Predicting a User's Persona Using Computer's Specifications, CPU Utilization, CPU Temperature & Application Usage Time

Keshan Chen | Vince Wong | Jonathan Zhang
kec180@ucsd.edu | vlw003@ucsd.edu | jsz002@ucsd.edu
DSC180B B09 - Intel
March 7, 2021

Abstract

During the first half of this project, we learned about Intel's telemetry framework. The framework allows remote data collection from devices with Windows operating systems. Two important components of the telemetry framework are the Input Library (IL) and Analyzer Task Library (ATL). The IL exposes metrics from a device and the ATL generates on-device statistics from the data collected by the IL. In the second half of the project, we used pre-collected data provided by Intel that used their telemetry framework to create a classification model. Our goal with the model was to predict the persona of a user using their computer's specifications, CPU utilization, CPU temperature, and time spent on certain types of applications. User personas were provided by Intel which classified if users were casual web users, gamers, communication, etc.. The classifications of these personas were done by Intel based on the amount of time users spent on certain applications based on their usage of different types of .exe files. For example, if a majority of a device's time is spent on an application like Skype, they are most likely classified as a communication user. Similarly, if a user spends a majority of their time on the League of Legends .exe file, they are most likely classified as a gamer. After training multiple classification models, we were able to predict user personas with 64% accuracy using a gradient boosting classifier. In the following paper, we will discuss our hypotheses, processes, methodologies, and results.

CCS Concepts

• **Computing** → Machine learning
• **Mathematics of Computing** → Probability and statistics; Mathematical analysis

Keywords

Telemetry, gradient boosting classification, extra trees classification, AdaBoost classification, neural networks, scikit-learn

ACM Reference format

Vince Wong, Jonathan Zhang, Keshan Chen. 2021. Predicting a User's Persona Using Computer's Specifications, CPU Utilization, CPU Temperature & Application Usage Time. *La Jolla , CA, USA, 5 pages.*

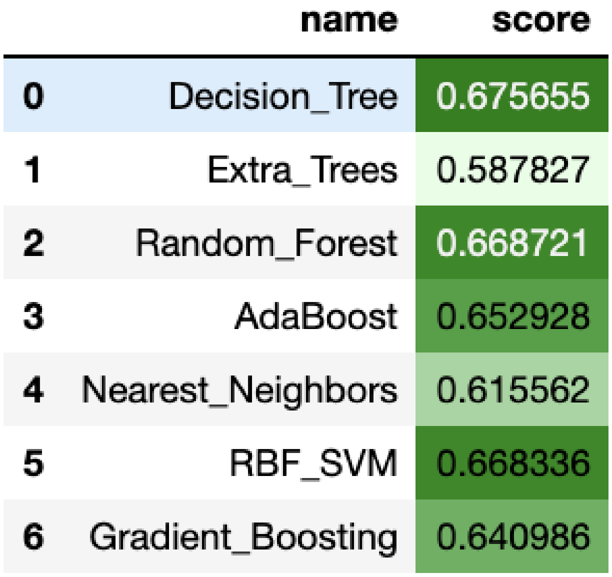
1. Introduction

We used four datasets to answer this question which were provided by the Intel Corporation team – hw_metric_histo.csv, system_sysinfo_unique_normalized.csv, frgnd_backgrnd_apps.csv, and ucsd_apps_exe_class.csv. All four datasets were pre-collected by Intel using Intel's System Usage Report (SUR) collector using their telemetry framework. hw_metric_histo contains information about a laptop's average CPU utilization and temperature. system_sysinfo_unique_normalized contains data on a device's specifications (CPU, GPU, number of cores, etc.) and their predetermined persona provided by Intel (gamer, casual user, office, entertainment, etc.). frgnd_backgrnd_apps.csv provides information on the devices' time spent on certain applications and ucsd_apps_exe_class.csv contains information on the .exe files' application type classification. By combining these datasets, we created a dataframe with the device's specifications, CPU utilization, CPU temperature, and application usage to predict the respective user's persona. To make our predictions, we used multiple scikit-learn classification models. We trained a total of seven different classification models, but ultimately chose to analyze and delve deeper into our radial basis function SVM, AdaBoost, and gradient boosting classification models based on their performance and some interesting shortcomings.

There were a total of eleven personas provided within the dataset: web user, casual user, gamer, casual gamer, office/productivity, content creator/IT, communication, Windows Store application user, entertainment, file & network sharer, and unknown. We believe that some personas such as web users and casual users are very similar in terms of their device specifications; thus, we made an assumption that features such as CPU utilization, CPU temperature, etc. would be similar as well. Because of this, we decided to reduce the 11 different classifications down to four categories. Web users, casual users, communication, Windows Store application users, entertainment, and file & network sharers were categorized as “casual web users” (encoded as 0). Gamers and casual gamers were labeled as “gamers” (encoded as 1). Office/productivity and IT/content creators were condensed into “IT/content creators” (encoded as 2). Lastly, the unknown category remained as “unknown” (encoded as 3). These categories were numerically encoded so that they could be used in a machine learning model. With the encoded target variables and the prepared features from the aforementioned datasets, we were ready to begin training our classification models.

2.1 Methodologies - Initial Model Testing

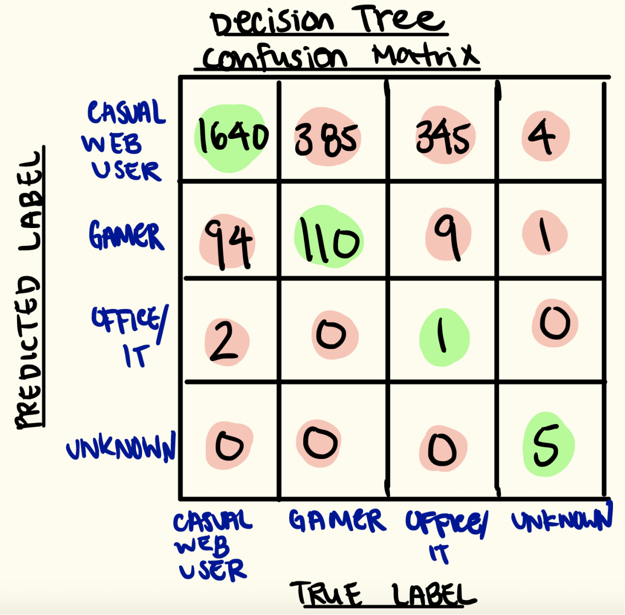
To begin model selection, we ran a for-loop training and tested multiple scikit-learn classification models: decision trees, extra trees, random forest, AdaBoost, three nearest neighbors, radial basis function SVM, and gradient boosting classifiers. The data was split with 80% of the data as the training set and the other 20% as the test set using scikit-learn’s .train_test_split(). Inside the for loop, the models were trained, tested against the test set, and then scored using scikit-learn’s .score() function. The top five performing classification models were decision tree, random forest, radial basis function SVM, AdaBoost, and gradient boosting classification with accuracy scores of 67.67%, 66.87%, 66.83%, 65.29% and 64.10%, respectively (see Figure 1).



	name	score
0	Decision_Tree	0.675655
1	Extra_Trees	0.587827
2	Random_Forest	0.668721
3	AdaBoost	0.652928
4	Nearest_Neighbors	0.615562
5	RBF_SVM	0.668336
6	Gradient_Boosting	0.640986

Figure 1. *Accuracy scores of user’s persona prediction from the seven classification models.*

Albeit the decision tree model performed the best, we realized that it was not doing well in predicting users who were not casual web users. Because a majority of the user personas in our dataset are “casual web users”, the model could classify everyone as a “casual web user” and still achieve a higher accuracy score due to class imbalance. The decision tree, random forest, AdaBoost, and radial basis function SVM all suffered from this flaw. A confusion matrix of our decision tree classifier is shown below to depict the issue of class imbalance (see Figure 2).



Decision Tree Confusion Matrix

PREDICTED LABEL / TRUE LABEL

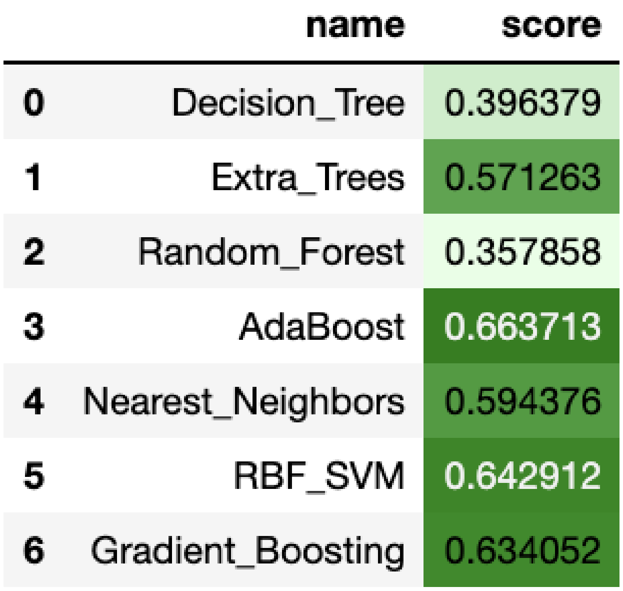
	CASUAL WEB USER	GAMER	OFFICE/ IT	UNKNOWN
CASUAL WEB USER	1640	385	345	4
GAMER	94	110	9	1
OFFICE/ IT	2	0	1	0
UNKNOWN	0	0	0	5

Figure 2. *Decision tree confusion matrix and example of class imbalance.*

From the figure, we can see that it does a good job at classifying the casual users with 94.47% accuracy; however, the model is unable to classify gamers, IT/office users, and unknowns consistently. From this initial run of model testing, we knew that we needed to better handle the class imbalance.

2.2 Methodologies - Model Selection & Class Imbalance Mitigation

To fix class imbalance, we added scikit-learn's class_weight = 'balanced' parameter to the decision tree, extra trees, random forest, and radial basis function SVM. Adding this parameter had large effects on some models. Accounting for class imbalance in the decision tree and random forest models dropped the models' accuracy from 65% to 40% and 64% to 36%, respectively. The accuracy of extra trees and radial basis function SVM did not change much from adding the class_weight = 'balanced' parameter. Initially, the accuracy of the extra trees and radial basis function SVM were 65% and 64%, respectively; however, after accounting for class imbalance the accuracies were 67% and 65%, respectively. The updated accuracy scores are shown below (see Figure 3).



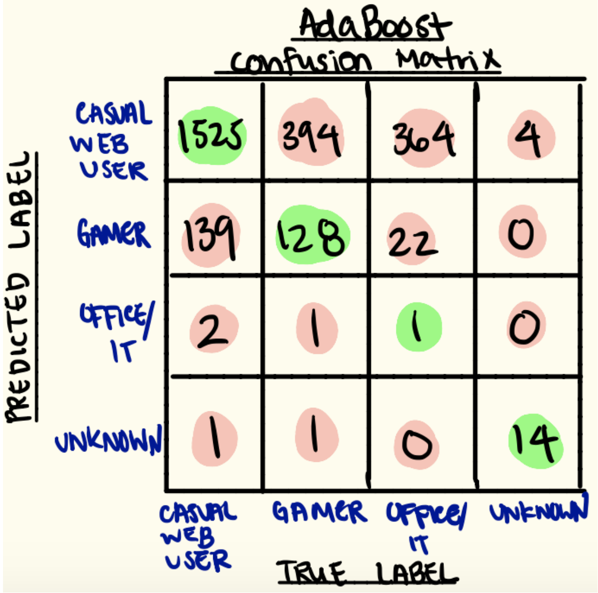
	name	score
0	Decision_Tree	0.396379
1	Extra_Trees	0.571263
2	Random_Forest	0.357858
3	AdaBoost	0.663713
4	Nearest_Neighbors	0.594376
5	RBF_SVM	0.642912
6	Gradient_Boosting	0.634052

Figure 3. *Accuracy scores from the seven classification models after adding class_imblance = 'balanced' to decision tree, extra trees, random forest, and radial basis function SVM classifiers.*

After adding the class_imbalance parameter, our top three models were the AdaBoost, radial basis function SVM, gradient boosting classifiers.

3.1 Results - AdaBoost Classifier

The AdaBoost classifier received an overall accuracy of 66%. The classifier predicted casual web users, gamers, office and productivity users, and unknowns with 91%, 24%, 0%, and 78% accuracy, respectively. The AdaBoost classifier still has a flaw with class imbalance as its stronger performance was based on its strong bias towards classifying most users as a casual web user. The confusion matrix is provided for a more in-depth visualization of its performance (see Figure 4).



AdaBoost
Confusion Matrix

PREDICTED LABEL

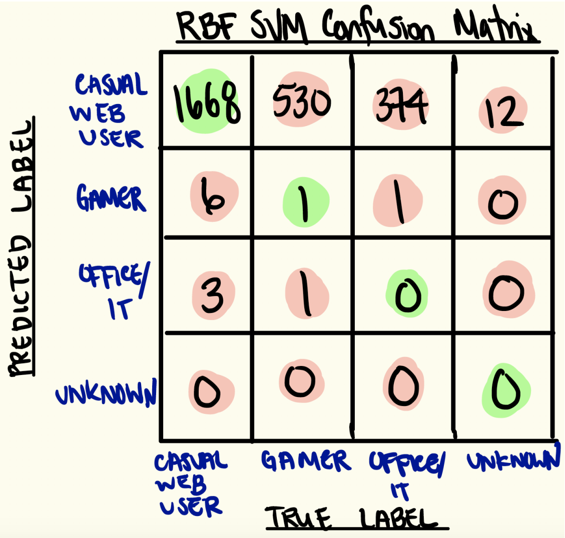
TRUE LABEL

	CASUAL WEB USER	GAMER	OFFICE/IT	UNKNOWN
CASUAL WEB USER	1525	394	364	4
GAMER	139	128	22	0
OFFICE/IT	2	1	1	0
UNKNOWN	1	1	0	14

Figure 4. *AdaBoost classifier confusion matrix.*

3.2 Results - Radial Basis Function SVM Classifier

The SVM classifier received an overall accuracy of 64%. The classifier predicted casual web users, gamers, office and productivity users, and unknowns with 99%, 0%, 0%, and 0% accuracy, respectively. Even though the class_imbalance parameter was set to 'balanced', the model had a very strong bias towards casual web users. The confusion matrix is provided for a more in-depth visualization of its performance (see Figure 5).



RBF SVM Confusion Matrix

PREDICTED LABEL

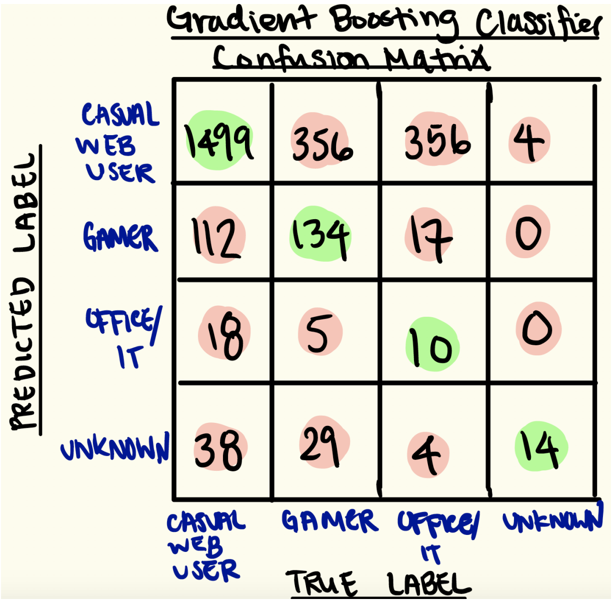
TRUE LABEL

	CASUAL WEB USER	GAMER	OFFICE/IT	UNKNOWN
CASUAL WEB USER	1668	530	374	12
GAMER	6	1	1	0
OFFICE/IT	3	1	0	0
UNKNOWN	0	0	0	0

Figure 5. *Radial basis function SVM classifier confusion matrix.*

3.3 Results - Gradient Boosting Classifier

The gradient boosting classifier received an overall accuracy of 63%. The classifier predicted casual web users, gamers, office and productivity users, and unknowns with 87%, 26%, 3%, and 78% accuracy, respectively. The gradient boosting classifier did the best overall. The model lost some accuracy with the predicting causal web users, but gained some accuracy compared to the other models for predicting gamers, office and productivity users, and unknowns. The confusion matrix is provided for a more in-depth visualization of its performance (see Figure 6).



Gradient Boosting Classifier
Confusion Matrix

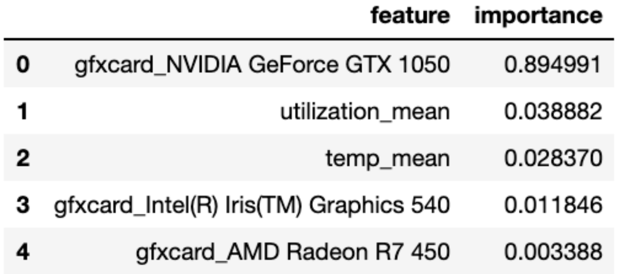
PREDICTED LABEL

TRUE LABEL

	CASUAL WEB USER	GAMER	OFFICE/IT	UNKNOWN
CASUAL WEB USER	1499	356	356	4
GAMER	112	134	17	0
OFFICE/IT	18	5	10	0
UNKNOWN	38	29	4	14

Figure 6. *Gradient boosting classifier confusion matrix.*

Because our gradient boosting classifier had the best overall performance, we decided to examine the most important features for the model. Using scikit-learn's feature_importances_ function, we looked at the model's five most important features. NVIDIA's GeForce GTX 1050 graphics cards, average CPU utilization, average CPU temperature, Iris 540 graphics cards, and AMD Radeon R7 450 graphics were our most important features (see Figure 7).



	feature	importance
0	gfxcard_NVIDIA GeForce GTX 1050	0.894991
1	utilization_mean	0.038882
2	temp_mean	0.028370
3	gfxcard_Intel(R) Iris(TM) Graphics 540	0.011846
4	gfxcard_AMD Radeon R7 450	0.003388

Figure 7. *Top five features and their respective importance calculated by scikit-learn's feature_importance_ function.*

4. Discussion

Our goal was to create a classification model to predict a user's persona based on their device specifications, CPU utilization, CPU temperature, and their time spent on different types of applications. We trained a total of seven models through the scikit-learn package: decision trees, extra trees, random forest, AdaBoost, three nearest neighbors, radial basis function SVM, and gradient boosting classifiers. After training and testing all of the models during our initial run, we realized that our gradient boosting classification model performed the best as it was able to predict casual web users and unknown users with 87% and 78% accuracy, respectively. It predicted gamers and office and productivity users with 26% and 3% accuracy, respectively. Though the accuracy of these two categories were not particularly the greatest, they did a much better job of predicting these categories in comparison to our AdaBoost and radial basis function SVM models. We found that our best model was the gradient boosting classifier which predicts a user's persona with 64% accuracy.

Taking a deeper look into our gradient boosting classifier and its features, we saw that the presence of NVIDIA's GeForce GTX 1050 graphics cards, Iris 540 graphics cards, AMD Radeon R7 450 graphics cards as well as average CPU utilization and average CPU temperature were our most important features in predicting a user's persona.

Acknowledgements

We would like to thank Professor Aaron Fraenkel, Balaji Shankar Balachandran, and Farouk Mokhtar from the UC San Diego Halıcıoğlu Data Science Institute for their help throughout the quarter. We would also like to thank our mentors Jamel Tayeb, Sruti Sahani, Chansik Im, Praveen Polasam, Bijan Arbab, and Julien Sebot from the Intel Corporation for their guidance and feedback during our project.