COSC 6342
Machine Learning

Instructor: Dr. Ricardo Vilalta

A Comparative Study of Machine Learning Classification Models to Predict Device (Computer/Laptop) Configuration

Farzana Yasmin 1877538
Mohammad Imtiaz Nur 1878074

December 09, 2020

1. Introduction:

Due to the increasing immense popularity in the last two decades and robust areas of implementation, Machine Learning (ML) has become the most used topic in recent years. The main goal of ML is to make computer programs capable of learning automatically without human assistance and take decisions for proper adjustment in the actions accordingly. But the general idea of ML is to understand and develop intelligent models that can perceive information from the existing dataset and take action that maximizes the outcome. So developing a project for ML, we have chosen the task of comparing different ML models to get the best accuracy to predict a device (laptop/computer) configuration for different users that meet their requirement in conducting research as well as regular usage.

2. Problem and Related Work:

2.1 Problem Statement:

Since the beginning of the COVID-19 pandemic, the higher education system is going through an unprecedented disturbance as most of the universities restricted in-person activities and mostly went virtual for academic curriculum as well as research. Several studies have been conducted by the universities to understand how higher education and research are being affected and how to address the most severe issues to take immediate actions to cope up with the new normal of working remotely.

One of the biggest concerns of conducting higher education and research activities virtually is to ensure each student has the proper device (laptop, computer, smart devices etc.) with sufficient configuration. The Computational Physiology Lab (CPL) of University of Houston (UH) has conducted a survey in July 2020 among the affiliated members of the university such as students (both undergraduate, graduate), faculty and staffs to know what type of computer and other smart devices they are using for their education as well as research activities.

During the pre-covid time, students can easily access several computer labs, library computers for their respective purposes while after being shifted to the virtual curriculum they became dependent on their personal laptops and computer which in some cases might not be sufficient for their studies. For example, some courses needed substantial programming in MacOs enabled devices which can be easily done at computer labs and classes, but due to remote classes it became a problem for those who have devices of other operating systems. Also the graduate and postgraduate students/researchers have to use comparatively high configuration computers for their research works. As a result, it is quite evident that students and researchers needed devices who were dependent on the university's property for their works. Considering these issues, we have planned to evaluate the dataset and will try to find a proper ML model for our classification problem that has the best capability of predicting the most accurate configuration of laptop or computer especially processor, memory, and harddisk depending on respondent’s department, education label, workload, and several other criterias.

2.2 Background Study:

There have been several comparative analyses of different classification models on different domains and on different datasets. Some of the comparative analysis is listed below:

- **Machine Learning Classification Models with SPD/ED Dataset: Comparative Study of Abstract Versus Full Article Approach [1]:**
  In this paper, authors opted for automating the bibliographic research stage in the biomedical domain. They have compared several classifiers (SVM, Random Forest, Decision Tree, KNN, and Gradient Boosting) of supervised learning that learn and predict a categorical response. They adopted widely accepted performance measures such as accuracy, precision, recall and f1-score to assess the performance of classifiers. Their experiment was done on 300 full paper as well as on 300 abstracts and they showed great comparison among all the models.

- **Comparative Study of Different Classification Techniques: Heart Disease Use Case [2]:**
  In order to evaluate the performance of the classification algorithms, authors of this paper presented a brief comparison of different classification techniques using WEKA. They have explored several algorithms like BN,SVM,ANN,FPT and DT on several parameters such as Kappa statistic, Mean absolute error (MAE), Root mean squared error (RMSE), Relative absolute error (RAE) and Root relative squared error (RRSE). After their experimental analysis and cross-validation for each model they found that SVM outperforms all other ML models.

- **Comparative Study of Different Machine Learning Models for Breast Cancer Diagnosis[3]:**
  The authors of this paper studied the outcome of different classification models like logistic regression, KNN classifier, naïve Bayes, state vector machines, decision trees, random forest classifier on Wisconsin diagnostic breast cancer (WDBC) dataset and measured the performance based on accuracy and confusion matrix obtained. Different types of feature engineering like dimensionality reduction using discriminant analysis and principal component analysis is performed on the features.
- **A Comparative Study on Machine Learning Classification Models for Activity Recognition[4]:**
  Machine learning technique involves generating a model and fitting it with some training data from which it can correlate among the input features to make predictions. From various types of ML models, it is a big concern to find the best model which can predict the accurate output. For different data sets, every model performs differently and can produce significantly different accuracy. The best practice is to compare the accuracy of the different models on the same dataset. The authors have shown the comparative performance of different supervised and unsupervised learning models. After running various classification models, they have shown that the RF model performs the best with a higher accuracy of 99%. They also studied the impact of PCA on the performance of Artificial Neural Network model and SVM.

As we have to predict device configuration (processor, memory, hard disk), we need to explore different classification models of supervised learning such as Logistic Regression (LR), Random Forest(RF), Decision Tree(DT), K-Nearest Neighbour (KNN), Support Vector Machine (SVM) and some ensembling techniques like bagging and stacking.

Logistic Regression (LR):

LR is the simplest classification model which is a predictive algorithm that predicts a dependent categorical variable using independent variables. The dependent variable needs to be categorical whereas the independent variables have the flexibility to be either numerical or categorical. For modeling the conditional probability, LR uses a logistic function whose result always falls between 0 and 1.

The logistic function can be defined as 1/(e−Y + 1), where Y is any number and e is the base of natural logarithms. An S-shaped graph can be formed from this function against Ywhere value will be between 0 to 1. [5]

- The value of the function tends to zero if Y is a large negative number, so e−Y becomes significantly large.

- Similarly, the value of the function equals 1, if Y is a large negative number, so e−Y approaches zero.

Besides having several advantages of being an easy and fast classification method, usability for multiclass classifications; it has some disadvantages like it requires proper selection of features, expects good signal to noise ratio and also outliers & collinearity and outliers alters the accuracy of the model.

Decision Tree (DT):

A tree-based algorithm to solve classification and regression problems named Decision Tree where trees are developed from a set of attributes by taking decisions on a series of tests on those attributes. The tree structure is developed by these tests where every internal node is referred to as decision nodes. For the resulting tree structure, this model continuously splits the dataset depending on the given criteria that ensures the maximum separation of data [6]. Hence, a greedy based recursive algorithm is used to build the tree structure.

Due to the greedy-based process at every step of tree construction, this model has a disadvantage of finding the single best variable with optimal spit-point which causes information loss. Compared to other models it doesn't require any preprocessing of data. It can also handle colinearity much efficiently and the predicted result can be explained with ease by traversing the decision made by the tree. This model has many hyperparameters to tune the model for example - criterion for selecting next node by the cost function, max depth for defining the maximum allowable depth of the tree, and also for the minimum splitting of tree leaf nodes as well sample data.

Random Forest (RF):

A trademark registered by Leo Breiman and Adele Cutler named "Random Forests" was introduced in 2001 [7] where decision trees are randomly constructed and trained on subsets of different data to perform parallel processing. As the RF model is made from multiple decision trees, it is expected to be more accurate, robust to handle overfitting better than other models. To obtain regression and classification outputs, RF model has a set of decision trees ensemble with bagging method where it predicts output by calculating the mean for regression and majority voting for classification.

Besides being more accurate than earlier mentioned models, RF models also efficiently handle overfitting of data and support implicit feature selection to derive important features. Main disadvantage of this feature is being computationally slower and complex for larger forests.

K-Nearest Neighbors (KNN):

The KNN classifier is able to classify the objects by selecting the nearest values for corresponding features. K in KNN indicates the neighbors which are used for classification of a problem [8]. Hence, KNN usually explores neighborhoods and assumes the test neighbour to be similar to them and predicts the result. For finding the best neighbour KNN uses the euclidean distance algorithm.

KNN classifier can provide good results for optimum values of k. As the computations occur in the runtime of a KNN model, it happens to be a lazy learning model. Though it has a very few hyperparameters to tune like k-value and distance function; it will drop performance significantly if the k-value isn't selected properly.

Support Vector Machine (SVM):

SVM is mostly used for regression, classification and outlier recognition, and. It has support for solving both linear and non-linear problems. While SVM with kernels are used for the solutions which aren't linearly separable; the non-kernel version of SVM used for linear problems [9].

In linear SVM, for maximizing the classification margin between two planes, the model derives a new hyperplane. For N number of features in the dataset, the hyperplane needs to be a N-1 dimensional subspace. In the feature space, the boundary nodes are known as support vectors. The maximum margin is derived based on the relative position of the support vectors and in the midpoint of the support vectors a hyperplane can be derived.

In non-linear SVM, for training all the data a kernel function is used to derive the hyperplane Initially, a hyperplane is derived by training the data where the labels are linearly separable. Later, the model classifies the labels in the hyperplane by a linear curve. From this we get a non-linear solution when the classification outcomes are projected back to the feature space. SVM is popular to solve complex problems by it's kernel trick as it uses a convex optimization function, so that a global minima can be achieved always.

Ensemble Learning:

Ensemble method usually combines multiple ML algorithms into one model to predict output. It mainly focuses on decreasing variance (bagging) and bias (boosting) or improving predictions (stacking) [10].

Bagging:

Bagging is a process to reduce the variance of a prediction by averaging multiple estimates together. As an example, we can train N number of different trees on different subsets of the data and compute the ensemble.

Stacking:

Stacking is a technique that combines multiple models (classification or regression) by a meta-regressor or meta-classifier. Based on a complete training set, the base model is computer. Then on the outputs of the base level models, the meta-model is trained. The base level can consist of multiple learning algorithms.

3. Implementation:

3.1. Dataset Description

The dataset is collected from a survey conducted by the computation physiology lab (CPL) of University of Houston (UH). A total of 1574 respondents from different departments have participated in the survey where they were asked about basic informations about their educational stage, affiliated department, ethnicity, primary device for work and it's configuration (i.e. OS, processor, memory, harddisk), nature of their work/research depending on computer use (i.e. Standard User [document writing/email/web browsing] & Power User [programming/analytics, in addition to standard use]), data backup method and frequency, smartphone, smartwatch, accessible internet speed etc. Most of the features were categorical to ensure congruence of data.

3.2. Data Preprocessing

Data cleaning was necessary for our dataset as it was a raw dataset with a lot of null values in several features. After removing the responses consisting of null values we were left with 1088 individual respondent's data.

3.2.1. Exploratory Data Analysis (EDA)

It is always the best practice to perform EDA in a dataset to understand the nature of data, distribution of individual features among the dataset and to get some graphical representation of the data. Here's a plot [Figure 1] of the participants according to their position in the university. In our dataset, it is defined as "participant class". Also, we have plotted them according to their usage of computer (standard user/ power user) in Figure-2.

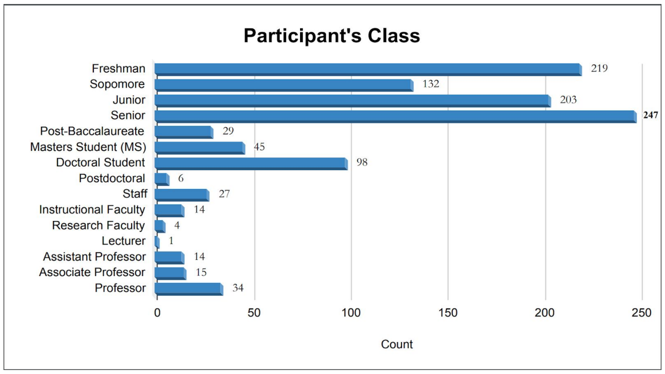

Figure 1

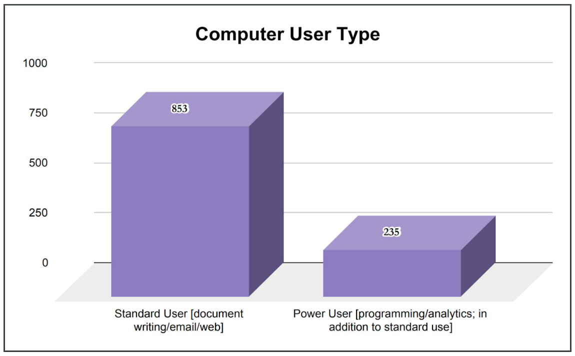

Figure 2

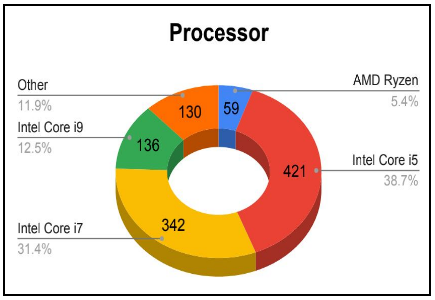

Figure 3

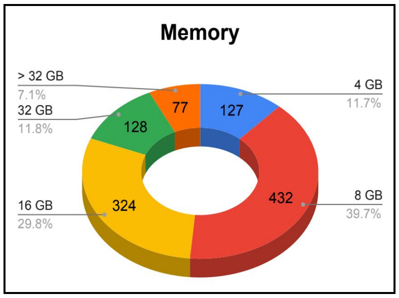

Figure 4

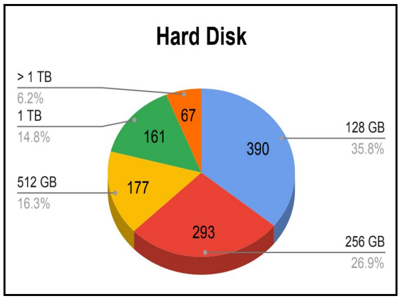

Figure 5

In figure 3, 4, 5 above we have shown the distribution of our output features processor, memory and harddisk respectively according to the participant's response. Clearly, it is evident that the data has some imbalance in the distribution for all the features which might affect our prediction.

3.2.2. Categorical Data Encoding

Most of our data being categorical, we had to convert them to numeric values. While exploring the suitable method for converting the values we considered two methods - label encoding and one-hot encoding.

In the label encoding method, we needed to simply convert the unique values of each column to a numeric value. The only problem with label encoding is that the ML models often consider the numeric values of a column more significant than the other values of the same column according to their values [11]. Though we initially converted all columns value by individual label encoding, we kept these only for those columns (i.e. participant's class, computer user type, processor, memory, harddisk) which have significance in ascending ordered value of the column. Examples are as follows:

We classified the participant class values in Figure-1 above and given then values as the order shown in figure starting from 1 as an increasing sequential order; where the values somehow reflect the more weight is given to the comparatively higher class.

Similarly, we have done that on the columns computer user type (1:standard user, 2: power user), memory (4 GB, 8 GB, 16 GB, 32 GB, > 32 GB) etc.

For all other columns, we used one-hot encoding to get the numeric values. This method converts each category's unique value into a new column and assigns a value 1 or 0 (depending on for true or false) to the column. Our column number increased due to one-hot encoding as we had many categorical values for most of the columns.

It wasn't necessary to normalize any of our data as we had categorical values for each column and no values were outliers compared to the values of the corresponding columns.

3.3 Experimental Analysis

Initially, we developed a correlation matrix to find relations between the features of our dataset and omitted the insignificant features like ethnicity, gender, screen size which shouldn't be considered for predicting a required device configuration of a participant.

We have split our dataset as 80% for training data and 20% for test data and performed parameter tuning on the training dataset. To mitigate the effect of the imbalanced dataset, used stratification of the dependent variables during splitting. So that training and test datasets contain examples of each class in the same proportions as in the original dataset.

The training data then fitted to the ML models. We have used the models Logistic Regression, Decision Tree, Random Forest, KNN, SVM, Bagging and Stacking for each output feature (OS, processor, hard disk, memory) individually.

We performed cross-validation (cv) of the models for each output feature in order to estimate the performance of the model in general when it is used to make predictions on data not used during the training of the model.

For parameter tuning of each model, we used the method grid search by giving a range of hyperparameters to obtain the best result from the model.

We performed a bagging method to reduce variance where we got less accurate output. We also used stacking for each output feature as stacking usually provides better performance than the base models.

3.4. Results and Comparison

While predicting OS, we performed 3-fold cross-validation as one class has samples of 4. That's why we couldn't use folds more than 4. Compared to 4-fold cross-validation, CV=3 showed better results. For other output features (processors, harddisk, memory) we have performed k-fold cross validation where k=10.

Here are the results of our experiment:

OS:

	Logistic Regression	Random Forest	KNN	Decision Tree	SVM	Bagging-Tree	Bagging-KNN	Stacking
Accuracy(%)	**55.63**	49.89	53.68	49.08	54.83	53.68	53.91	51.15
Precision(%)	**50**	51	49	**50**	47	48	47	49
Recall(%)	**55**	52	53	50	51	52	51	53
F1 Score(%)	**52**	51	51	50	49	49	49	51

Processor: CV = 10

	Logistic Regression	Random Forest	KNN	Decision Tree	SVM	Bagging-Tree	Bagging-KNN	Stacking
Accuracy(%)	41.84	42.20	40.23	36.55	40.92	42.18	**42.64**	34.94
Precision(%)	30	**41**	40	38	35	36	39	34
Recall(%)	38	**42**	39	39	41	37	39	42
F1 Score(%)	32	**39**	35	36	33	36	34	37

Hard-disk: CV= 10

	Logistic Regression	Random Forest	KNN	Decision Tree	SVM	Bagging-Tree	Bagging-RF	Bagging-KNN	Stacking
Accuracy(%)	**43.1**	35.75	39.2	35.98	41.84	41.03	40.82	41.15	34.94
Precision(%)	36	41	**42**	40	38	**42**	41	40	40
Recall(%)	41	42	**44**	40	**44**	43	41	41	42
F1 Score(%)	35	41	41	39	37	**42**	40	39	38

Memory: CV = 10

	Logistic Regression	Random Forest	KNN	Decision Tree	SVM	Bagging-Tree	Bagging-RF	Bagging-KNN	Stacking
Accuracy(%)	40.69	35.86	41.26	36.44	**41.84**	40.46	33.94	41.61	37.93
Precision(%)	27	31	30	32	30	**34**	31	32	32
Recall(%)	41	33	34	34	**42**	37	34	35	36
F1 Score(%)	31	31	31	32	31	**35**	31	33	33

Observations:

The overall result is quite unsatisfactory due to the imbalance in the dataset as aforementioned. In most cases, Logistic regression and SVM outperform others.

Bagging is performed for the lowest accuracy holders(weak learners) (decision-tree, RF, KNN). After bagging is done, performance increases than these base models.

Usually, stacking improves performance than the base models. But in our case, stacking didn't increase accuracy as the correct predictions of the base models are strongly correlated. In stacking, we ensembled decision tree, KNN, SVM, and random forest classification models with logistic regression as meta classifier. We could add more models and check if the accuracy increases.

3.5. Conclusion

From the observation of our result, we can see that Logistic regression and SVM have outperformed other models in most of the predictions. SVM has the advantage of getting the global minima and the use of kernel increases the performance of the classification. And logistic regression worked out in the simple case of multiclass classification. Moreover, bagging also improved the performance of the base models as we know ensembling exploits the dependence between the base learners and bagging reduces the variance of the data.

In the future, to make the other non-linear classification models achieve better performance, we need to do further data preprocessing and data engineering like weighting the methods on each class or oversampling the minority class so that the problem of having imbalanced data can be get rid of and the metrics of performance can be improved. We also plan to incorporate more models for performing stacking and check if the performance is better after ensembling for classification. Finally, after observing the performance of the classification models and choosing the best model, we can develop an interface for recommending devices based on user priorities and feature input.

4. References:

1. Khadhraoui M., Bellaaj H., Ben Ammar M., Hamam H., Jmaiel M. (2020) Machine Learning Classification Models with SPD/ED Dataset: Comparative Study of Abstract Versus Full Article Approach. In: Jmaiel M., Mokhtari M., Abdulrazak B., Aloulou H., Kallel S. (eds) The Impact of Digital Technologies on Public Health in Developed and Developing Countries. ICOST 2020. Lecture Notes in Computer Science, vol 12157. Springer, Cham. https://doi.org/10.1007/978-3-030-51517-1_31

2. H. Bouali and J. Akaichi, "Comparative Study of Different Classification Techniques: Heart Disease Use Case," 2014 13th International Conference on Machine Learning and Applications, Detroit, MI, 2014, pp. 482-486, doi: 10.1109/ICMLA.2014.84

3. Kumar A., Poonkodi M. (2019) Comparative Study of Different Machine Learning Models for Breast Cancer Diagnosis. In: Chattopadhyay J., Singh R., Bhattacherjee V. (eds) Innovations in Soft Computing and Information Technology. Springer, Singapore. https://doi.org/10.1007/978-981-13-3185-5_3

4. Nabian, Mohsen. (2017). A Comparative Study on Machine Learning Classification Models for Activity Recognition. Journal of Information Technology & Software Engineering. 07. 10.4172/2165-7866.1000209.

5. Prakash Nadkarni, Chapter 4 - Core Technologies: Machine Learning and Natural Language Processing, Editor(s): Prakash Nadkarni, Clinical Research Computing, Academic Press, 2016, Pages 85-114, ISBN 9780128031308, https://doi.org/10.1016/B978-0-12-803130-8.00004-X.

6. Quinlan, J.R. Induction of decision trees. Mach Learn 1, 81–106 (1986). https://doi.org/10.1007/BF00116251

7. Cutler, A., Cutler, D.R., Stevens, J.R.: Random forests. In: Zhang, C., Ma, Y. (eds.) Ensemble Machine Learning, pp. 157–175. Springer, Boston (2012). https://doi.org/10.1007/978-1-4419-9326-7_5

8. Minakshi Sharma and Sharma Suresh Kumar, "Generalized K-Nearest Neighbour Algorithm-A Predicting Tool", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. 11, November 2013.

9. M. Makinaci, "Support vector machine approach for classification of cancerous prostate regions", World Academy of Science Engineering and Technology, vol. 7, pp. 166-169, 2005.

10. Faliang Huang, Guoqing Xie and Ruliang Xiao, “Research on Ensemble Learning”, International Conference on Artificial Intelligence and Computational Intelligence, 2009 (https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5376633)

11. Categorical encoding using Label-Encoding and One-Hot-Encoder. https://towardsdatascience.com/categorical-encoding-using-label-encoding-and-one-hot-encoder-911ef77fb5bd. [Online, last accessed December 9, 2020]