DIFFERENTIAL PRIVACY FOR CREDIT RISK MODEL

ABSTRACT

The use of machine learning algorithms to model user behavior and drive business decisions has become increasingly commonplace, specifically providing intelligent suggestions/recommendations to automated decision making (like in loan applications etc.). This has led to an increase in the use of customer's personal data (such as age, SSN, and demographics, etc.) to analyze customer behavior and predict their interests in a company’s products and/or services. Increased use of this customer personal data can lead to better models but also to the potential of customer data being leaked, reverse-engineered, and mishandled.

In this paper, we assess differential privacy as a solution to address these privacy problems by building privacy protections into the data engineering and model training stages of predictive model development. Our interest is pragmatic implementation in an operational environment, which necessitates a general-purpose differentially private modeling platform/framework, and we evaluate one such a tool from LeapYear as applied to the Credit Risk modeling domain. Credit Risk Model is a major modeling methodology in banking and finance where user data is analyzed to determine the total Expected Loss to the bank. We examine the application of differential privacy on the credit risk model and evaluate the performance of a Differentially Private Model (DPM) with a Non-Differentially Private Model (NDPM).

Credit Risk Model is a major modeling methodology in banking and finance where user’s data is analyzed to determine the total Expected Loss to the bank. In this paper, we explore the application of differential privacy on the credit risk model and evaluate the performance of a Non-Differentially Private Model (NDPM) with a Differentially Private Model (DPM).

Keywords Differential Privacy, Credit Risk Model, Machine Learning

1 Introduction

In the past few years, the use of machine learning algorithms to model user behavior has been prevalent. With this, the personal data of the user such as Location, Age, SSN have been met with several attacks both from internal as well external attack vectors. A significant internal vulnerability is the risk that the data scientist or others with access to personal data may accidentally or maliciously leak confidential information, and so there is a need to prevent this from happening while enabling legitimate development and workflows on the underlying data. And it is imperative that the trained model does not inadvertently expose information about the users in the training data, a well-known risk [1].

This has led to research and applications in the area of Differential Privacy (DP). In this paper, we assess the application of Differential Privacy (DP) for Credit Risk Modeling, which deals with understanding and predicting risks associated with loan disbursement to one or more parties with multiple areas for risk modeling. Concretely, we perform exposure modeling including the probability of default—i.e. the total amount of (potential) exposure at the time of default, how much the loan is expected to be worth at the time of default, and the overall loss if there is a default.
This paper deals with exposure modeling including the probability of default, i.e., the total amount of (potential) exposure at the time of default, how much the loan is expected to be worth at the time of default, and the overall loss if there is a default.

The purpose of this study was to address two main research questions about using DP to develop credit risk models:

- Can privacy-preserving models perform as well as non-private models in terms of key outcomes such as loss estimation and variance?
- What are the operational considerations of privacy-preserving models, such as computational overhead and implementation of trained models?

2 Brief look at Differential Privacy

[DP] Differential Privacy provides a mathematical basis for privacy protection, with provable limitations on the amount of information that can be extracted about a particular record in a database, such as a customer or transaction, regardless of what other information or computational resources are available to a would-be adversary [2]. The mathematically provable privacy protection of [DP] stands in contrast to approaches such as anonymization and masking, which are highly vulnerable to re-identification [3]. [DP] can provide protection for two vulnerabilities in the context of predictive model development: a model developer who needs to leverage training data, including private information, for exploratory analysis and modeling can do so without direct access to the source data and with mathematical protection against reverse engineering of protected information; and the [DPM] protects information leakage from individual records in the training data. Differential Privacy has been applied to multiple domains such as on census data [4] where they have used Rappor as their differential privacy framework [5]. US Census Bureau adopted differential privacy to protect the End-to-End Census Test [6]. It has been applied to obfuscate the images from dataset [7] and recently the ImageNet has blurred the images in the dataset with minimal impact on the accuracy. [DP] has applications in the space of Health where analysts have to analyze private health records [8]. [DP] has been applied to create Recommender Systems which can handle private datasets to protect the user’s history [9]. Differential Privacy can also apply to Location Datasets such as the path of taxis and users’ location history which is private information to the user and can likely be misused [10].

3 Credit risk models

Credit risk measures the risk of a loss that may occur if a borrower does not make the required payments on the loan or fails to meet his or her debt repayment obligations. Credit risk modeling is a technique used by lenders to estimate the level of risk associated with each borrower. The risk for the lender is of several kinds, ranging from disruption to cash flows and increased collection costs to loss of interest and principal. That’s why it’s important to be able to anticipate credit risk as accurately as possible. Credit risk on a variety of individual characteristics (factors), and the nature of dependency could be quite complex. Lenders employ sophisticated models which excel at capturing this dependency.

There are several major factors to consider while determining credit risk. From the financial health of the borrower and the consequences of a default for both the borrower and the creditor to a variety of macroeconomic considerations. The credit risk of a borrower is typically quantified via three major components, which are modeled separately:

- **Probability of Default (PD)**: This model gives if the borrower will default or not. It is the likelihood that the borrower will not be able to make scheduled repayments of the loan. When PD is higher the lender will charge a higher rate. It is a classification model which would determine if the borrower will default or not within a specified time frame given KYC information, historical repayment data (of similar customers), and present and future economic outlook.

- **Exposure at Default (EAD)**: EAD is the maximum amount that the bank is exposed to, at the moment the borrower defaulted on the loan obligation. It is often modeled focusing on a credit conversion factor, which is defined as a ratio to the original loan amount, so that:

\[
\text{exposure\_at\_default} = \text{total\_funded\_amount} \times \text{credit\_conversion\_factor}
\]

\[
\text{credit\_conversion\_factor} = \frac{\text{total\_funded\_amount} \times \text{total\_recovered\_principal}}{\text{total\_funded\_amount}}
\]
where:

- \textit{total\_funded\_amount} is the initial amount that was lend to the borrower
- \textit{credit\_conversion\_factor} is the proportion of the original funded amount that the borrower defaulted on.

The borrower may have already payed back some part of the amount that has been initially borrowed.
- \textit{total\_recovered\_principal} is the amount that has already been payed back by the borrower before the borrower has defaulted.

\textbf{• Loss Given Default (LGD):} LGD is the amount which financial institute loses when a borrower defaults on a loan after recovery. It is the loss that can not be recovered after the default.

\begin{equation}
\text{loss\_given\_default} = 1 - \text{recovery\_rate}
\end{equation}

where:

- \text{recovery\_rate} is the loss that can be recovered.

Using the output from the models we calculate the Total Expected Loss by using the formula.

\begin{equation}
\text{total\_expected\_loss} = \sum (\text{probability\_of\_default} \times \text{exposure\_at\_default} \times \text{loss\_given\_default})
\end{equation}

where:

- \textit{total\_expected\_loss} is the total amount that the bank is expected to lose on all the borrowers.

\section{Methodology}

\subsection{Data}

To build the Credit Risk Model we used Lending Club dataset which is publicly available for the years 2016 and 2017. The dataset has 39K records and multiple features about the lender such as age, loan amount, the status of the loan, total recovered principal, recoveries, address, loan status \cite{11}. We are using this data to determine loss to the bank over a period of time when a customer defaults. We have split the data into train and test datasets with an 80:20 ratio.

\subsection{DP Software Selection}

A key objective of this study was to evaluate an operationally realistic end-to-end \textit{DP} workflow for developing credit risk models, to address privacy vulnerabilities for both the model development process (i.e., to protect data from the model developer) as well as implementation of the trained \textit{DPM} (i.e., to protect training data from end-users of the model). Furthermore, given that financial institutions utilize a broad range of data platforms and services, from on-premise servers to various cloud providers, it was desirable to evaluate software that was not limited to a particular proprietary computing environment. Other selection criteria included: support for \textit{DP} modeling algorithms relevant to credit risk modeling; computational scalability for rapid iteration; and ease of use, i.e., similarity to common non-\textit{DP} analytics tools.

Though \textit{DP} is a very active area of research, today there are relatively few organizations and open source efforts that offer \textit{DP} software for practical applications. Some provide \textit{DP} algorithms for certain types of calculations (e.g., Google \textit{DP} \cite{12}, IBM DiffPrivLib \cite{13}) or tools focused on specific applications such as statistical releases (e.g., OpenDP \cite{14}), and yet others address different privacy scenarios than financial modeling, such as local differential privacy (e.g., Apple \cite{15}).

The \textit{DP} analytics platform from LeapYear Technologies \cite{16} is intended to support end-to-end private workflows like our use case. All interactions with the source data are differentially private, meaning the analyst performs data engineering, exploratory data analysis, feature generation, and model training interactively but without direct access to the data. Furthermore, privacy exposure is quantified and tracked for all interactions and actively managed to prevent information leakage across multiple uses of the same data. LeapYear thus meets the operational objectives to apply \textit{DP} to the entire workflow. LeapYear uses Apache Spark for open-source scalable computation, with a Python API that is similar to non-\textit{DP} open-source analytics tools (e.g., PySpark). And it operates on Linux servers that are supported by a broad range of cloud and on-premise data environments.
4.3 Pre-processing

The source data was prepared for modeling with a similar process for DPM and NDPM modeling. These steps included:

- Binning of states, home ownership variables, and purpose of the loan.
- Removal of highly correlated and unwanted columns from the dataset such as loan amount, zip code, and member id.
- Treatment of null values by imputing the median of the data attributes; and one-hot encoding of categorical variables to ensure compatibility across algorithms, some of which automatically handle categorical variables while others do not.

For DPM development preprocessing was performed with the LeapYear platform, emulating the scenario where an analyst accomplishes this preprocessing without direct access to the source data (from a database or a data lake). Notably, such transformations are deterministic and exact in LeapYear because DP randomization is applied at the point of computing answers to questions that analyst requests to see, not to the underlying data manipulations.

4.4 Algorithms

- **PD**: The dependent variable for modeling PD would be loan status which would determine if the borrower is good or bad. We have used multiple models to calculate PD such as Logistic Regression and Gradient Boosting Trees. Comparatively Gradient Boosting Trees performed better.
- **EAD**: The dependent variable for modelling EAD would be credit conversion factor(CCF) defined above (see 2). To predict CCF we have used the Random Forest regression Model.

\[
\text{PredictedEAD} = \text{TotalFundedAmount} \times \text{predictedCCF}
\]

- **LGD**: The dependent variable for modeling LGD is the recovery Rate. LGD is simply (1 - Recovery Rate). Usually, LGD is calculated using beta regression, which we have implemented by creating two models. A logistic regression classifier to model whether the recovery rate will be non zero and a linear regression model to predict the recovery rate when the recovery rate is non zero. For LGD we have used gradient boosted trees to model if the recovery rate is non zero and to predict the recovery rate we have used Random Forest Model.

5 Result

5.1 Model Performance

To see whether the DPM would generate comparable results to traditional modeling approaches, we have implemented a process to a) train models on a sub-sample of the data and b) generate model predictions on the same training data sample. We have then executed this process several times on different data samples and compared the actual and model predicted loss across these data samples:

| Run | Total Actual Loss | Total Predicted Loss | Relative Difference |
|-----|------------------|----------------------|---------------------|
| 1   | $8,428,504       | $7,758,930           | 11.510              |
| 2   | $8,282,985       | $7,603,560           | 7.556               |
| 3   | $8,087,437       | $7,417,115           | 14.553              |
| 4   | $8,188,585       | $7,501,638           | 15.294              |
| 5   | $8,282,985       | $7,603,560           | 17.556              |
| 6   | $8,287,423       | $7,595,251           | 23.102              |
| 7   | $8,379,437       | $7,501,638           | 25.426              |
| 8   | $8,188,585       | $7,501,638           | 28.462              |

Table 1: Individual Run Results
As we can see in Table 2 above, switching from a traditional modeling approach to a differentially private one - led to less than 5% change in the aggregate expected loss prediction. This effect is well in line with the effect of choosing a different sub-sample for training models, as shown in Table 1 above and illustrated in Figure 3 below.

Table 2: Average Results

| Model Version                                                                 | Avg Actual Loss | Avg Predicted Loss | Avg Relative Difference from Actual |
|-------------------------------------------------------------------------------|-----------------|--------------------|-------------------------------------|
| Model trained in PySpark with full access to the data                         | $8,319,741      | $7,109,281         | 17.21                               |
| Model developed via LeapYear with only differentially private compute access | $8,317,839      | $6,995,703         | 19.37                               |

As we can see in Table 2 above, switching from a traditional modeling approach to a differentially private one - led to less than 5% change in the aggregate expected loss prediction. This effect is well in line with the effect of choosing a different sub-sample for training models, as shown in Table 1 above and illustrated in Figure 3 below.

Figure 1: Actual Loss for DPM and NDPM

Figure 2: Predicted Loss for DPM and NDPM

Figure 2 explains the different runs on the dataset with different sample of the dataset as training and testing. It compares the performance of each run of the DPM and NDPM model on the dataset.
5.2 Operational Considerations

The processing time required to train the model was somewhat slower for the DPM, 5.4 minutes, compared to the NDPM model, 3.1 minutes. Prediction time was essentially the same between the two models. Pre-processing time, such as feature generation, was quite different between the private and non-private methods in this experiment, taking 13.5 minutes for the DPM vs. 0.3 minutes for the NDPM. LeapYear advised that feature calculations can be optimized to reduce or eliminate this difference, but this was not explored further in this study, and pre-processing time is generally less important than training and prediction time for operational implementation. It is possible to productionalize DPM developed with LeapYear by first mapping the trained model to one of the common formats such as sklearn, pmml, and pfa. This process was demonstrated for linear regression and logistic regression in this study. More complex tree-based models can be similarly converted from LeapYear’s standard format to other formats in principle, but this was not demonstrated during this study. Training can be done in a LeapYear environment but the deployment of LeapYear models can be done by persisting the model to the storage disk and then loading the model in a non-LeapYear environment.

Another desirable advantage of the DPM is to keep the data hidden from the analyst during the model development process, making the sensitive data of the individual customers more secure. Note that this requires access controls in addition to DP, as with the LeapYear platform.

In addition to generating model predictions for individual records, these LeapYear model predictions for individuals could be explained and rationalized using similar tools as traditional open-source models - e.g. shap, see link: https://shap.readthedocs.io/en/latest/index.html. This workflow can be enabled by mapping models trained with LeapYear - to more traditionally supported formats such as scikit-learn.

6 Conclusion

In the context of the research question “Can privacy-preserving models perform as well as non-private models?”, it was observed that DPM perform comparably to their NDPM counterparts while providing added peace of mind for customers and banks on data privacy protection. Operationally, computational overhead for DPM was negligible for model predictions, of comparable magnitude for model training, and noticeably larger for data pre-processing, though the operational impact can be mitigated or eliminated via automation and optimization of pre-processing. Production deployment of trained DPM can be accomplished via conversion to standard model formats. Most importantly, DPM can be developed without direct access to source data via a combination of DP and access controls, as with the
LeapYear platform. Additionally, the ability to add noise in a controlled manner specific to a given use case will allow banks to avoid data duplications in their data lake.

References

[1] Charles Elkan Zhanglong Ji, Zachary C. Lipton. Differential privacy and machine learning: a survey and review., 2014.
[2] Aaron Roth Cynthia Dwork. The algorithmic foundations of differential privacy. foundations and trends in theoretical computer science, 9(3-4), 211-407., 2014.
[3] Vivek Kumar Singh1 Yves-Alexandre de Montjoye, Laura Radaelli and Alex Pentland. Unique in the shopping mall: On the reidentifiability of credit card metadata. science, 347(6221), 536-539., 2015.
[4] Keyu Zhub Ferdinando Fioretto, Pascal Van Hentenryck. Differential privacy of hierarchical census data: An optimization approach., 2021.
[5] Aleksandra Korolova Úlfar Erlingsson, Vasyl Pihur. Rappor: Randomized aggregatable privacy-preserving ordinal response., 2014.
[6] John M. Abowd. The u.s. census bureau adopts differential privacy., 2018.
[7] Jörg-Rüdiger Sack William L. Croft and Wei Shi. Obfuscation of images via differential privacy: From facial images to general images., 2021.
[8] Khaled E El Emam Fida Kamal Dankar. The application of differential privacy to health data., 2015.
[9] Frank McSherry and Ilya Mironov. Differentially private recommender systems: Building privacy into the netflix prize contenders., 2009.
[10] Wanlei Zhou Tianqing Zhu, Gang Li and Philip S. Yu. Differential privacy and applications., 2017.
[11] Tabitha Kemboi and Mohammad R. Islam. Project: Lending club data analysis, 2019.
[12] Google. Differential Privacy. This repository contains libraries to generate differentially private statistics over datasets. https://github.com/google/differential-privacy
[13] IBM Corporation. Diffprivlib, A general-purpose library for experimenting with, investigating and developing applications in, differential privacy., 2018. https://github.com/IBM/differential-privacy-library
[14] openDP. https://github.com/opendp
[15] Apple. Differential Privacy Overview. https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf
[16] LeapYear.io. Mathematically proven privacy for the enterprise. https://leapyear.io/