Capstone Project:

ML-Enabled Phishing Email Detection

Presenter:

Rayile Adam
V01073183

Tahir Jamil
V01071592

Ricky Lin
V00044860

Quoc Dung Van
V01073195

Zhang Zhang
V01046193

Chris McLaughlin
V00912353

University of Victoria

ECE592B and ECE597 © Summer 2025

Team Contribution

Tahir Jamil / Chris Mclauglin

Data exploration

Support Vector Machine

Ricky Lin / Zhang Zhang

Data Pre-processing

Bayes with TF-IDF

Logistic Regression

Rayile / Quoc Dung:

Bayes with BoW

Random Forest

Sentence Embedding

Background - Phishing Email

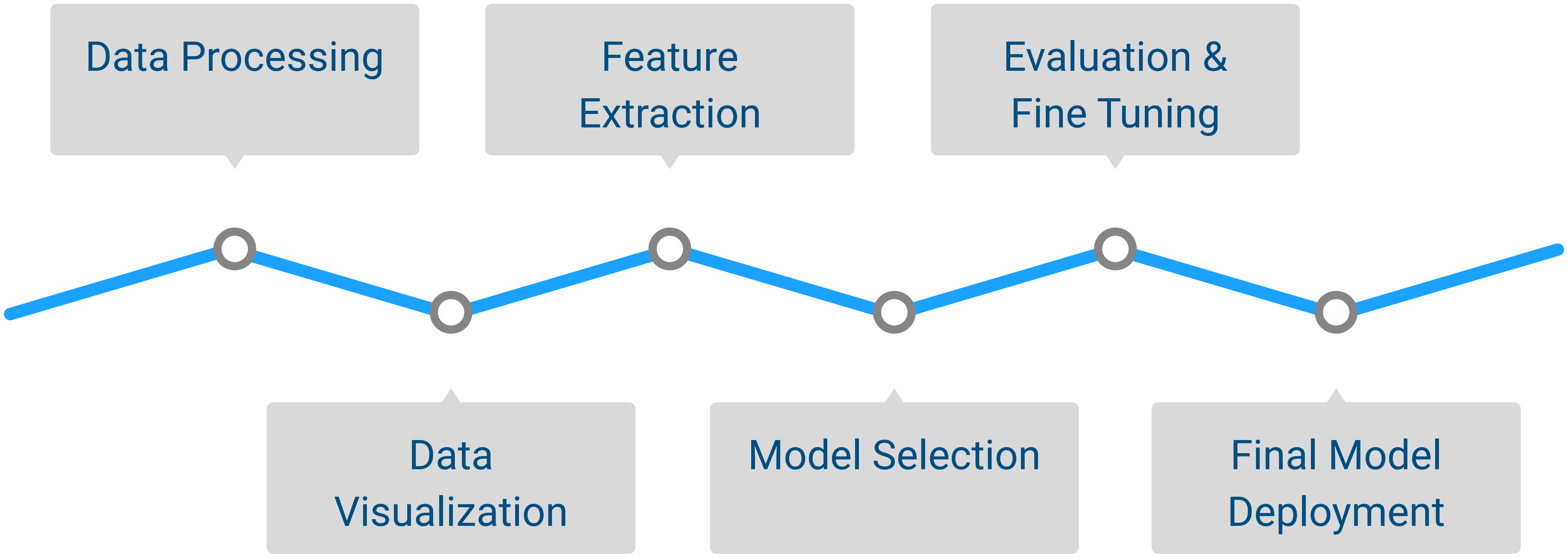
- Phishing Threat: email-based cyber tricking users
- Traditional Methods: Blacklists, rule-based filters
- Machine Learning Potential: adapts to evolving phishing tactics
  - Naive Bayes, Logistic Regression, SVM, NLP

Project Goal:

- Develop an ML-powered system to classify emails as phishing or legitimate.
- Compare performance across several models: Naive Bayes, Logistic Regression, SVM, and Random Forest.

University of Victoria

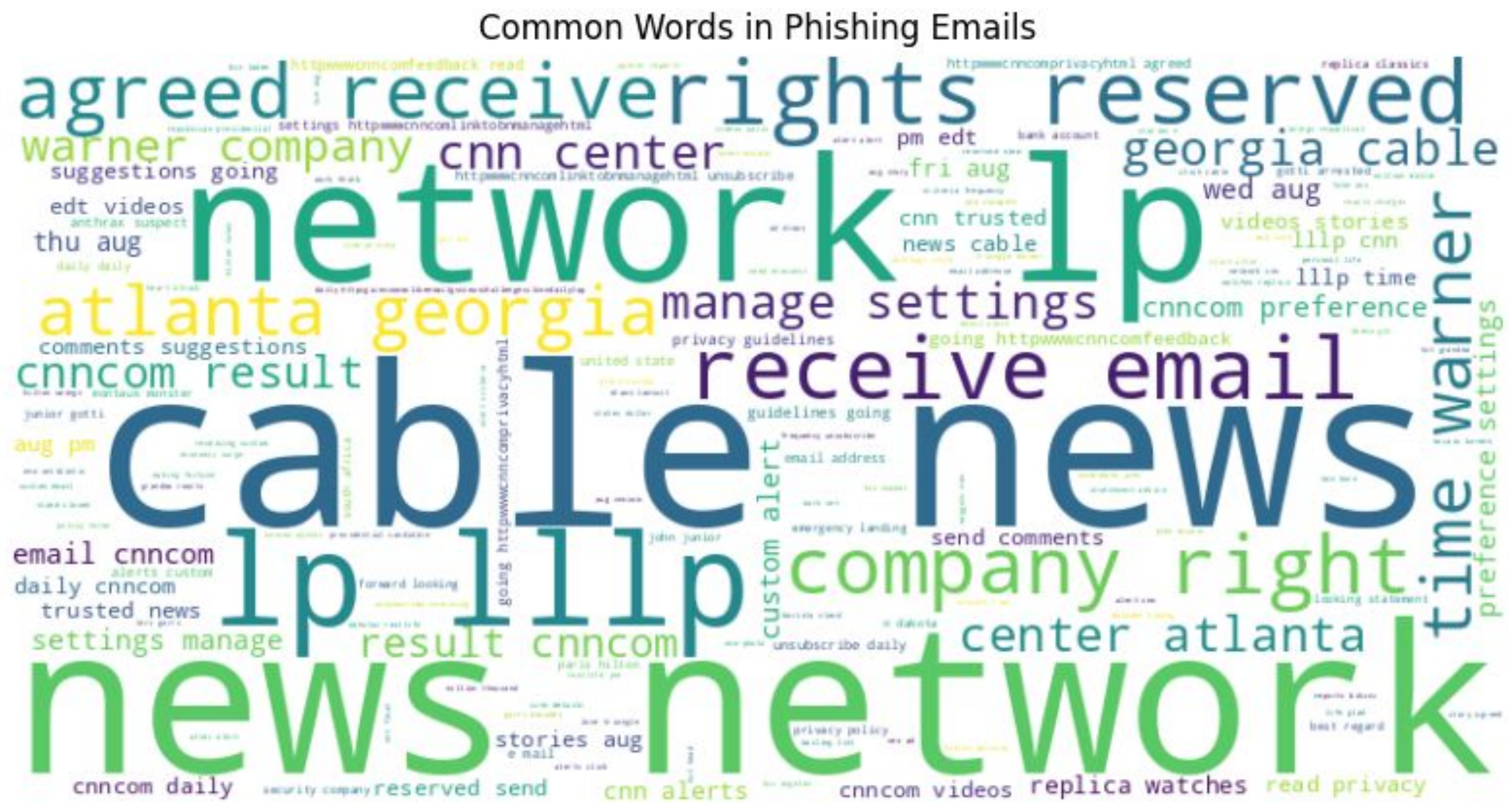
Data Science Workflow



Data Processing and Visualization

- ❖ Kaggle phishing email dataset
- ❖ Total email: 164552
  - ➢ Spam: 85729
  - ➢ Non-Spam: 78823

- ❖ Data cleaning:
  - ➢ Lowercasing
  - ➢ Remove URL and Email Address
  - ➢ Remove HTML Tags
  - ➢ Remove Special Characters
  - ➢ Trim whitespace



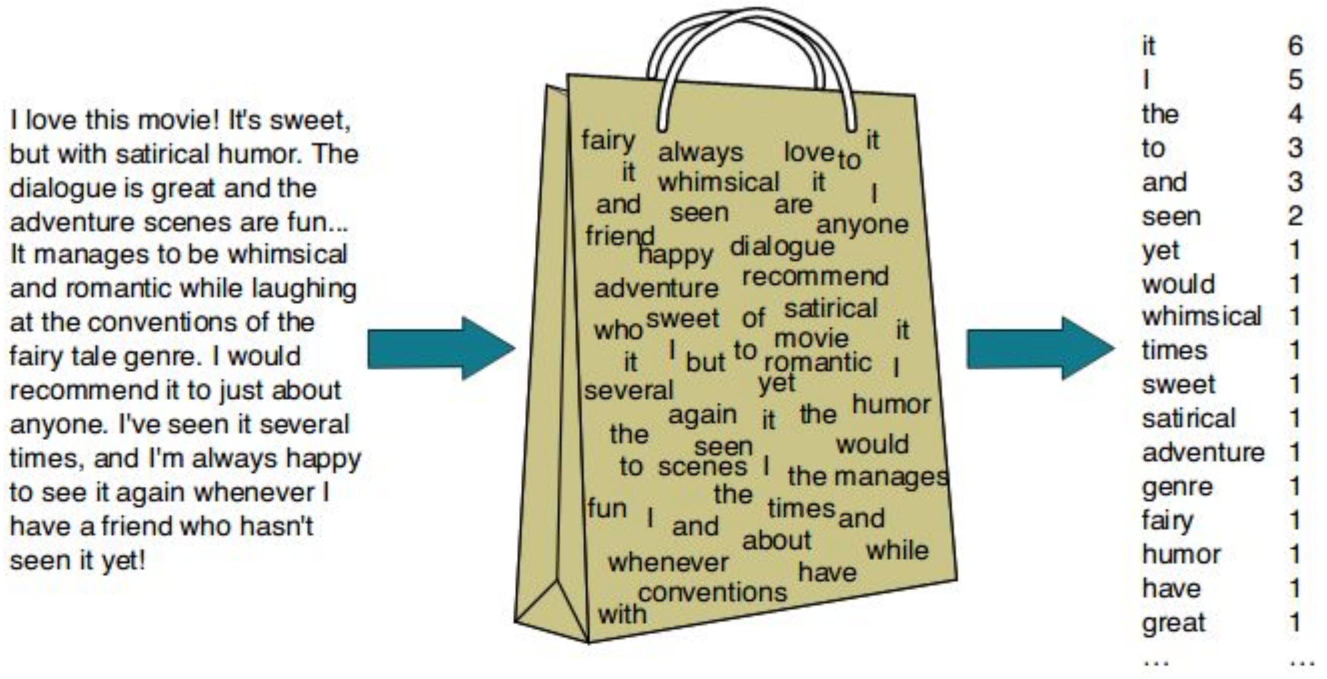
Feature Extraction

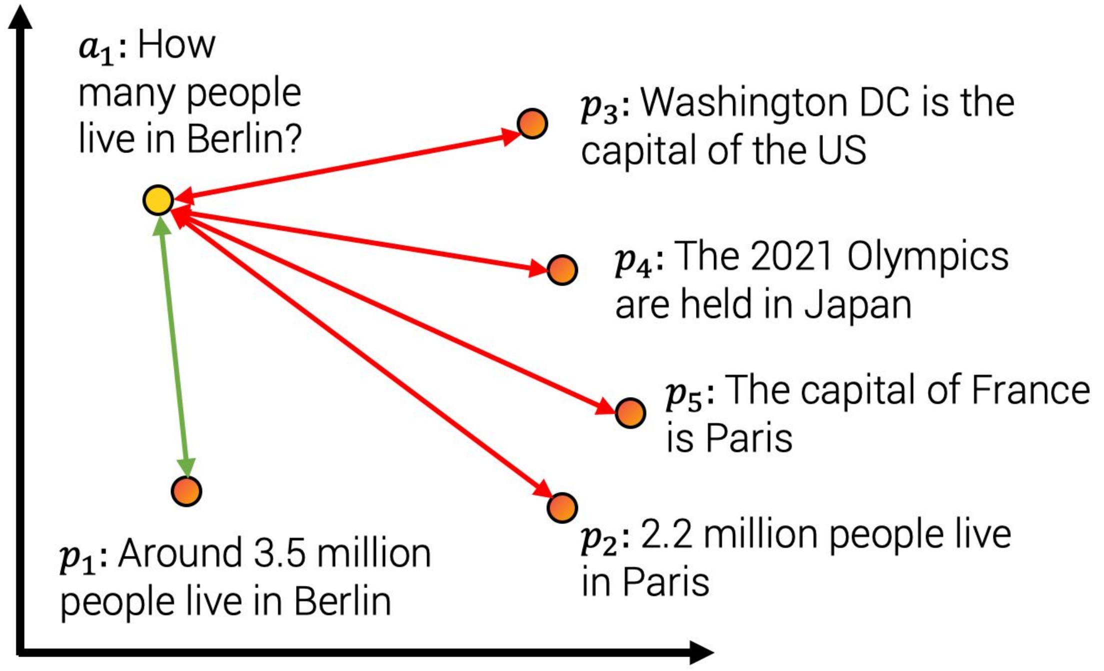
- **Bag of Words** is a simple text representation technique that converts text into fixed-length vectors based on word frequency.

- **TF-IDF** (Term Frequency–Inverse Document Frequency) measures how important a word is in a document relative to a collection of documents.
  - we build a word frequency with these important word

- **Sentence Embedding** represents an entire email as a dense vector that captures its semantic meaning.

- **Other Feature**: URL count





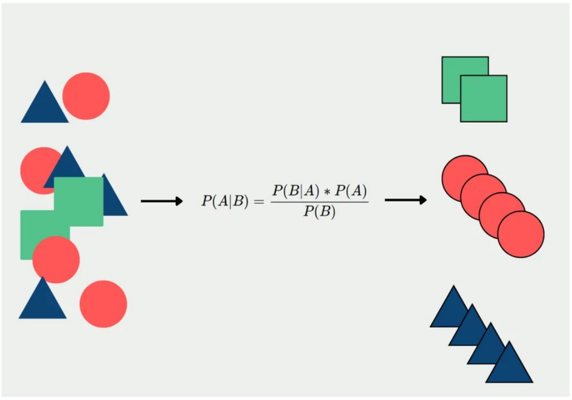
Model Selection

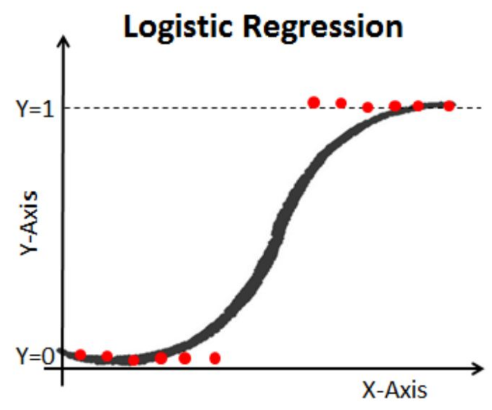
- Naive Bayes (Multinomial): A probabilistic classifier often used for text data, assuming feature independence and modeling word counts using a multinomial distribution.

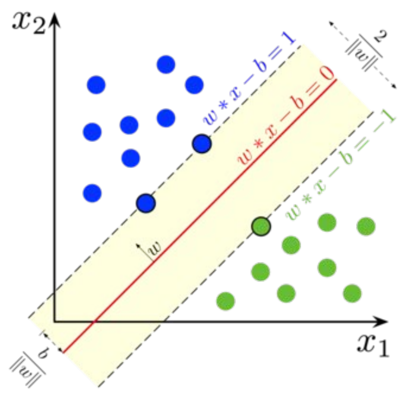
- Logistic Regression: A linear model that estimates the probability of a binary outcome using the logistic (sigmoid) function.

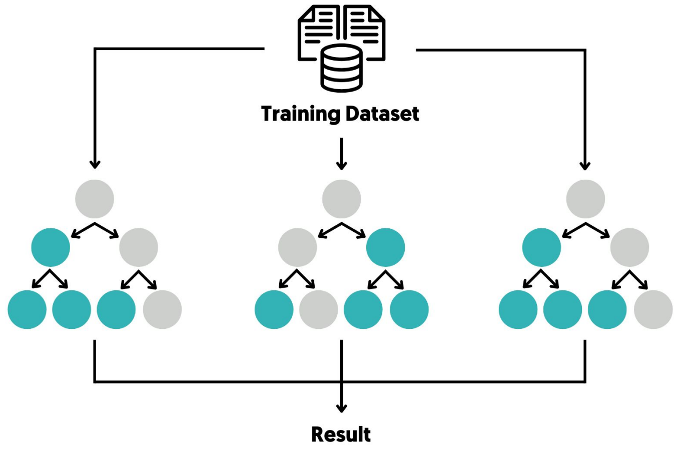
- SVM (Support Vector Machine): A discriminative classifier that finds the hyperplane maximizing the margin between classes in feature space

- Random Forest: An ensemble of decision trees that improves accuracy and robustness by aggregating predictions from multiple randomized trees.



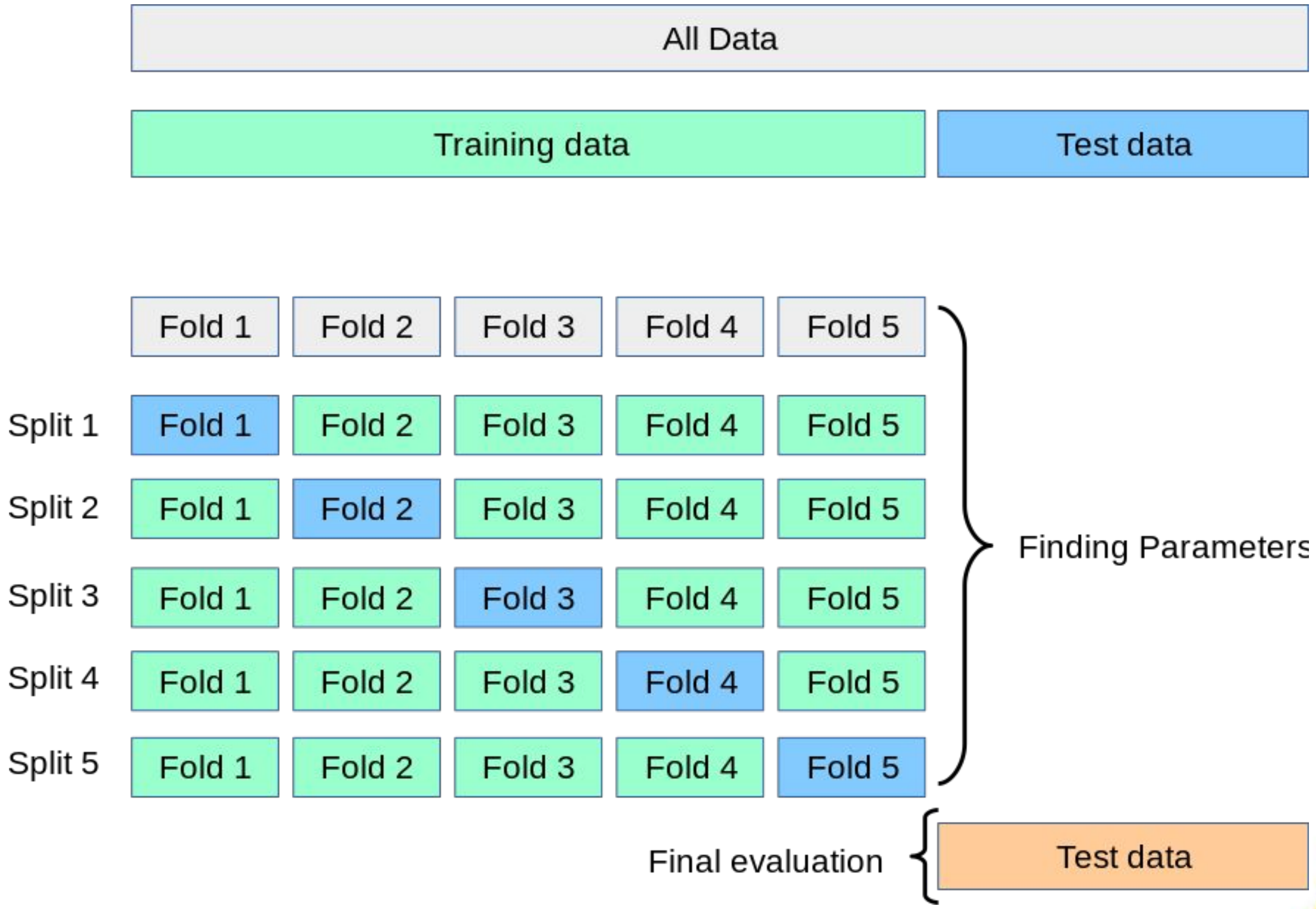






Model Evaluation and hyper-parameter tuning

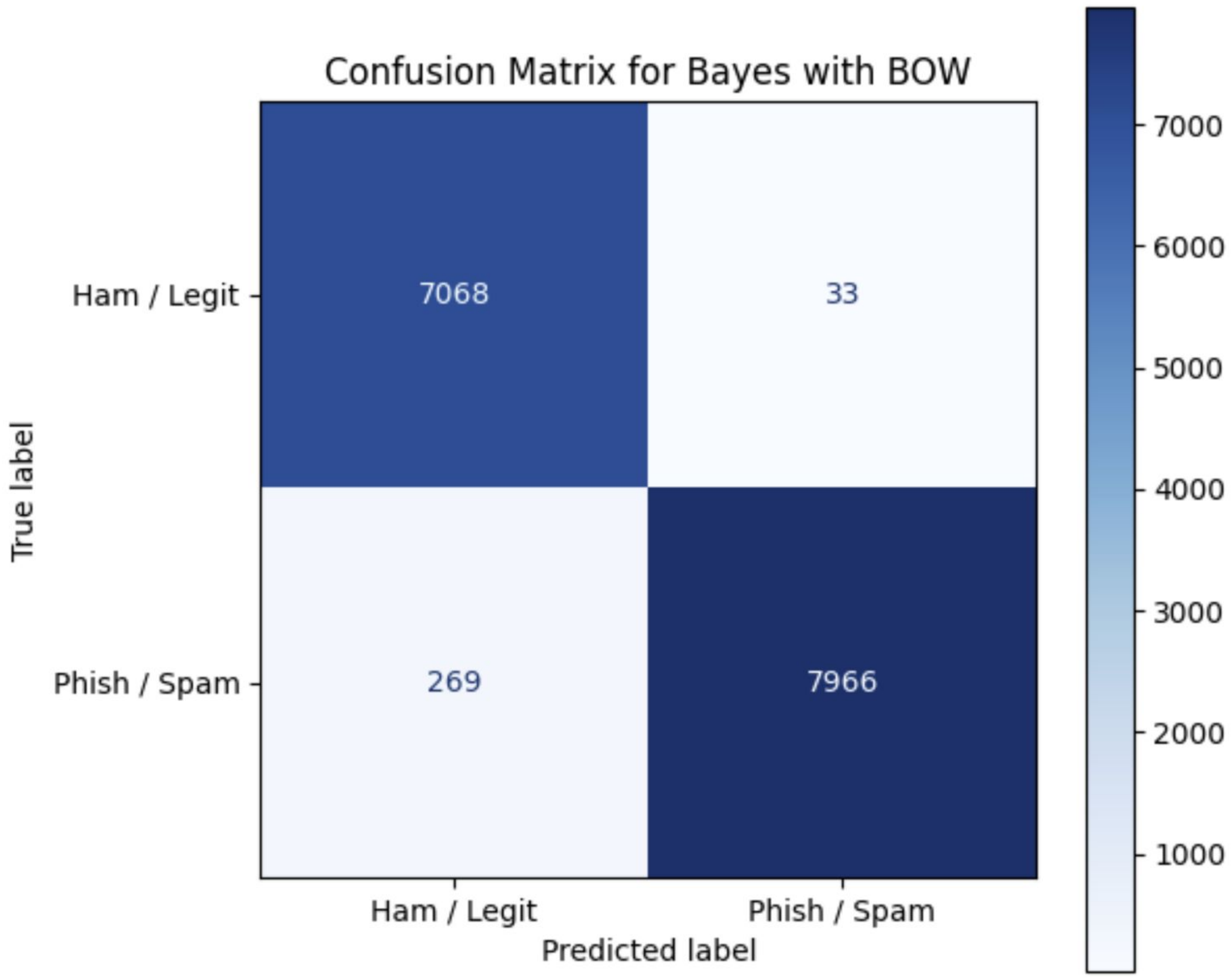
Split Data	Split data into training and testing dataset 80-20
k-fold cross validation	Do k-fold cross-validation on training dataset with our hyperparameters
Mean +- std	Calculate the mean and std of k-fold performance for each hyper parameter combination
Best param selection	Select hyperparameter that proceduce the high mean with a small std
Evaluation on test set	Evaluate on untouch testing dataset

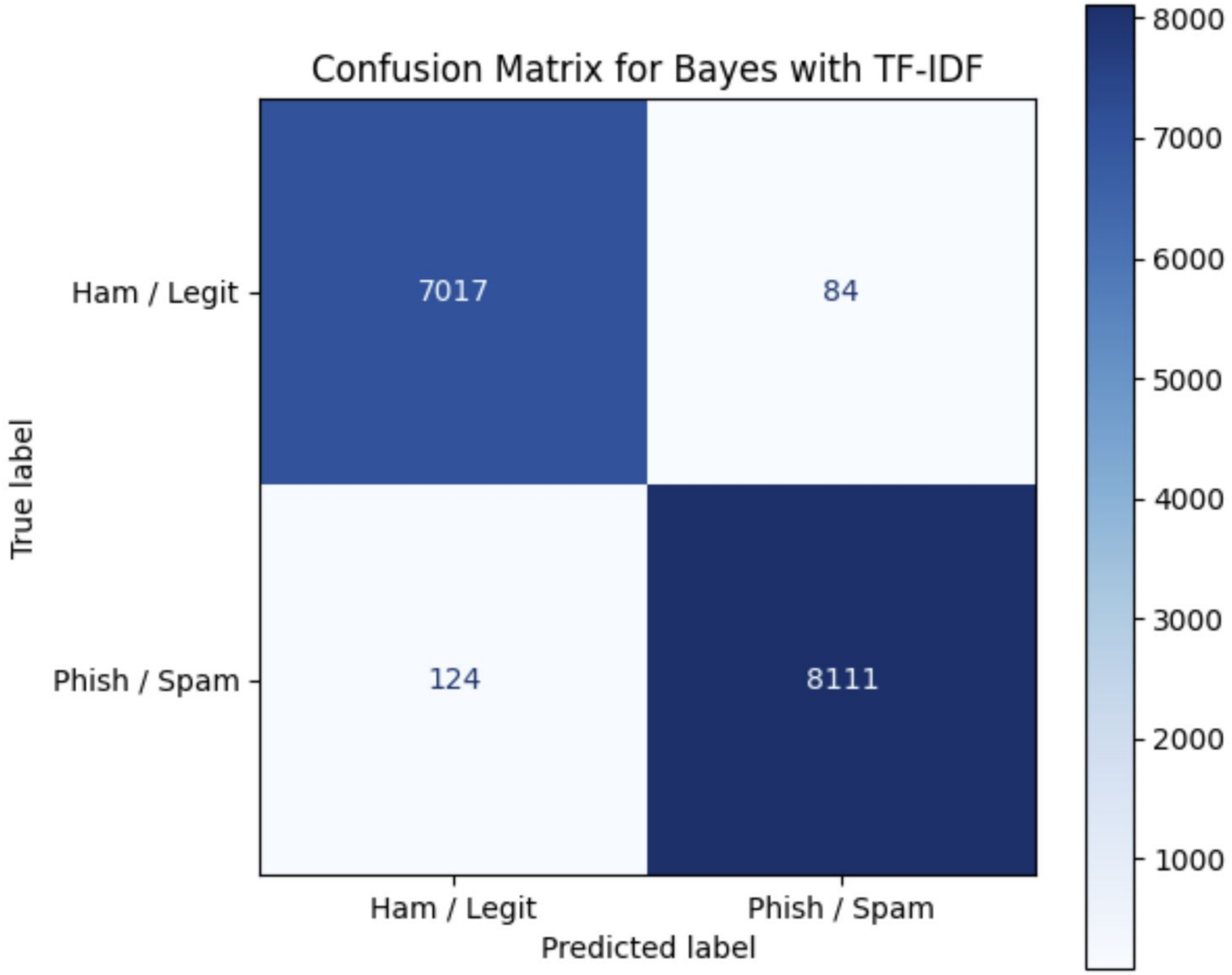


Result: Naive Bayes

Feature	Params	Accuracy	Precision	Recall	F1	AUC	Train & Predict Time
BOW	5000 features Laplace smoothing alpha = 1	0.9538	0.9756	0.9346	0.9547	0.9885	sub sec
TF-IDF	5000 features alpha = 1	0.9598	0.9789	0.9431	0.9607	0.9945	sub sec
BOW	1,017,504 features Laplace smoothing alpha = 1	0.9803	0.9959	0.9673	0.9814	0.9977	32.4s
TF-IDF	1,017,504 features Laplace smoothing alpha = 1	0.9864	0.9897	0.9849	0.9873	0.9991	33.3s

Result: Naive Bayes



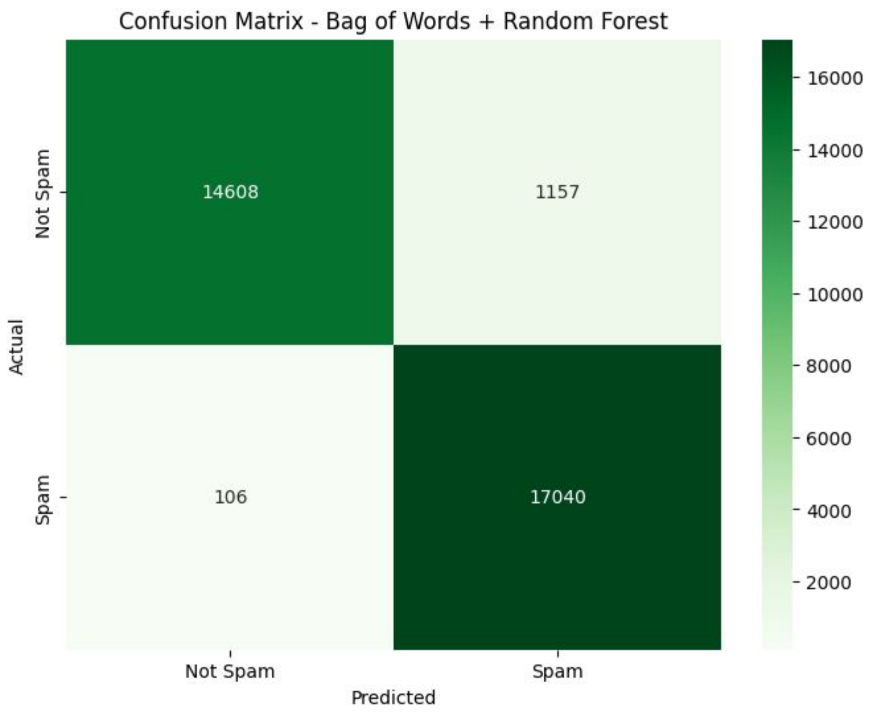


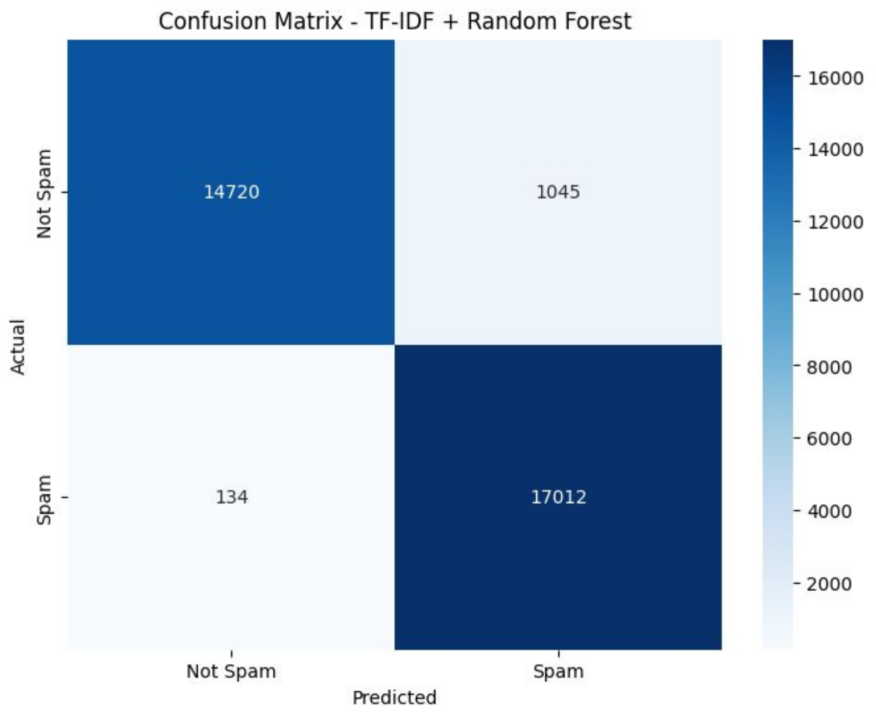
Result: SVM

Feature	Params	Accuracy	Precision	Recall	F1	AUC	Train & Predict TIme
BOW	5000 features kernel=linear	0.9904	0.9905	0.9903	0.9904	0.9984	3 hours
TF-IDF	5000 features kernel=linear	0.9877	0.9871	0.9893	0.9882	0.9989	3 hours
TF-IDF	5000 features kernel=rbf	0.9967	0.9962	0.9976	0.9969	0.9998	6 hours
Sentence Embedding	vector size = 384	0.9914	0.9914	0.9921	0.9918	0.9994	2 hours + 1 hour of embedding

Result: Random Forest

Feature	Params	Accuracy	Precision	Recall	F1	AUC	Train & Predict TIme
BOW	5000 features n_tree=50 max_depth=30 n_sample_per_split=4 n_sample_per_leaf=1	0.9627	0.9391	0.9929	0.9652	0.9955	6 minutes
TF-IDF	5000 features n_tree=50 max_depth = 30 n_sample_per_split=4 n_sample_per_leaf=1	0.963	0.9394	0.9931	0.9655	0.9959	6 minutes





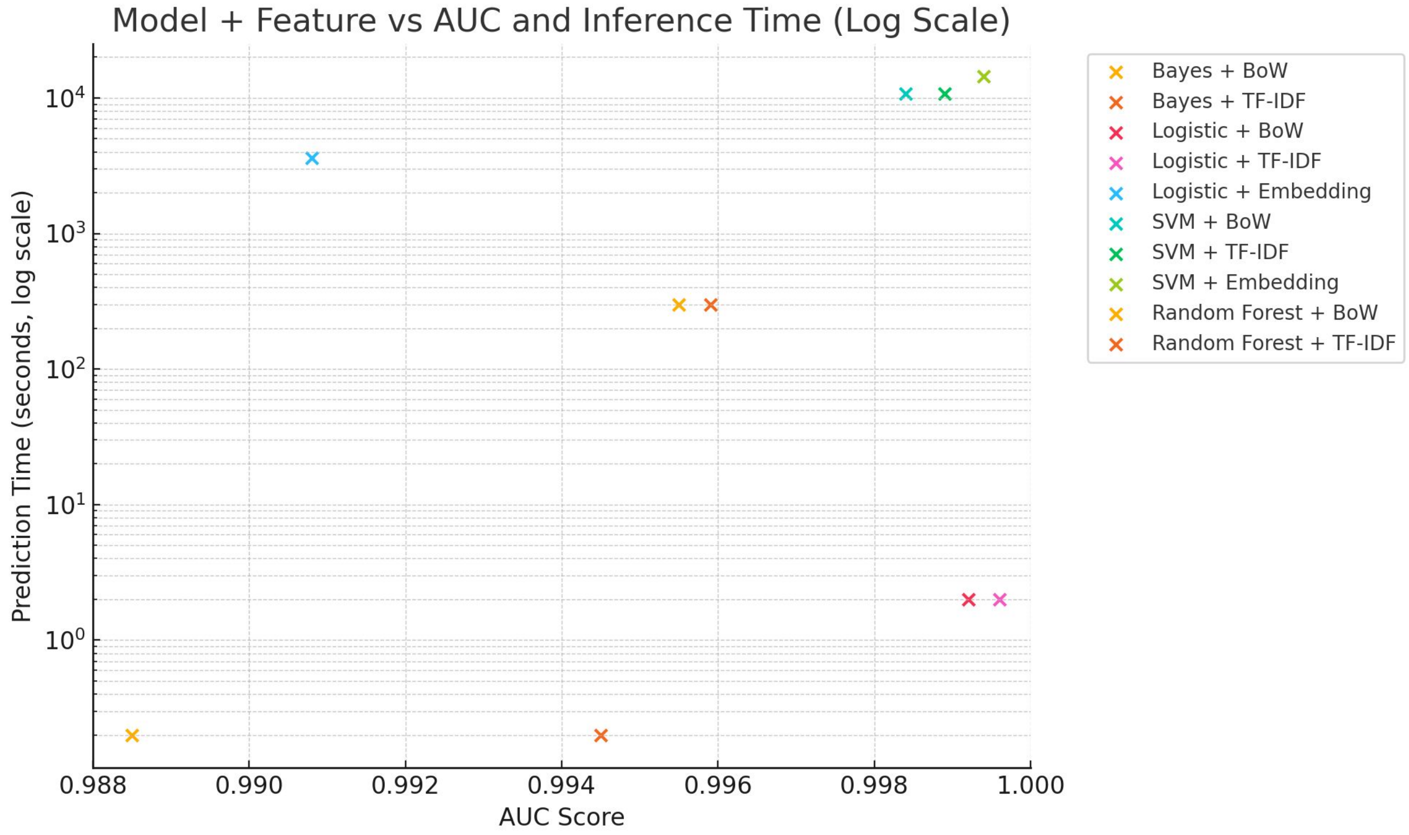
Result: Logistic Regression

Feature	Params	Accuracy	Precision	Recall	F1	AUC	Train & Predict TIme
BOW	5000 features C=10 Penalty= l2	0.9917	0.991	0.9931	0.992	0.9992	several seconds
TF-IDF	5000 features C = 100 Penalty = l2	0.9929	0.9924	0.9939	0.9932	0.9996	several seconds
TF-IDF + URL count	5000 features C = 100 Penalty = l2	0.9929	0.9924	0.9939	0.9932	0.9996	several seconds
Sentence Embedding	vector size = 384	0.9551	0.9535	0.9606	0.9571	0.9908	several seconds + 1 hour of embedding

Comparison - Model Deployment

- Metrics: High F1-score and AUC with reasonable prediction time
- Depend on our application

	BoW	TF-IDF	Embedding
Bayes	F1 & AUC: medium Really fast	F1 & AUC: medium Really fast	N/A
Logistic	F1 & AUC: high fast	F1 & AUC: high fast	F1 & AUC: high slow*
SVM	F1 & AUC: very high really slow	F1 & AUC: very high really slow	F1 & AUC: high really slow
Random Forest	F1 & AUC: medium really slow	F1 & AUC: medium really slow	N/A



Conclusion

- **ML-based models** can effectively detect phishing emails with high accuracy and robustness.
- **TF-IDF + Logistic Regression** and **SVM (RBF kernel)** delivered the best performance (F1 > 0.99, AUC > 0.999).
- **Sentence Embeddings** offer strong semantic understanding but are more computationally expensive.
- **Trade-off** exists between model accuracy and prediction time — critical for real-time prediction.
- **Recommendation**: Use Logistic Regression with TF-IDF for balance between performance and efficiency.

Thank you so much for your listening! ☺️