GRACE HOPPER CELEBRATION

ANITA B.ORG

Applying Natural Language Processing to Software Engineering: the Case of Classifying Security Bug Reports

Mayana Pereira

Data Scientist

Microsoft

A Complex Engineering Environment

~47,000 Engineers all potentially contributing security bugs

Over 2,000 products in development

100+ different Bug Tracking Systems

Unique labels and queries for security bugs

Dependency on Engineers correctly labeling security bugs

Data Source

Data from engineering and compliance systems across Microsoft

Identifying security-related issues among reported bugs is a pressing need among software development teams.

Security issues call for **more expedited fixes** to meet compliance requirements and ensure the **integrity of the software and customer data**.

What is a Security Bug?

- Software code flaw
- Software Design issue
- Operational flaw of implemented software in production

Impact of machine learning-based solution for security bug identification

More complete identification of security bugs.

More accurate identification of security bugs

Expedite security bug solving.

Main Challenges

- Dataset consists of Bug titles only
  - Not all Bugs have descriptions in our database
  - Sensitive data such as credentials are commonly found in bug descriptions
- Manual identification of security issues error-prone. It is estimated that around 30% of bugs are mislabeled.
  - Development team's lack of expertise in security
  - fuzziness of certain problems
- Security space has a very high acceptance criteria
  - Precision & Recall > 90%

Data

- The data was collected from various teams across Microsoft in the years 2015, 2016, 2017 and 2018.
- All reports were closed and verified at some point by a Security Engineer.
- The training dataset consists of Bug titles and bug labels.

XSS	Security
Buffer Overflow	Security
Button in wrong place	Non-Security
Wrong font in main page	Non-Security
Button does not work	Non-Security

Objectives

Goal 1

Obtain a Classifier that is “as close as possible” to a Security Expert for the task of classifying a bug report as security/non-security by using only bug titles as input data.

Goal 2

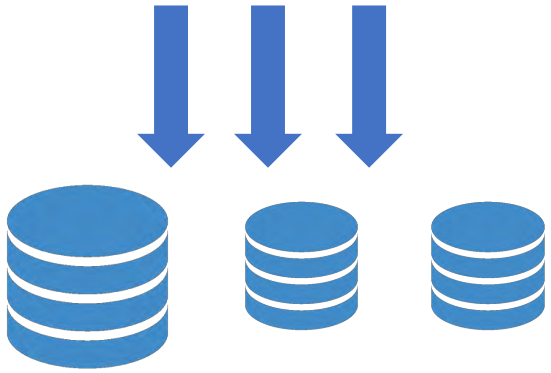
Understand whether noisy data sets can be used to train security bugs identification systems. How different classifiers behave in the presence of noise for identification of Security Bugs?

Goal 1

Machine learning classifiers for security bug Identification

Is it possible to train a ML model for security bug identification?

Bug Reports Database



Training data

Validation data

Test data

Training data

Validation data

Trained Model

More Challenges

Unique vocabulary

specific terms that relates to software vulnerabilities and security flaws -- > needed our own corpus

Interpretability is important!

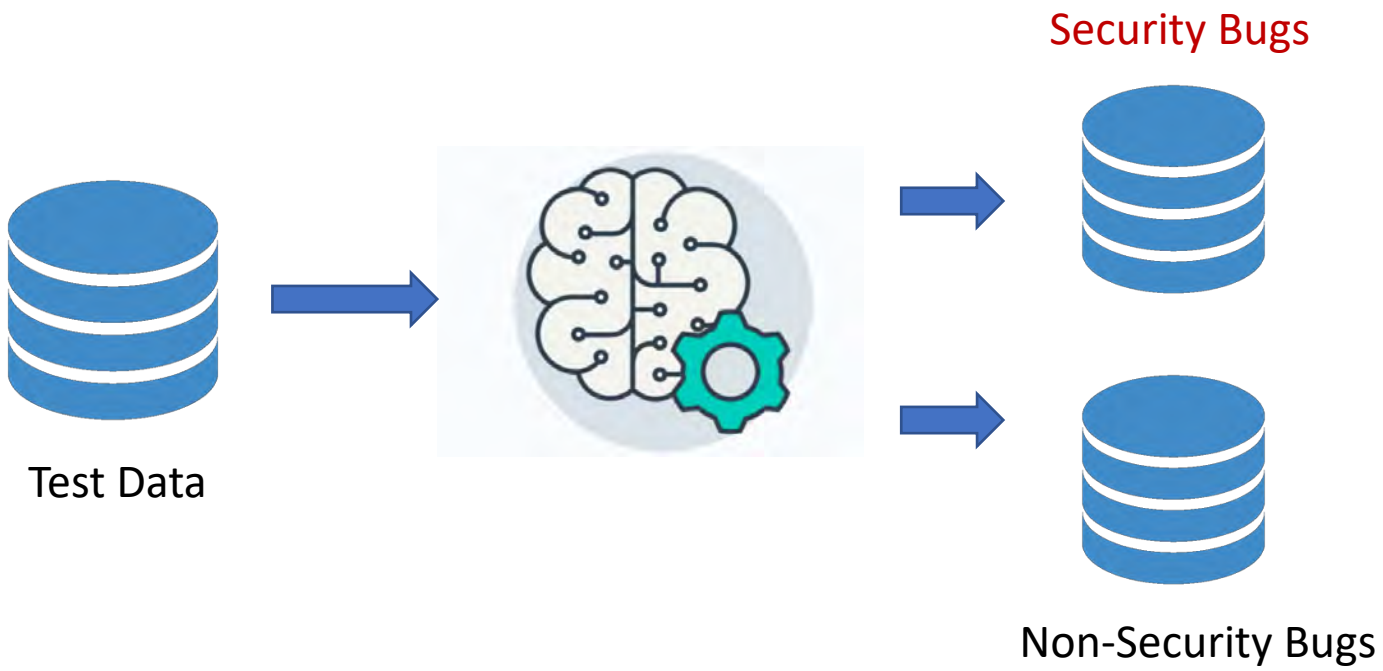
How do we do it?

- **Represent titles as fixed sized vectors using text frequency – inverse document frequency (TF-IDF) technique.**
  - **For a given term *t* document *d*, TF-IDF will attribute a weight to *t* by measuring the frequency of the term in *d* and the number of documents that contains *t* in the entire data set.**

Interpretable machine learning techniques and more complex machine learning techniques

- **Logistic Regression**
- **Naïve Bayes**
- **Boosted Decision Trees**

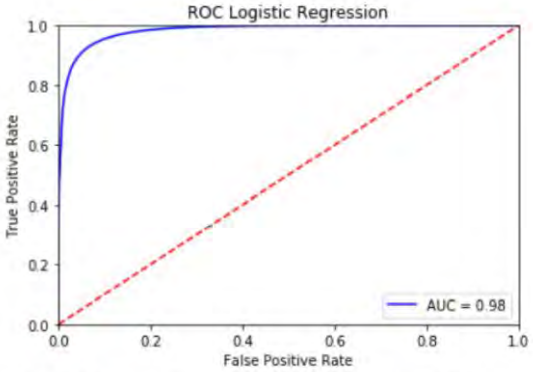
Test models trained using different techniques



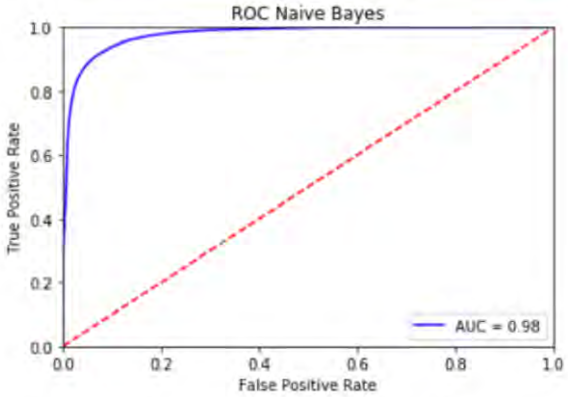
Interpretable models perform well

PERFORMANCE OF DIFFERENT ML TECHNIQUES IN SECURITY BUG CLASSIFICATION

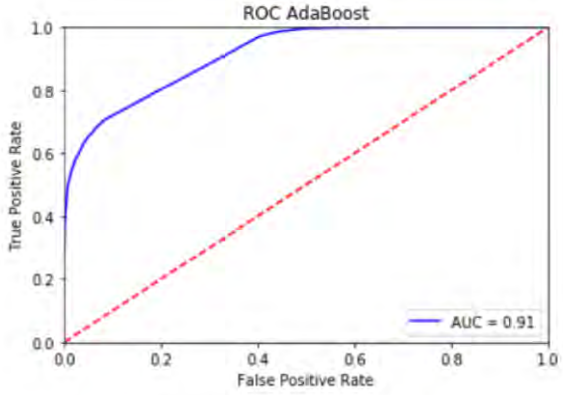
ML Model	Metric			
	Acc	*TPR*	*FPR*	*AUC*
Logistic Regression	0.9309	0.9353	0.0735	0.9831
Naive Bayes	0.9210	0.9189	0.0769	0.9770
AdaBoost	0.8122	0.7018	0.0774	0.9143



(a) ROC curve for Logistic Regression Model



(b) ROC curve for Naive Bayes Model

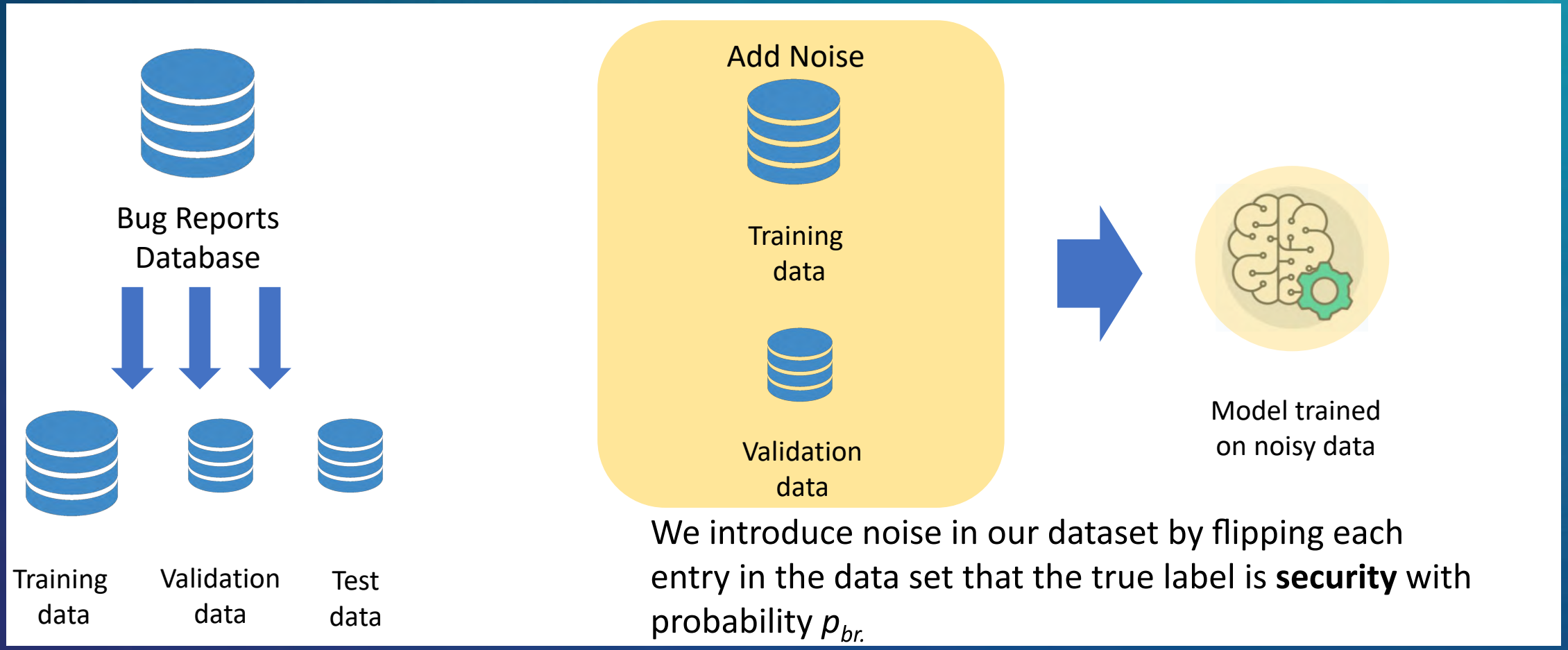


(c) ROC curve for AdaBoost Model

Goal 2

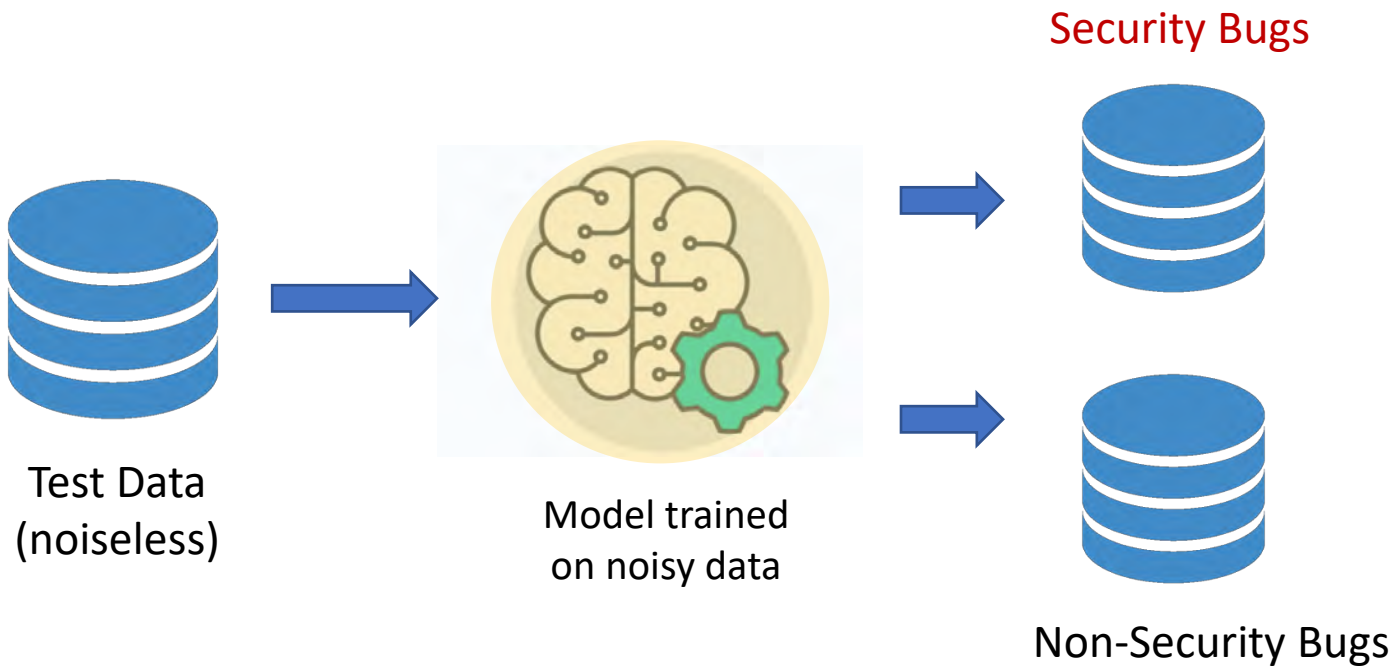
Understand whether noisy data sets can be used to train security bugs identification systems

Can we train our model using noisy data?



We introduce noise in our dataset by flipping each entry in the data set that the true label is **security** with probability $p_{br.}$

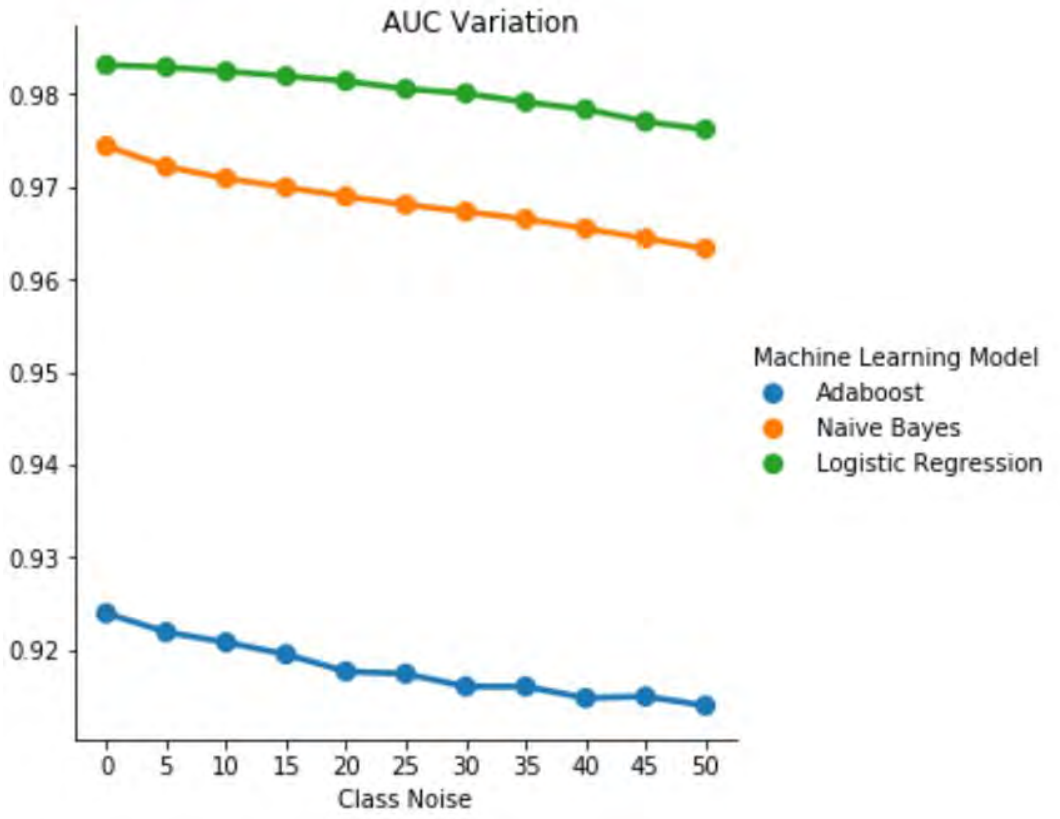
Can we train our model using noisy data?



Training data set does need to be perfect

Single-class noise

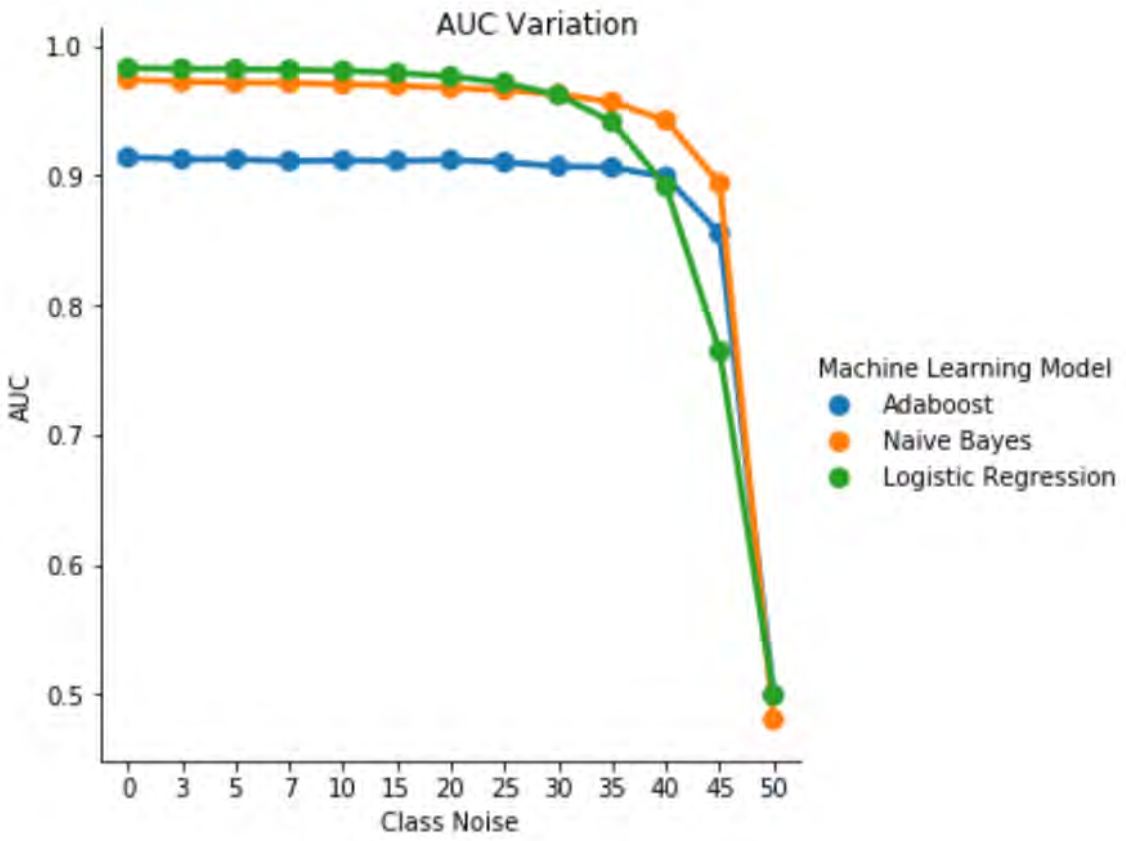
p_{sbr} value	Machine Learning Model		
	logistic regression	***naive Bayes***	***AdaBoost***
0.0	0.983	0.974	0.923
0.05	0.982	0.972	0.921
0.10	0.982	0.970	0.920
0.15	0.981	0.969	0.919
0.20	0.981	0.968	0.917
0.25	0.980	0.968	0.917
0.30	0.980	0.967	0.916
0.35	0.979	0.966	0.916
0.40	0.978	0.965	0.914
0.45	0.977	0.964	0.914
0.50	0.976	0.963	0.913



Imperfect Data Can Work!

Class-independent noise

p_{sbr} value	Machine Learning Model		
	logistic regression	*naive Bayes*	*AdaBoost*
0.0	0.9827	0.9739	0.9140
0.05	0.9820	0.9716	0.9125
0.10	0.9808	0.9703	0.9116
0.15	0.9792	0.9692	0.9113
0.20	0.9763	0.9676	0.9120
0.25	0.9714	0.9658	0.9102
0.30	0.9621	0.9626	0.9071
0.35	0.9412	0.9566	0.9062
0.40	0.8917	0.9425	0.8989
0.45	0.7645	0.8939	0.8553
0.50	0.4994	0.4806	0.4996



Takeaways

Bug Titles contain a lot of information!

- We have shown the feasibility of security bug report classification based solely on the title of the bug report. This is particularly relevant in scenarios where the entire bug report is not available due to privacy constraints.

- Our classification model that utilizes a combination of TF-IDF and logistic regression performs at an AUC of 0.9831.

- **Imperfect Data can work!**
  - All three classifiers are robust to single-class noise.
  - The decrease in AUC is very small (0.01) for a level of noise of 50% (single-class noise).

- Finally, class-dependent noise significantly impacts the AUC only when there is more then 35% noise in both classes.
- The first systematic study on the effect of noisy data sets for security bug report identification.

Please remember to complete the session survey in the mobile app.

THANK YOU!

GRACE HOPPER CELEBRATION

ANITA B.ORG

#GHC19