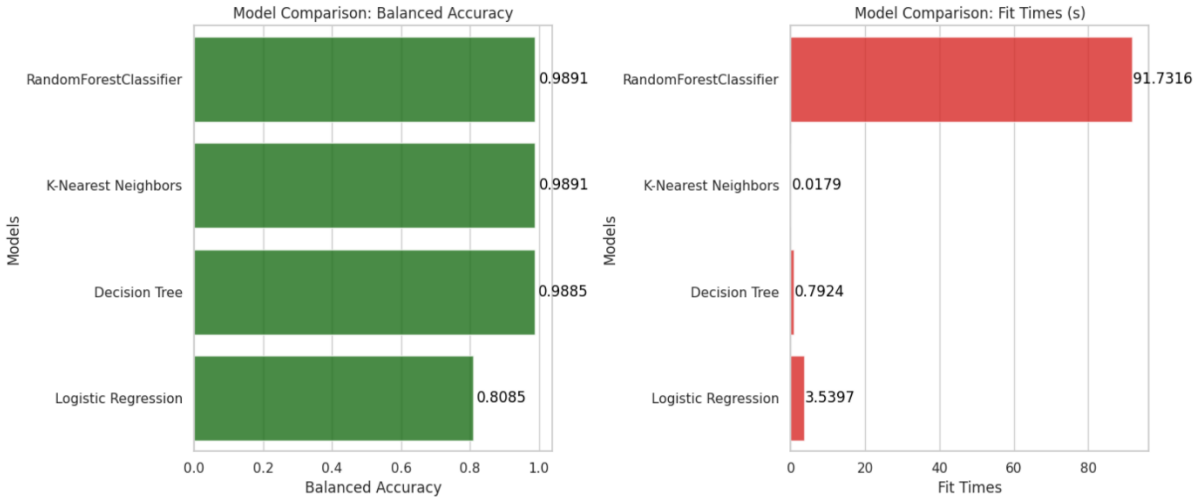
Model Card: Binary Classification

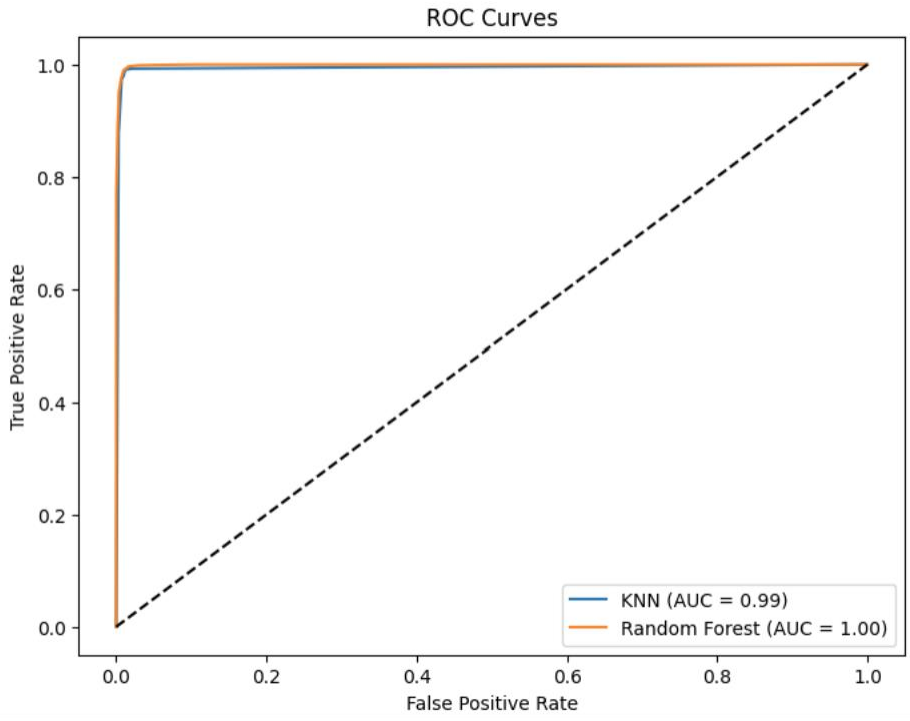
1. **Models Evaluated**
    a. LogisticRegression with L1 regularization
    b. LogisticRegression with L2 regularization
    c. RandomForestClassifier
    d. DecisionTreeClassifier
    e. KNeighboursClassifier
    f. SVM
2. **Use Case**
    a. Network traffic data from packet sniffers was gathered already and provided as a clean dataset for training and testing (Please refer to data sheet) for further information.
    b. **In scope:** We perform usual EDA (exploratory data analysis) and data manipulation before we feed in the data for binary classification.
    c. Out of scope: Obtaining real data on network threats is a hard task in this case we were fortunate to get the synthetic and original data. (please refer to data sheet) for further information. Generating synthetic data is by itself an effort worth doing as a project.
    d. We use this dataset to do our classification analysis and produce results.
3. **Factors**
    From the start our models performed above average balanced accuracy > 85%. This is in a large part because the data was synthetic and clean. We took precautions like hiding 30% of the data from training and testing sets and eventually moved forward to do final evaluation of the models. Needless to say results above average attaining 98%+ accuracy.
4. **Metrics**
    In large part first pass model selection depended largely on training times, fit times and balanced accuracy.

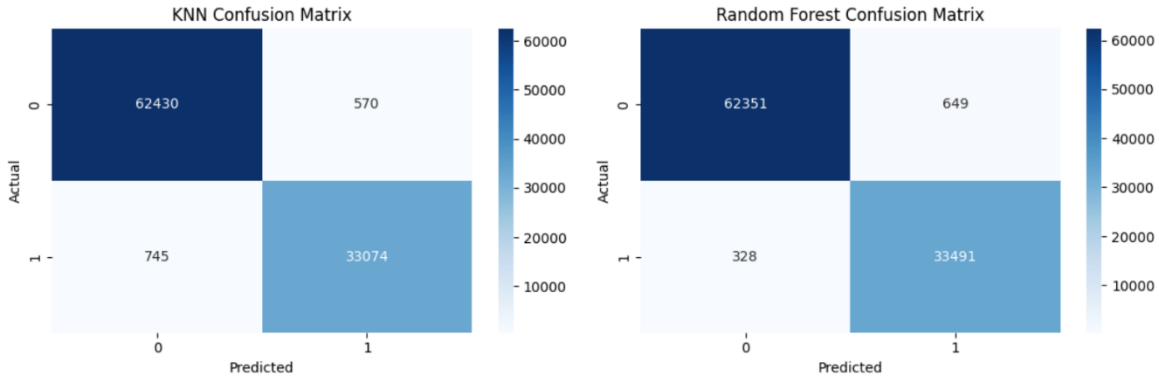
Narrowed down top four model benchmarks:



index	Model Name	Balanced Accuracy	Fit Time (s)	Interpretable	best_params
0	RandomForestClassifier	0.9891	91.7316	No	{'max_depth': 15, 'min_samples_split': 2, 'n_estimators': 300}
1	K-Nearest Neighbors	0.9891	0.0179	No	{'n_neighbors': 5, 'p': 1, 'weights': 'distance'}
2	Decision Tree	0.9885	0.7924	Yes	{'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 2}
3	Logistic Regression	0.8085	3.5397	Yes	{'C': 70.54802310718645, 'max_iter': 100, 'penalty': 'l2'}

Top two model ROC Curve and Confusion Matrix





5. **Dataset**

   Please refer to the data sheet to get mode details.

6. **Caveats and recommendations**

   Post data cleaning, encoding, and scaling we are left with 49 features. Reducing the features is critical for meaningful model training and fitting compute times. Critical feature reduction and selection was performed by performing correlation matrix and RandomForestClassifier.

   Even with features reduced to top 25 or even top 15 features SVM was computationally expensive.

   SVM evaluation was aborted for that reason. GridSearchCV even with 15 features with google colab free tier compute took multiple hours.

Recommendation would be to be cautious and reduce the data set to absolute minimum features to train models. If time is critical (often that is the case with cybersecurity) recommendation would be to use KNN classifier for binary classification to identify threats vs normal traffic.

7. **How to use it in your organization?**
   Generate network traffic (headers from packet captures) these can often be too large to handle. Exercising caution to generate just enough not more than 1 Million would be a great idea to train the model. Input data may have to be cleaned up appropriately before feeding it into the model. If this model is in production developing KPI (Key Performance Index) of the model is important to prevent model drifts. You may have to retrain the model since new threats are developed more frequently now.