balance for train-test).

	#samples	#features	#default percentage
pre-train	158774	38	0.19
train	148179	38	0.20
val	38017	38	0.26

Table 9: Data points for Pre-training evaluation(threshold 90 percentile on average current balance for train-test).

	#samples	#features	#default percentage
pre-train	158774	38	0.18
train	105845	38	0.20
val	28851	38	0.26

5.2.3 Experiment Results

Here we recorded results when we did not pre-trained and directly trained on the *train* set. We also recorded metrics when we pre-train on our *pretrain* set both in a supervised and unsupervised fashion and then fine-tune on our target *train* set.

For no pre-training, we trained the train set and selected the best model based on AUROC on the validation set. We observe pre-training does improve the performance for our problem compared to no pre-training/ Also supervised pre-training over-performs than self-supervised contrastive pre-training described in SAINT Somepalli et al. (2021) when we had 80 percentile for loan

amount. The improvement over no pre-training is not significant for more extreme threshold. Also pretraining was less effective when used annual income as threshold.

Table 10: Pre-training (supervised and unsupervised) with 80 percentile threshold on loan amount

Model	AUROC	AUPRC
No-pretrain	0.682	0.478
Self-supervised pretrain	0.685	0.482
Supervised pretrain	0.687	0.487

Table 11: Pre-training (supervised and unsupervised) with 90 percentile threshold on loan amount

Model	AUROC	AUPRC
No-pretrain	0.682	0.489
Self-supervised pretrain	0.681	0.486
Supervised pretrain	0.682	0.490

Table 12: Pre-training (supervised and unsupervised) with 80 percentile threshold on annual income

Model	AUROC	AUPRC
No-pretrain	0.716	0.401
Self-supervised pretrain	0.707	0.389
Supervised pretrain	0.713	0.403

Table 13: Pre-training (supervised and unsupervised) with 90 percentile threshold on annual income

Model	AUROC	AUPRC
No-pretrain	0.712	0.397
Self-supervised pretrain	0.714	0.399
Supervised pretrain	0.712	0.400

Table 14: Pre-training (supervised and unsupervised) with 80 percentile average current balance

Model	AUROC	AUPRC
No-pretrain	0.703	0.348
Self-supervised pretrain	0.697	0.344
Supervised pretrain	0.696	0.344

Table 15: Pre-training (supervised and unsupervised) with 90 percentile average current balance

Model	AUROC	AUPRC
No-pretrain	0.702	0.329
Self-supervised pretrain	0.695	0.318
Supervised pretrain	0.699	0.327

6 Conclusion

In this project firstly we compared SAINT which is a deep tabular model for evaluating creditworthiness using banking data compared to other classical models. We also explored how to pre-training and then fine-tune on segments of clients with a lot fewer data. We explored self-supervised learning architecture described in SAINT Somepalli et al. (2021) paper and also evaluated supervised pre-training where we used the labels for pre-training.

From the experiment result, we observe that SAINT architecture marginally over-performs Logistic Regression, Decision trees and XGBoost. We get AUROC/AUPRC 0.7/0.42 using SAINT. We only get AUROC/AUPRC of 0.65/0.37 for Decision trees. XGBoost gives a slightly better 0.68/0.4. With Logistic regression, we get even less favourable results with AUROC/AUPRC of 0.68/0.39.

We also observe that self-supervised pre-training over-performs over no pre-training for Loan data. Also, supervised pre-training is slightly better than self-supervised contrastive pretraining. For Self-supervised contrastive pre-training, we get AUROC/AUPR of 0.685/0.482 whereas standard supervised pre-training gives a slightly better AUROC/AUPR of 0.687/0.487. For 90 percentile cutoff though the gains are not much noticeable. Also pretraining was less effective when we segment client based on annual income and average current balance.

We can conclude that the strategy of using data from other segments for pre-training on a larger set does seem to work for credit default risk prediction on segments of clients where we do not have the luxury of large data when we want to train segment based on loan amount.

References

Addo, P. M., Guegan, D., and Hassani, B. (2018). Credit risk analysis using machine and deep learning models. *Risks*, 6(2):38.

Arik, S. O. and Pfister, T. (2020). Tabnet: Attentive interpretable tabular learning.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805.

Galindo, J. and Tamayo, P. (2000). Credit risk assessment using statistical and machine learning: basic methodology and risk modeling applications. *Computational Economics*, 15(1):107–143.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. (2020). Bootstrap your own latent: A new approach to self-supervised learning.

Hendrycks, D., Mazeika, M., Kadavath, S., and Song, D. (2019). Using self-supervised learning can improve model robustness and uncertainty.

Huang, X., Khetan, A., Cvitkovic, M., and Karnin, Z. (2020). Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678*.

Huang, Z., Chen, H., Hsu, C.-J., Chen, W.-H., and Wu, S. (2004). Credit rating analysis with support vector machines and neural networks: a market comparative study. *Decision support systems*, 37(4):543–558.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2020). Albert: A lite bert for self-supervised learning of language representations.

Popov, S., Morozov, S., and Babenko, A. (2019). Neural oblivious decision ensembles for deep learning on tabular data. *arXiv preprint arXiv:1909.06312*.

Somepalli, G., Goldblum, M., Schwarzschild, A., Bruss, C. B., and Goldstein, T. (2021). Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. *arXiv preprint arXiv:2106.01342*.

Tian, Y., Chen, X., and Ganguli, S. (2021). Understanding self-supervised learning dynamics without contrastive pairs. *arXiv preprint arXiv:2102.06810*.

Ucar, T., Hajiramezanali, E., and Edwards, L. (2021). Subtab: Subsetting features of tabular data for self-supervised representation learning. *Advances in Neural Information Processing Systems*, 34.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need.

Yoon, J., Zhang, Y., Jordon, J., and van der Schaar, M. (2020). Vime: Extending the success of self-and semi-supervised learning to tabular domain. *Advances in Neural Information Processing Systems*, 33.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization.