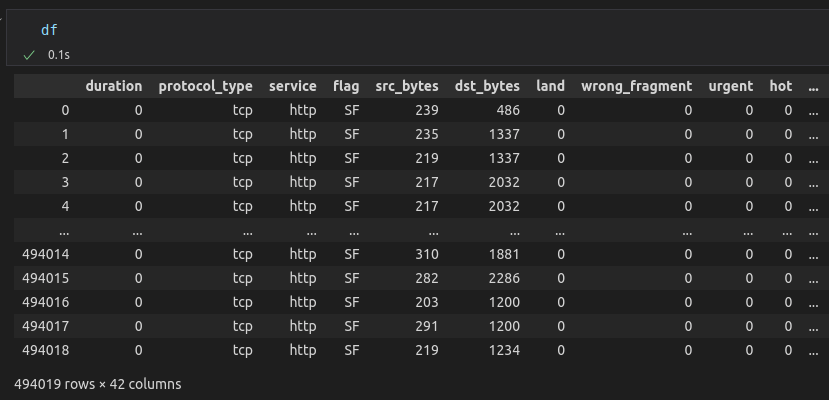
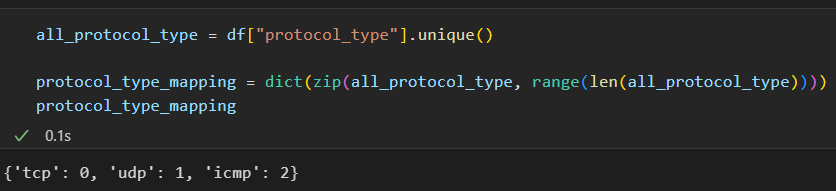
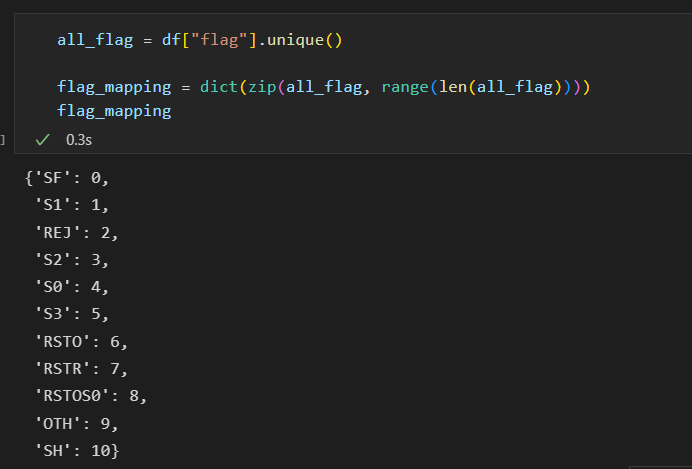
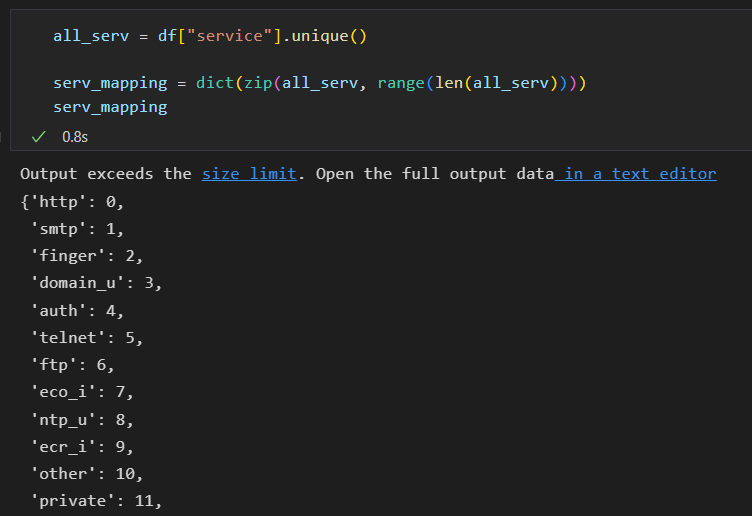
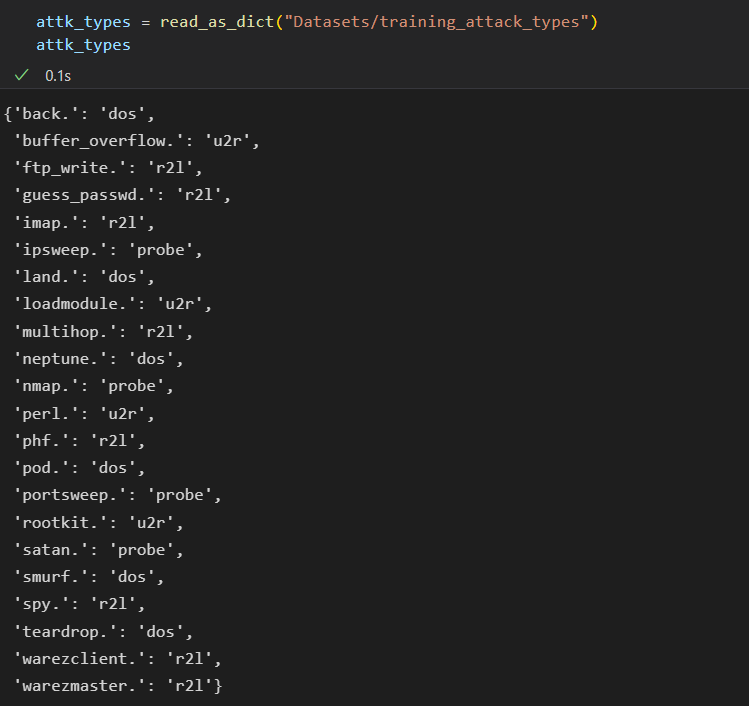
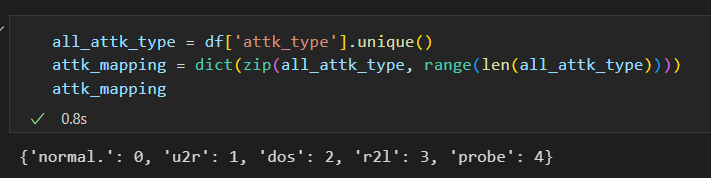
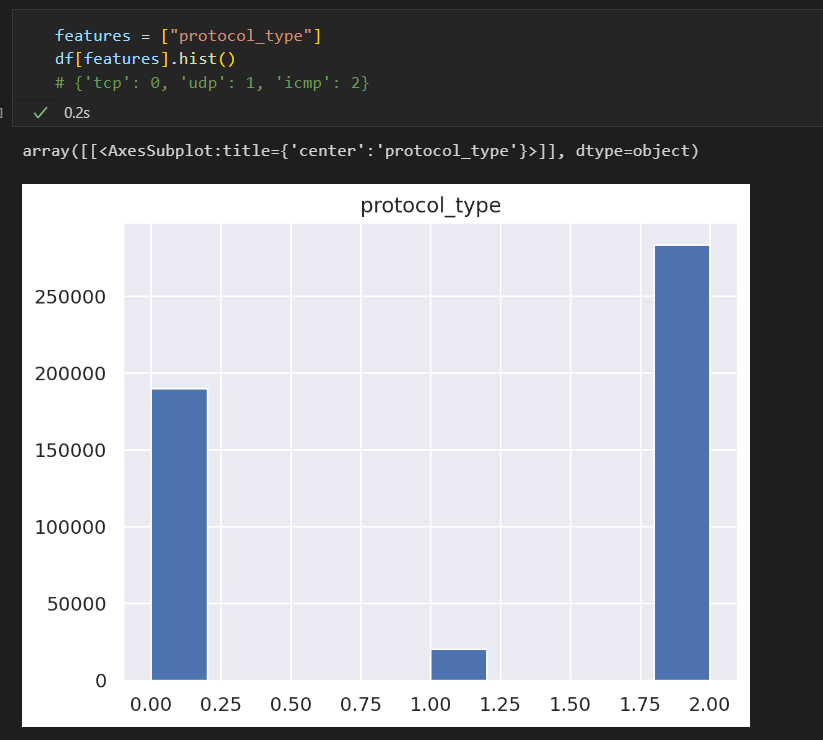
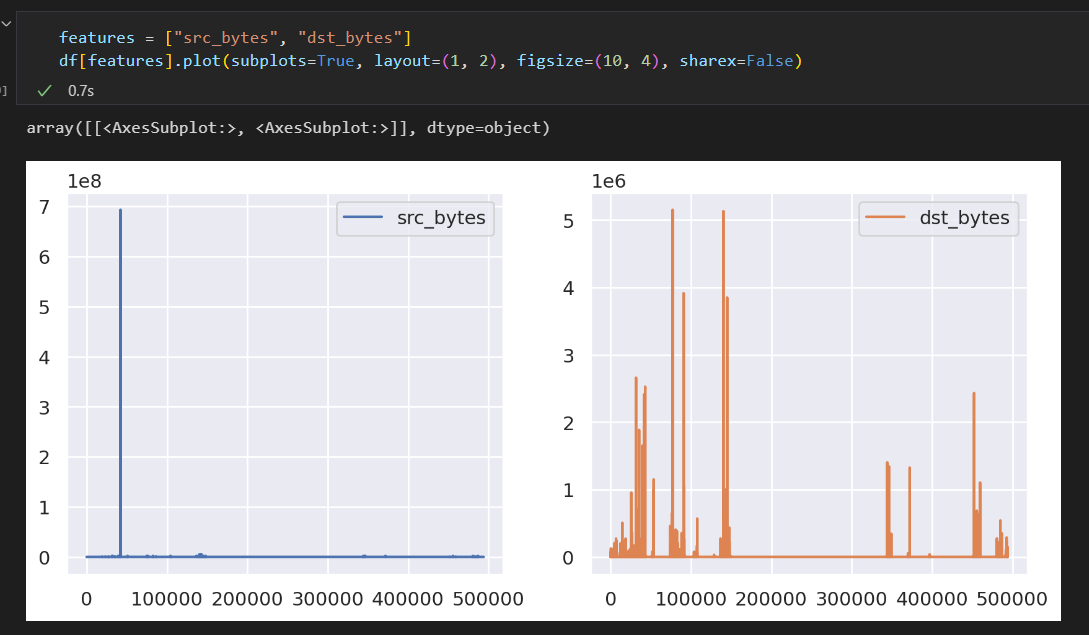
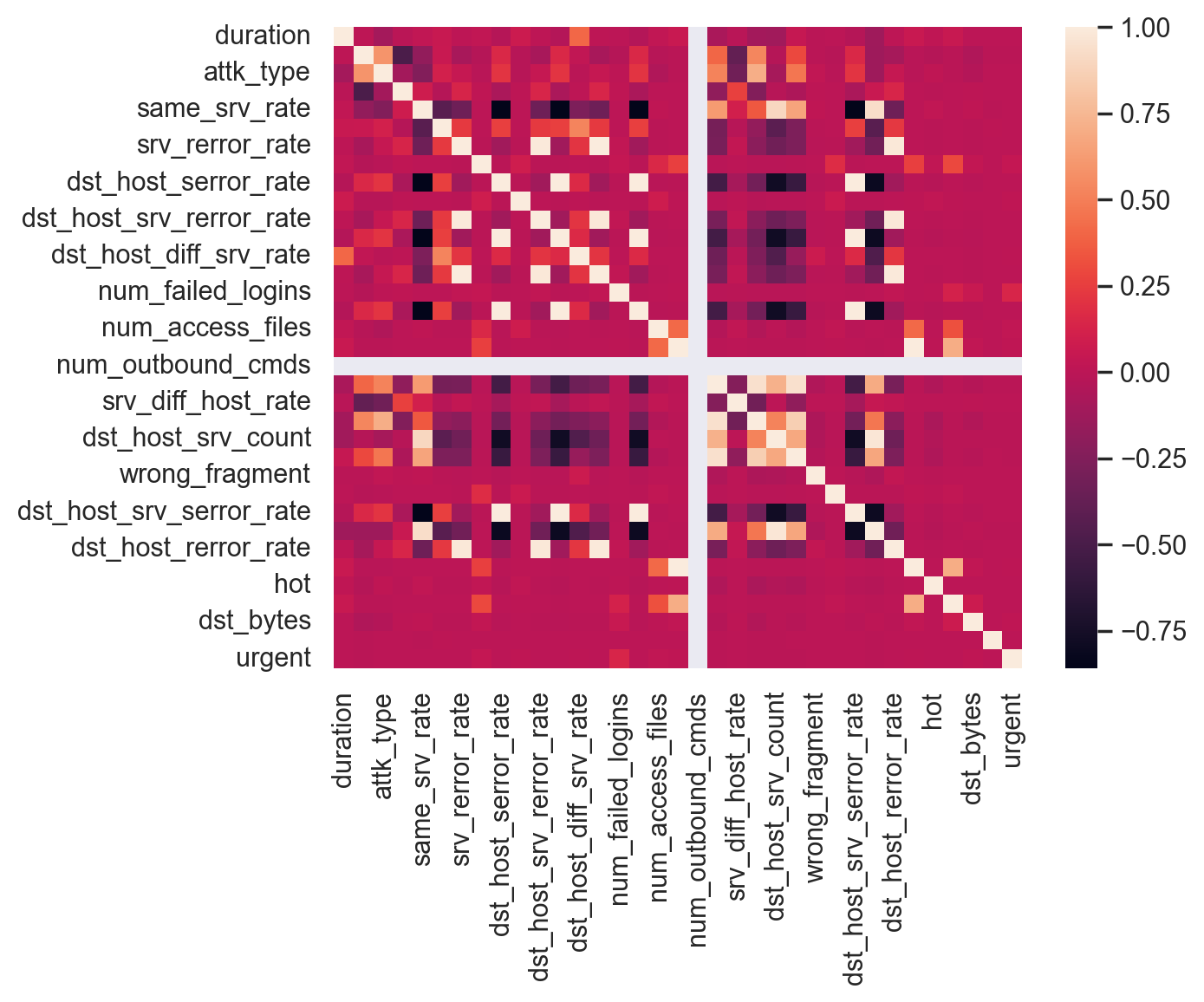
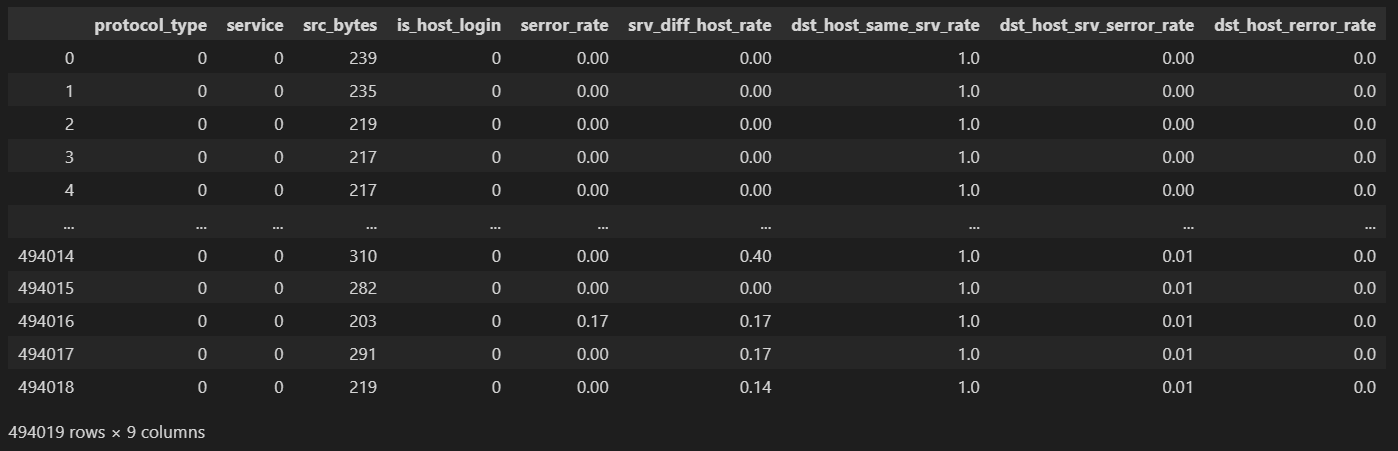
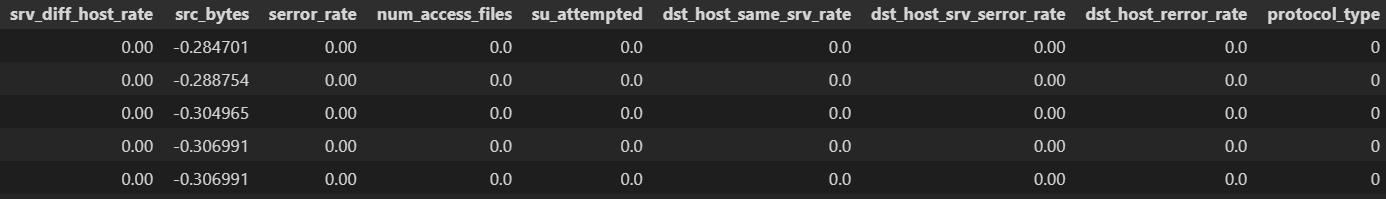
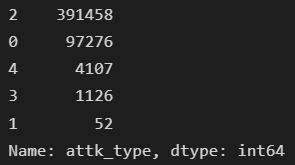
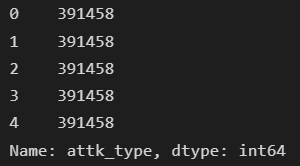
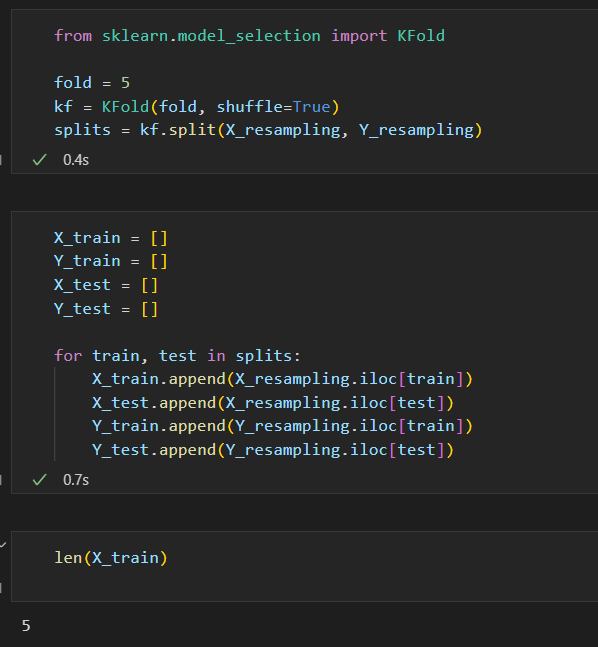
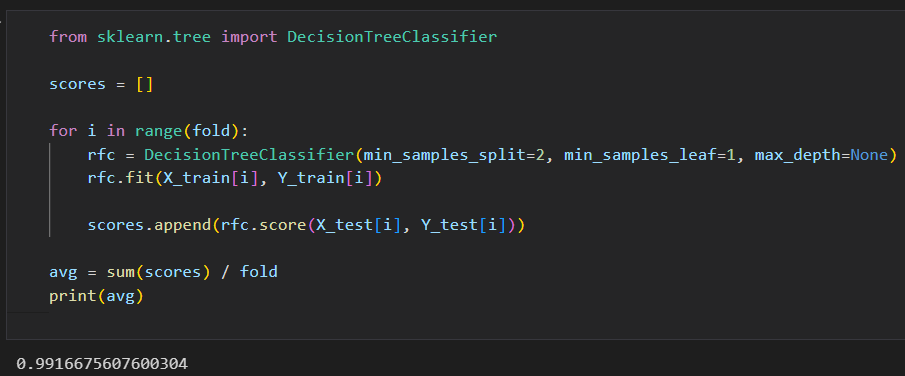
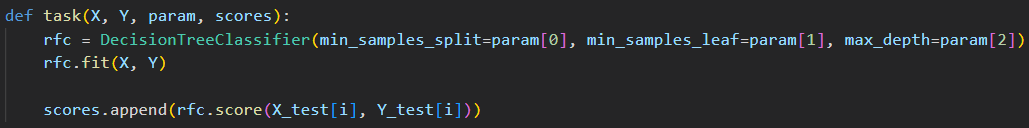
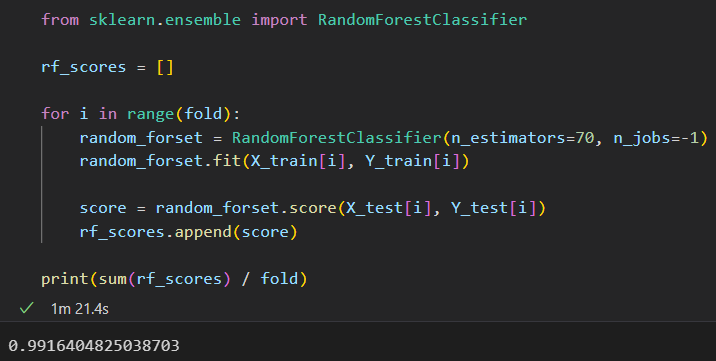
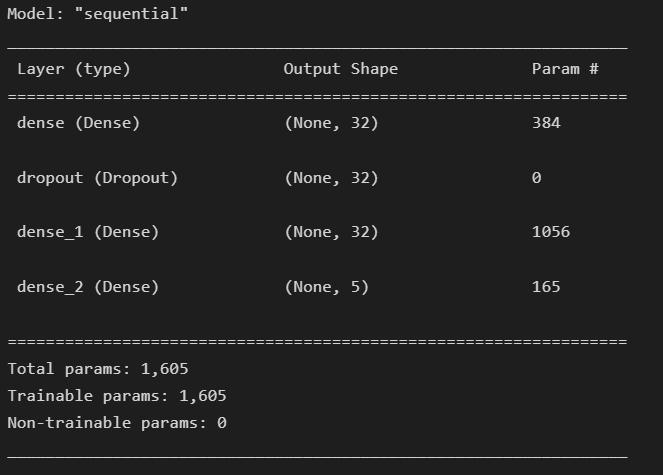
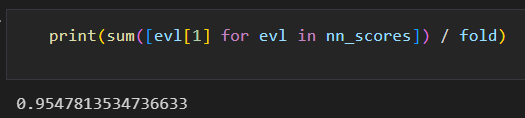
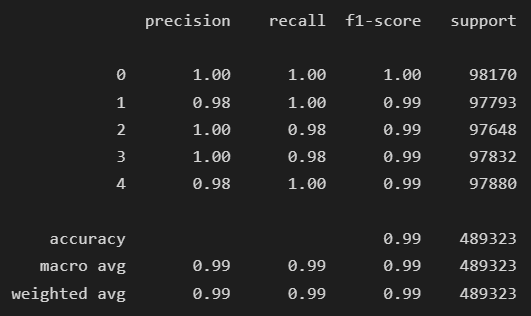
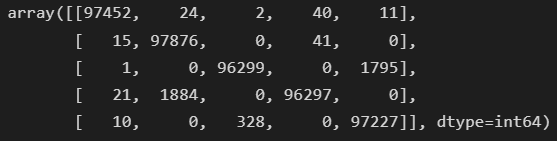
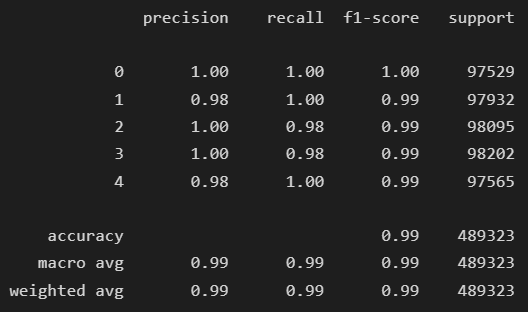
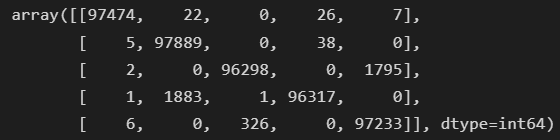
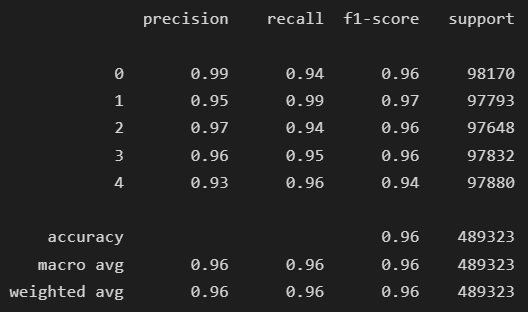
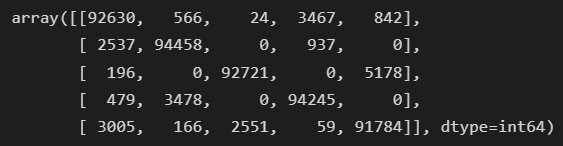
MLFN Lab1 Report

資科工一 游庭瑋

1. 資料前處理

首先，直接先使用pandas讀取指定資料集，格式使用csv，會發現其中有一行出現無法讀取錯誤，直接刪去該行。  
之後重新讀取資料集，會發現此檔案沒有column header，因此將標頭資訊kddcup.names讀入之後寫入標頭，如下:



1. 轉換非數值資料  
   資料集裡面有部分非數值資料，要將其轉為用數字代表的symbol，方便後續處理  
   須處理的feature: protocol\_type, flag, service  
     
     
     
     
   而要預測的目標(attk\_type)則需特別處理，因為原本的類別必須被轉為4種攻擊類別及正常5種類別，並再轉為數字。根據資料集給的資訊，可以整理出下列對應表  
     
   再由這個對應表依序對應到數字上  
     
     
   整理完的資料集即不會出現文字的feature
2. 資料視覺化  
   下圖為分析傳輸協議種類，0為TCP，1為UDP，2為ICMP。  
     
     
   下圖則是src\_bytes和dst\_bytes的長條圖分析，其中橫軸單位為bytes  
     
     
   可以看出，src\_bytes非常集中在某一個區塊，因此可以推測src\_bytes並不是一個在分析上很好的feature。  
   下方展示的是熱圖，熱圖是各個feature之間的相關性程度視覺化的呈現，因此熱圖只適用於純數值資料，所以在繪製熱圖時必須先將非數值資訊先移除才會有合理的分析。  
   
3. 特徵選取  
   這裡使用scikit-learn的SelectFromModel來選取特徵，使用的模型是RandomForest  
     
   接著加上一些人工判斷認為重要的參數，如protocol\_type(dos攻擊常用UDP)、su\_attemped(嘗試取得su權限相對是危險行為)、num\_access\_files(取用大量檔案通常也是異常行為)  
   因此我們共有11個features: protocol\_type, service, src\_bytes, is\_host\_login, serror\_rate, srv\_diff\_host\_rate, dst\_host\_same\_srv\_rate, dst\_host\_srv\_serror\_rate dst\_host\_rerror\_rate , su\_attempted, num\_access\_files.
4. 特徵轉換  
   為避免有離群值的出現，因此將特徵以RobustScaler做轉換。RobustScaler會自動刪去低於1st IQR及高於3rd IQR的資料
5. OverSampling  
   在統計原始資料集時，有發現各種攻擊類別的分布極度不平均  
     
   為避免在後續訓練失準，這裡將資料做OverSampling，讓較少的資料仍有多次的學習機會  
   
6. K-fold Cross Validation  
   這裡使用K=5作為參數，並將結果裝進list中  
   
7. 訓練模型  
   這裡選用三種，分別為Decision Tree, Random Forest, MLP
   1. Decision Tree  
        
      測試100個iteration，找出最佳參數  
        
      
   2. Random Forest  
      
   3. Neural Network (MLP)  
      這邊使用tensorflow keras，以GPU加速運算  
      下圖為模型概況  
        
      平均分數  
      
8. Precision, Recall, F1 score, Confusion Matrix  
   這裡將完整的資料集切分成training set和test set後，分別訓練上述三種模型，並得出結果  
     
   Decision Tree  
     
     
     
   Random Forest  
     
     
     
   Neural Network(MLP)  
     
   
9. 討論與總結  
   在這次實驗中，我們使用了各式資料工程的方法來將資料透過機器學習模型分析並預測。有發現到的是，這筆資料在各個攻擊類型分布上相當的不平均，甚至有幾萬對十幾筆的狀況，因此若沒有使用OverSampling，出來的結果會幾乎無法預測較少比例的分類；但就這裡的測試結果，如果用平衡資料再次做測試時，該分類的準確率也相當的低，因此資料收集還是相當的重要的。另外，還有發現如果使用scikit learn內建的MLPClassifier，比起tensorflow等專門深度學習的套件會慢上許多，也少了很多的彈性，因此若需要使用MLP等深度網路，還是直接使用tensorflow, pytorch等框架才會有較高的效率。