CS229

Encryption and Machine Learning: How Classifications May Be Impacted by Encryption

Yosheb Getachew, Meg Richey, Joshua Shongwe
Department of Computer Science
Stanford University
yoshebg@stanford.edu, jshongwe@stanford.edu, richey@stanford.edu

Abstract

Given the abundant applications of machine learning methods, data acquisitioning has become a paramount task for corporations and organizations that utilize ML to inform decision making protocols and aid in problem-solution ventures. Due to the integration of ML into numerous fields that require discretion in data collection procedures by organizations such as hospitals, technological goods distributors, and financiers, an imperative to interpolate security protocols into machine learning techniques has emerged to ensure confidentiality is preserved for individuals contributing to these datasets. Prevailing literature evidences measures taken to safeguard data in machine learning, via privacy-preserving classification methods. In this project we endeavor to create our own privacy-preserving classification model using machine learning frameworks from CS 229 such as SVM and logistic regression, and compare their performance to classical models trained on un-encrypted data.

1 Introduction

Computer science offers an innovative and efficient medium to expeditiously solving high density analysis problems that require high complexity computations or large breadth data searches. This resolves many contemporary problems, one of which is classification. Classifiers have provided us with efficient solvable methods for problems such as medical or genomics predictions, spam detection, facial recognition, and financial prediction and analysis. The methodology of implementation of a general classifier in a supervised learning problem is relatively straightforward with respect to most prototypical machine learning problems, and can be summarized in two phases: the training phase during which the algorithm learns a model $\mathcal{W}$ from a data set of labeled examples, and the classification phase that runs a classifier $\mathcal{C}$ over a previously unseen feature vector $\mathbf{X}$, using the model $\mathcal{W}$ to output a prediction $\mathcal{C}(\mathcal{W}, \mathbf{X})$. One weakness of classification methods is its susceptibility to interactions between training data and test data, especially as multiple features combined can allow for anonymized data to be re-associated with the individual. We attempted to combat said susceptibility through the following process:
1. Encrypt data according to encryption key.
2. Train a classifier on it.
3. Try to classify under a different encryption scheme.
4. Determine can the classifier preserve anonymity and see through the encryption to identify the features.

While larger companies might always have the capacity and background knowledge to both encrypt images before classification and knowing that they should encrypt sensitive images, having simple "off-the-shelf" methods available to users is highly valuable. Another pertinent use is to easily implement classification methods for start-up software services that might run the risk of potentially harmful images being shared. For example, in many startups and even Google/Facebook crawling algorithms that involve users uploading and sharing images and content or massive data scrapes, it is someone's or something's job to manually scan for potentially inappropriate photos. This encryption combined with the classification would allow for models to scan for inappropriate photos while not revealing the potentially sensitive content unnecessarily.

2 Related Works

Weiru Wang, Chi-Man Vong, Yilong Yang, and Pak-Kin Wong noticed early, the problematic nature of utilizing massive data sets of real-people for Machine Learning training sets. Expressing "numerous corperations (such as Google, Baidu, etc.) [utilize] search algorithms to crawl out images with queried objects"[6]. Given the timeframe, Weiru and team were making novel strides in the realm of privacy-preserving protocols in the heap of Machine Learning. The goal is emulates our approach: prioritizing effectiveness and efficiency whilst maintaining preservation of privacy on all images. Weiru and team inspired the process inspired the process of treating all input images as "classified" and training a Machine Learning Algorithm to classify encrypted images without decryption. Anthony Barnett highlights the strength in combining a Support Vector Machine and a "Somewhat Homomorphic Encryption"[1]; and thus, we utilized such an approach, in effort to provide a novel idea with commercial value.

One of the main issues we found with what was currently available is the issue of scalability, which we even encountered issues with in our solution. In order to encrypt data at this magnitude, there needs to be a lot more information stored, and many solutions fall short in regards to both storage and issues decrypting the data to make it useful again. As mentioned in the partial image article[7] pre-processing can be done on images to isolate specific parts of the image to encrypt, however this assumes that there is only a small fraction of the image that is necessary to encrypt and still requires processing before. Given our limited resources, it was deemed overly ambitious as RAM was repeatedly exhausted in attempt to emulate–a full image encryption. We see applications of privacy preserving protocols utilized in sensitive data fields, e.g. the medical field[4]. As depicted above, all listed approaches interpret input data as "classified", encrpyt said data, utilize classification strategies, and train an algorithm to determine the classified input without declassification.

A more complex application of the related works, our project can be seen in Raphael Bost's "Machine Learning Classification over Encrypted Data" implementing hyperplane decisions, Naive Bayes, and Face Detection Classifiers coupled with a plethora of imported libraries to develop, securely multiplex and face detection classifiers.[2]

3 Dataset and Features

We procured a Loan Approval dataset from Kaggle [7], as it encoded features characterized as confidential such as credit report information, demographical content, and financial statements. This dataset was comprised of 614 examples, partitioned according to the conventional 80-20 split of training-test data in contemporary machine-learning literature(491 training datapoints, 123 test datapoints). Each datapoint $r \in \mathbb{R}^{11}$, with the following feature labels: Gender, Marriage Status, Dependents Status, Education, Employment Status, Applicant Income, Co-Applicant Income, Loan Quantity, Loan Term, Credit History, and Property Area. We selected this dataset as it satisfied our criteria for presenting a comprehensive financial profile of each example datapoint, as well as encoding the principal queries of a standardized loan application one can obtain at an appropriate financial institution.

After visualization of the labeled data to acquire a graphical depiction of the example data, we preprocessed the data for modeling by first splitting the data into example matrices and label vectors. We then stratified the data into training example matrices, training label vectors, and correspondingly test example matrices and test label vectors. We next performed data augmentation, by imputing any missing values so as to avoid any divergence in future optimization due to holes in the input. Finally, we concluded data preprocessing with further augmentation by converting all categorical variables with supports subsets of $\mathbb{R}$ to indicator variables (e.g. Number of Dependents, whose support was $\mathbb{N}$, was decomposed into three binary features labeled "$Dependents = 1$", "$Dependents = 2$", and "$Dependents = 3+$"). This transformed all datapoints from elements of $\mathbb{R}^{11}$ to elements of $\mathbb{R}^{14}$ (we note that we did not convert numerical variables with supports $\mathbb{R}$, such as applicant and co-applicant income).

Subsequent to the modeling of our logistic regression classifer and SVM model, we replicated the Loan Approval dataset and transformed the feature values according to Fernet AES symmetric encryption [8].

Definition 1. ***Symmetric Key Encryption*** *is a cryptographic algorithm that uses the same secret key for its operation and, if applicable, for reversing the effects of the operation. It is also known as a secret-key algorithm.*

To implement this encryption scheme, we harnessed the cryptography module Cryptography.io to generate a training key $\mathbf{key_{train}}$ and a test key $\mathbf{key_{test}}$ to appoint a distinct encryption protocol to respective example datapoints from the training and test partitions. If we denote $\mathbf{key_{train}} = \mathbf{k_1}$ and $\mathbf{key_{test}} = \mathbf{k_2}$, we can consider these encryption keys as maps $\mathbf{k_i} : \mathcal{S}^n \to \mathcal{E}_i^m$, where $\mathcal{S}^n$ is the space of all alphanumeric-symbolic strings of length n, and $\mathcal{E}_i^m$ is the encryption space of the corresponding key $\mathbf{k_i}$.

4 Methods

To examine the effects of encryption on logistic regression and SVM modeling, we devised a novel ansatz encryption algorithm, as a primitive proof-of-concept of the capability of a machine learning model to accurately perform on encrypted data after training on distinctly encrypted examples. We initialized our experimental methodology by first obtaining a control metric set of accuracies and F1-scores for logistic regression and SVM on the unperturbed data. Recall the probabilistic assumption of logistic regression: given the hypothesis $h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$, the parameters of the model are fit via maximum likelihood according to the conditional probability statement $p(y|x;\theta) = (h_\theta(x))^y(1-h_\theta(x))^{1-y}$. Maximizing the log-likelihood yields the stochastic gradient ascent rule $\theta_j := \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}$.

The architecture of this protocol is given here:

```
forall all training examples X[i][j] do
    if Numerical Variable then
        compute length=n of feature value;
        encrypted feature value=k(1/2)(feature value) ;
        select first n numerical values from encrypted feature value;
        replace as effective encrypted feature value;
        if Insufficient Digits in encrypted feature value then
            pad received encrypted feature value with insignificant digits until length n
        end
    end
    if Categorical Indicator Variable then
        encrypt feature value = k(1/2)(feature value);
        select first numerical value in encrypted feature value;
        if Feature Value < Baseline then
            Encrypted Indicator Value == 0
        end
        else
            Encrypted Indicator Value == 1
        end
    end
end
```

Algorithm 1: Naive Encryption Algorithm

We applied two heuristics to establish this encryption algorithm, and via iterative evaluation determined the second to be optimal (see Methods). The first heuristic was to extract the first digit in the encrypted feature value byte string (existence of a numeric character in the encrypted feature value is guaranteed to to the byte string attribution protocol of Fernet AES symmetric encryption [8]). We observed that this heuristic led to poor performance and virtually no learning by the classifier, since the lack of granularity in our assignment of the effective encrypted feature value was a choice that led to frequent overlap in effective encrypted datapoints. This can be visualized as a nontrivial number of dimensions in the original feature space collapsing upon themselves under a singular map given by the encryption.
From this we employed our second heuristic for our encryption baseline. Here we first determined the significant figures of each unencrypted example's feature value. From this we encrypted each training/test example according to its respective encryption key, then mapped this encrypted bytestring a subspace of numeric strings by removing all alphabetic or symbolic characters. At this stage, we

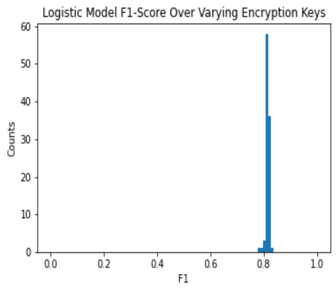
We split our experiments up by data type, focusing on numbers and then on images. For the numbers part, the method was as follows.

For the images section, we wanted to see if a facial recognition model could accurately tell the difference between face images and non-face images after using a rotational encryption algorithm. We began with batching small portions of our data sets without pipe-lining the entirety of the massive data sets. From this, we were able to successfully encrypt and decrypt images. Given the pre-processing was functional, we used a rotational encryption algorithm in an attempt to preserve some sort of mathematical relation between the location of the pixels, while still being unrecognizable to the naked eye for the massive data sets. During this schema, we ran into RAM issues that heavily delayed further progress, so we reverted back to our small-batches process. In our small-batch analysis, we learned that the feature extractor being used "HoG" was heavily reliant on images being batched to a preset feature acquisition that was hidden from the user. We were able to tinker with the system to find a usable batch size, which prompted us to resize our encrypted face/non-faces. From this, we saw the HoG Machine Learning Algorithm was highly capable of detecting non-faces from faces; however, in attempt to locate the "frame" of the face after training on encrypted images this boundary was highly inaccurate in juxtaposition to the default training. On a large scale, if this did work, it would show that there was a way to easily encrypt images for the use of classification, which is important given the rise of image classification in every day use.

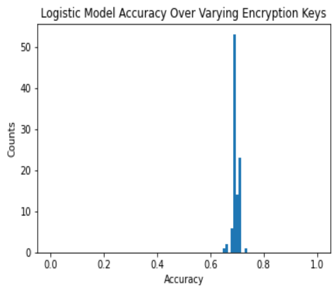
5 Experiments/Results/Discussion

One of the biggest setbacks that we ran into with the photo processing was a limitation on RAM within colab. From the amount that we were able to run, it appears that what we were running was"working" to the extent of the small dataset that we were able to pass in, but when passing in a dataset of actually significant size, we ran into issues. Why we are predicting that this would have worked on the full dataset, however, is based on the mathematics of the encryption being able to retain the relation between pixels to train the HOG and therefore being able to tell which images had faces and which did not.

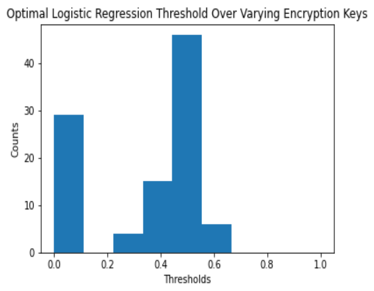
5.1 Figure 1. F1 Score



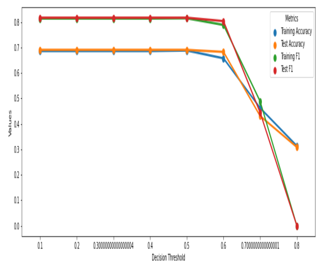
5.2 Figure 2. Model Accuracy



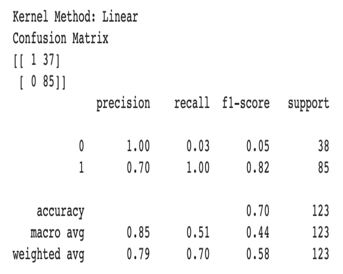
5.3 Figure 3. Logistic Regression Threshold



5.4 Figure 4. Encrypted Decision vs Accuracy F1

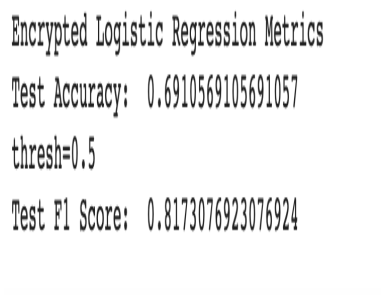


5.5 Figure 5. Kernel Threshold



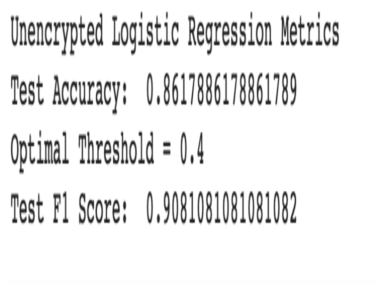
```
Kernel Method: Linear
Confusion Matrix
[[ 1 37]
 [ 0 85]]
              precision    recall  f1-score   support

           0       1.00      0.03      0.05        38
           1       0.70      1.00      0.82        85

    accuracy                           0.70       123
   macro avg       0.85      0.51      0.44       123
weighted avg       0.79      0.70      0.58       123
```

5.6 Figure 6. Logistic Regressin Methods



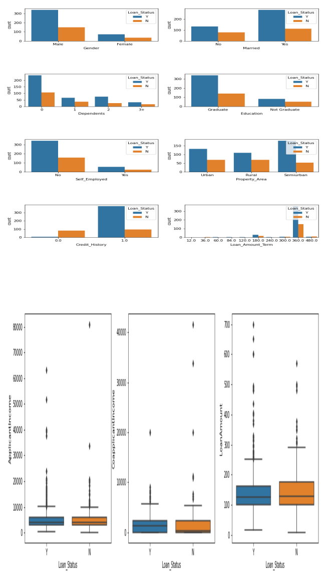
```
Encrypted Logistic Regression Metrics
Test Accuracy:  0.6910569105691057
thresh=0.5
Test F1 Score:  0.8173076923076924
```

5.7 Figure 7. Kernel Threshold



```
Unencrypted Logistic Regression Metrics
Test Accuracy:  0.8617886178861789
Optimal Threshold = 0.4
Test F1 Score:  0.9081081081081082
```

5.8 Figure 8. Categorical Vis



6 Conclusion/Future Work

One of our biggest issues with this project was just the capacity of colab for our research inputs. Future work would definitely be to focus on larger datasets of images and testing other types of encryption to see if the relationships remained more intact to increase accuracy. Another area that we would have liked to progress to but would be an area for future work is to work with audio files as well. Voice encryption is an area that is becoming increasingly more valuable as many applications and devices enable voice commands to control the devices and applications, and must be able to recognize and work with a variety of different languages, dialects, slang words, accents, voice tones, etc. There is a lot more potential for variety in audio than there is in a face, for example, so there is a greater need for more training data, and therefore access to people's personal data and information. Additionally, the content of audio files can be much more revealing than just an image. Therefore, this is an area that would be incredibly helpful and useful for this process to be applied.

7 Contributions

Most of the project was split evenly between research and code between the three of us. Yosh did a lot of the focus on the non-image model, and Meg and Josh focused on the image model. We all contributed to all of the writeups, and overall, it was an amazing collaboration.

References

[1] Barnett, Anthony, et al. Image Classification Using Non-Linear Support Vector ... eprint.iacr.org/2017/857.pdf.
[2] Bost, Raphael, et al. "Machine Learning Classification over Encrypted Data." Proceedings 2015 Network and Distributed System Security Symposium, 2015, doi:10.14722/ndss.2015.23241.

[3] "F-Score." Wikipedia, Wikimedia Foundation, 17 May 2021, en.wikipedia.org/wiki/F-score.

[4] Kaissis, Georgios, et al. ‘‘End-to-End Privacy Preserving Deep Learning on Multi-Institutional Medical Imaging.’’ Nature News, Nature Publishing Group, 24 May 2021, www.nature.com/articles/s42256-021-00337-8.

[5] Mohassel, P, and Y Zhang. ‘‘SecureML: A System for Scalable Privacy-Preserving Machine Learning.’’ IEEE Xplore, 2017 IEEE Symposium on Security and Privacy (SP), 2017, ieeexplore.ieee.org/abstract/document/7958569/figures#figures.

[6] Wang, Weiru, et al. ‘‘Encrypted Image Classification Based on Multilayer Extreme Learning Machine.’’ Multidimensional Systems and Signal Processing, Springer US, 13 Apr. 2016, link.springer.com/article/10.1007/s11045-016-0408-1.

[7] Wonyoung Jang, Sun-Young Lee. ‘‘Partial Image Encryption Using Format-Preserving Encryption in Image Processing Systems for Internet of Things Environment - Wonyoung Jang, Sun-Young Lee, 2020.’’ SAGE Journals, 24 Mar. 2020, journals.sagepub.com/doi/full/10.1177/1550147720914779.

Lit. Review [1]: In Barnett’s "Image Classification using non-linear Support Vector" where he touches on the merge between Support Vector Machines (SVMs) and a Somewhat Homomorphic Encryption (SHE) scheme to provide today’s privacy passionate era with a commercially valuable system. Barnett emphasizes the importance of the use case, expressing data breaches are a cost. "Companies will often have their own security and privacy concerns about handing over their captured images to the algorithm provider. At present, this problem is resolved contractually. One or other of the image data provider and algorithm provider agree to hand over their sensitive information, and trust the other party not to abuse it."[1]. However, given this encryption corresponds to the utilization of the algorithm we have developed a commercially valuable, privacy-protected manner to further traverse into the world of Machine Learning and Artificial Intelligence.

Libraries

[1] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12 (2011): 2825-2830.

[2] Chollet, F., others. (2015). Keras. GitHub. Retrieved from https://github.com/fchollet/keras

[3] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 0.1038/s41586-020-2649-2. (Publisher link).

[4] Hunter, John D. "Matplotlib: A 2D graphics environment." Computing in science engineering 9.3 (2007): 90-95.

[5] Virtanen, Pauli, et al. "SciPy 1.0: fundamental algorithms for scientific computing in Python." Nature methods 17.3 (2020): 261-272.

[6] McKinney, Wes. "pandas: a foundational Python library for data analysis and statistics." Python for High Performance and Scientific Computing 14.9 (2011).

[7] Abadi, Martín, et al. "Tensorflow: A system for large-scale machine learning." 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16). 2016.

[8] Umesh P. Image Processing in Python. CSI Communications. 2012;23.