Our research question is “How can we successfully predict if a website is a phishing website or not?”. Phishing is an internet crime, in which a website pretends to be a legitimate enterprise to solicit sensitive information from people such as usernames, passwords and social security numbers. In today’s digital age, cyberattacks and phishing attempts are becoming increasingly common, which pose serious security risks to individuals and organizations. This project will use the Phishing Websites Dataset to uncover the most significant features and predictors, aiming to build a reliable model that can act against phishing threats and enhance cybersecurity awareness.

The dataset is from the UC Irvine Machine Learn Repository. It has 11055 instances and 30 features. Out of the dataset, 65.7% of the URLs have an IP address. For URL length, 17.7% have a URL length of less than 54 characters, 1.2% between 54 and 75 characters, and 81.0% have over 75 characters. Lastly, 85.5% of the URLs include an @ symbol in the link. For data cleaning, the original dataset does not have any missing values. Exploratory Data Analysis centered on finding the ratio of values based on feature. Using a simple bar chart, one could assess how each feature was divided in the three categories of phishing, not phishing, and suspicious. This step allowed us to fully understand how to handle the application of a third category, in only some features. Additionally, a chi-square statistic test revealing which features were highly correlated in a positive phishing result. We found that the 6 most correlated features had the third target variable, indicating its strength in predicting a result.

A **Logistic Regression** with Lasso(L1), Ridge(L2), and ElasticNet(L1 + L2) penalties were used in completing the penalty regression. Each regression used the pipeline method to prevent data leakage in the cross validation steps. GridSearchCV was used on model_C(inverse regularization strength) in all models and the ratio of L1 in solely the ElasticNet regression. The KFold was completed with 5 folds for each model. Performance was assessed using accuracy score and confusion matrix. The best performing model was the Lasso Logistic Regression with parameters model_C=0.1. For **SVMs**, we applied StandardScaler within a pipeline to establish the model in which all predictors contribute equally. We utilized RBF, polynomial, and linear kernels to capture both linear and nonlinear patterns within the feature space. GridSearchCV with 5-fold cross-validation was used for hyperparameter tuning, with the parameters being C(regularization strength), kernel (linear, RBF, and poly), gamma (curvature for RBF and polynomial kernels), and degree (for polynomial kernel). We used cross-validation accuracy as the scoring metric, with the best performing SMV using the RBF kernel with the parameters (C=100, gamma = 0.01, degree = 2) although the degree is unused for the RBF but is included because of the grid. This resulted in a cross-validation accuracy of 95.87%. In addition, we utilized scikit-learn’s **RandomForestClassifier** and performed hyperparameter optimization with GridSearchCV 5-fold cross-validation, tuning the parameters n_estimators (number of trees), max_depth (tree depth), and max_features (number of features considered per split). The best-performing cross-validation accuracy among all the models was with the parameters n_estimators = 200, max_depth = 20, and max_features = ‘log2’ with a score of 96.66%. Due to Random Forest being invariant to feature scaling, there was no need to apply standardization. The model also provides for us feature importance scores to identify which website features have the strongest contribution to phishing detection. For the **Neural Network**, we used

torch.manual_seed() to ensure consistent results and converted -1 values in the result column to 0. Then, we set up two loops for cross validation where the first loop was the combination of different hyperparameters with itertools containing hidden size, learning rate, and batch size and second loop being the KFold with 5 splits. Lastly, we used Binary Cross Entropy as a criterion since the results are in binary and assessed the model performance with accuracy score.

For each of the aforementioned models (Penalty Regression, SVM, RandomForest, and Neural Network), we got the results presented in the table below:

Model	**Hyperparameter**	**Results**
Lasso Logistic	{'model__C': 0.1}	93.08%
SVM	{'svm__C': 100, 'svm__degree': 2, 'svm__gamma': 0.01, 'svm__kernel': 'rbf'}	95.87%
Random Forest*	{'max_depth': 20, 'max_features': 'log2', 'n_estimators': 200}	96.66%
Neural Network	{'hidden_size': 128, 'learning_rate': 0.001, 'batch_size': 128}	96.64%

The Random Forest model has the highest performance among all the models after only using the training dataset. We then evaluated the Random Forest model by training it on the entire training set and testing its accuracy with the testing set. We got an accuracy of 96.83% with a precision score of 0.98 for -1 (phishing) and 0.96 for 1 (legitimate) and a recall score of 0.95 for -1 (phishing) and 0.98 for 1 (legitimate).

An accuracy of 96.8% is an outstanding result for this dataset. For the confusion matrix, the model correctly identified 908 phishing websites and 1,233 legitimate websites. The precision and recall score indicates it made very few False Positives or False Negatives as well. The reason why Random Forest may perform better than a neural network is that it splits the data and fits its discrete nature, while neural networks treat numbers as continuous. And compared to Penalty Regression, it draws a straight line through the data, which may not capture the complexity of the dataset as well. Lastly, SVM may not yield the best performance as noisy data can make it hard for the model to find the margin.

One limitation is that the UCI Phishing dataset is quite old as it was posted back in 2015. Many things may have changed and advanced in the real world when it comes to phishing attempts that are inconspicuous. In addition, the dataset has a lack of content as it only has details on the URL and network properties. A URL may look perfectly fine, but it can still be a phishing website if it tries to solicit sensitive information. Finally, the 0 value in some of the columns creates a lot of ambiguity with it meaning suspicious. There are many ways to interpret what suspicious means and manipulating it mathematically can be a real challenge when one is trying to train models.