Microsoft Malware Detection Classification Model

Date: Oct 21 2023
Author: Minghan Yang, 3831285746

1. Data Preprocessing

a) Data Loading:
- Loaded both training and test datasets. Since the datasets are too large, so I choose 10% of these two datasets.

b) Feature Selection:
- Based on domain knowledge and data exploration, the following features were identified to potentially improve the model:
  - 'ProductName', 'SmartScreen', 'Firewall', 'IsProtected', 'AVProductStatesIdentifier', 'EngineVersion', 'AppVersion', and 'AvSigVersion'.

c) Data Cleaning:
- Handled missing values in the dataset.
- For features with categorical data types, filled null values using mode values.

d) Data Encoding and Feature Scaling:
- Utilized LabelEncoder to transform categorical features like 'ProductName', 'SmartScreen', 'EngineVersion', 'AppVersion', and 'AvSigVersion' into numeric labels.
- Applied StandardScaler to scale the 'AVProductStatesIdentifier' feature to ensure its values are on the same scale as other features, facilitating better model convergence.

2. Classification Modeling

a) Logistic Regression:
- Confusion Matrix:

```
[[ 75 532]
 [ 30 541]]
```

- Accuracy: 52.29%
- Precision: 50.42%
- Recall: 94.75%
- F1 Score: 65.82%

b) Random Forest:
- Confusion Matrix:

```
[[603    4]
 [555   16]]
```

- Accuracy: 52.55%
- Precision: 80%

- Recall: 2.80%
- F1 Score: 5.41%

c) Decision Tree:

- Confusion Matrix:

```
[[577   30]
 [464 107]]
```

- Accuracy: 58.06%
- Precision: 78.10%
- Recall: 18.74%
- F1 Score: 30.23%

d) KNN:

- Confusion Matrix:

```
[[558   49]
 [523   48]]
```

- Accuracy: 51.44%
- Precision: 49.48%
- Recall: 8.41%
- F1 Score: 14.37%

e) ANN:

- Confusion Matrix:

```
[[217 390]
 [114 457]]
```

- Accuracy: 57.22%
- Precision: 53.96%
- Recall: 80.04%
- F1 Score: 64.46%

3. Findings:

- Logistic Regression: Produced the highest recall, suggesting it's good at identifying positive malware instances. However, its precision is modest, which implies a fair number of false positives.

- Random Forest: Despite a high precision, its recall is abysmal. This implies it's over-predicting the negative class, and as a result, it's missing a significant number of positive malware instances.

- Decision Tree: Offers a balanced trade-off between precision and recall, making it a considerable model choice for this application.

- KNN: Has one of the lowest accuracies, and its recall is quite low, suggesting it may not be the best fit for this data.

- ANN: Showcases a strong balance between precision and recall, making it a strong contender alongside the Decision Tree.

4. Conclusion:

Based on the metrics, the Decision Tree and ANN models appear to be the most promising for this application. The choice between them depends on the specific operational context. If ensuring malware doesn't go undetected is crucial, ANN's higher recall might be preferred. However, if computational resources and model interpretability are a concern, the Decision Tree could be the choice.