Automated Retraining of Machine Learning Models

Article *in* International Journal of Innovative Technology and Exploring Engineering · October 2019

DOI: 10.35940/ijitee.L3322.1081219

4 authors, including:

Dr. Krishna Prakasha
Manipal Academy of Higher Education
24 PUBLICATIONS **31** CITATIONS
SEE PROFILE

Vasundhara Acharya
Manipal Academy of Higher Education
19 PUBLICATIONS **37** CITATIONS
SEE PROFILE

Automated Retraining of Machine Learning Models

Akanksha Kavikondala, Vivek Muppalla, Krishna Prakasha K* and Vasundhara Acharya

Abstract*: Data is the most crucial component of a successful ML system. Once a machine learning model is developed, it gets obsolete over time due to presence of new input data being generated every second. In order to keep our predictions accurate we need to find a way to keep our models up to date. Our research work involves finding a mechanism which can retrain the model with new data automatically. This research also involves exploring the possibilities of automating machine learning processes. We started this project by training and testing our model using conventional machine learning methods. The outcome was then compared with the outcome of those experiments conducted using the AutoML methods like TPOT. This helped us in finding an efficient technique to retrain our models. These techniques can be used in areas where people do not deal with the actual working of a ML model but only require the outputs of ML processes.

Keywords: AutoML, Data preprocessing, Feature extraction, Feature selection, Hyperparameter, Machine Learning Optimization, Pipeline, Retraining, TPOT.

I. INTRODUCTION

Traditionally, when one wants to apply a machine learning solution, they have to have knowledge of, various mathematical concepts, programming languages, APIs etc and must also be intuitive enough to know what kind of problem they are trying to solve for example: regression or classification and they should also be able to select appropriate algorithm for their problem to obtain maximum accuracy. One must spend a lot of time to either teach themselves ML concepts sufficiently or take a course on ML in an educational institution. This is obviously going to be very difficult if not impossible for people who are financially weak, people whose careers have little to nothing to do with learning mathematical concepts like non-STEM and non-finance careers etc. They might have some tasks which could be made significantly easier if they could apply ML solutions without having to learn all the concepts. Auto-ML aims to remove the need for a user to constantly get into implementation details like model selection[7], hyperparameter optimization[6],[8] etc. This is obviously an ambitious wish and there are multiple tasks that need to be automated to fully achieve it but our research will be limited to try and find a way to give our ML model the ability to detect inaccuracies in its predictions and decide the degree of inaccuracy at which retraining is necessary and retrain itself on its own.

A. Abbreviations and Acronyms

ML: Machine Learning
AI: Artificial intelligence
STEM: Science, Technology, Engineering &Mathematics
AutoML: Automated Machine Learning
TPOT: Tree Based pipeline optimization tool.
PCA: Principal Component Analysis

II. MACHINE LEARNING AND AUTOML

A. Machine Learning

ML enables the machine or a system to learn and improve its performance with experience, without being explicitly programmed. It is an application of AI. ML can be thought of as a human that tends to classify something on its own [23]. There are few types of ML. These include supervised machine learning, unsupervised machine learning, semi-supervised machine learning and reinforcement learning . Our research deals with supervised machine learning where the machine is trained using a labeled dataset. ML involves processes such data gathering, data preparation, selection of an algorithm, training, evaluation, parameter tuning and prediction. Data preparation takes about 90 percent of time in the data pipeline. ML can hence be time consuming and becomes expensive [1]. In order to automate the data preparation process, AutoML had been introduced.

B. AutoML

AutoML automates the machine learning pipeline and makes the preprocessing of data, feature extraction and feature engineering processes simpler without any human intervention. In theory, AutoML removes the burden of creating a model from the user. AutoML chooses the optimal values of hyperparameter themselves. Many cloud platforms like AWS, Google cloud, H2O etc provide users with AutoML implementation. Our research work involves using the TPOT package in python to automate and extend our re-training process [17],[22].

Revised Manuscript Received on October 05, 2019.
***** Correspondence Author
Akanksha Kavikondala, Department of Information and Communication Technology, Manipal Institute of Technology (Manipal Academy of Higher Education), Manipal, India. Email: akanksha.kavikondala@gmail.com
Vivek Muppalla, Department of Information and Communication Technology, Manipal Institute of Technology (Manipal Academy of Higher Education), Manipal, India. Email: vivekray174@gmail.com
Krishna Prakasha K*, Department of Information and Communication Technology, Manipal Institute of Technology (Manipal Academy of Higher Education), Manipal, India. Email: kkp.prakash@manipal.edu
Vasundhara Acharya, Department of Computer Science & Engineering Manipal Institute of Technology (Manipal Academy of Higher Education), Manipal, India. Email: vasundhara.acharya@@manipal.edu

C. TPOT

TPOT is an AutoML method and open-source software package developed. TPOT was developed by Dr. Randal Olson with Dr. Json H. Moore at the Computational Genetics Laboratory of the University of Pennsylvania and is still being extended and supported by this team [22], [20].
The goal of TPOT is to automate the building of ML

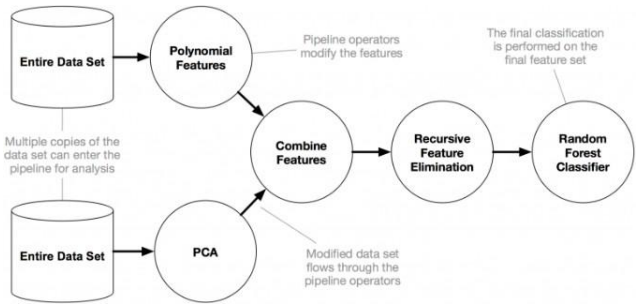

Fig 1: Working of TPOT

pipelines by combining a flexible expression tree representation of pipelines with stochastic search algorithms such as genetic programming. TPOT makes use of the Python-based scikit-learn library as its ML menu[10], [11]. Fig 1 shows the working of a TPOT module.

D. Advantages and Disadvantages

The key advantage of AutoML over ML is that one does not concern themselves with what is happening inside a ML process. A non expert can easily perform ML operation without any prior knowledge about ML. A good set of results can be obtained by AutoML than ML. However the cost of training in AutoML is higher than training in AutoML[22].

III. PROBLEM DEFINITION

Retraining machine learning models every time they become obsolete is time consuming. Especially when the application is complex and the training and evaluation datasets are large. There is a need to give the model capability to detect change in nature of data and retrain itself. We want to demonstrate a model performing this with the use of a classification problem. On music hosting sites like Spotify, there is a need to price advertisements differently to attract maximum number of advertisers. This can be done by pricing advertisements on popular songs of popular or trending artists higher than the not-so-popular ones. Our model should be able to detect periodic changes in popularity of various artists and classify them into categories for variable pricing.

IV. OBJECTIVE

Our main objective is to build a model that can retrain itself appropriately. To achieve this, we need to:
1. Design an algorithm that can identify and detect a threshold when the model becomes inaccurate.
2. Automate the algorithm and integrate an upgraded algorithm into our model.

V. METHODOLOGY

Our methodology in brief involves the following:

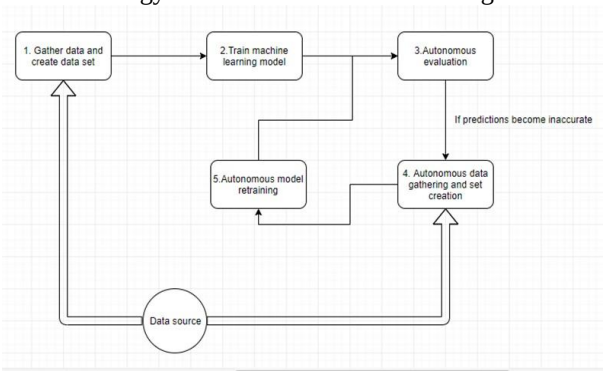

Fig 2: Flowchart of the project

A suitable dataset would be chosen in order to perform this experiment.
1. The model then would be evaluated in terms of its accuracy to predict the result. This forms the first part of the project.
2. The second part involves changing the data input values and testing the data with the existing model. If there is no change in the prediction, the model is retained else retrained.
To achieve our objectives, we have made use of Python and various ML libraries available for Python. Initially tested our algorithms out on single feature regression and classification problems, then extended it to the Spotify advertising problem. We have attempted to create a model that can dynamically draw data from the internet so we can know the popularity of songs in real time.

A. Extraction of data

We first created a developers account on Spotify and generated client id and secret key to access the data as a Spotify developer. Python provides the users with a Spotipy package which connects to Spotify. Using the credentials to create a spotipy object, we extracted the data using the function calls search and audio features. The search function extracted the song name, popularity of the song, artist of the song and the album to which it belonged. Using the track id, which was one of the attributes of the results returned by the search function, as a parameter[32] in audio features, we extracted the audio features like danceability, tempo, key, mode etc[5].

```
import spotipy
from spotipy.oauth2 import SpotifyClientCredentials
import pandas as pd
import time
#import ssl
#import certifi

#certifi.where()

#ssl._create_default_https_context = ssl._create_unverified_context


cid="xxxxxx"
secret= "xxxxxxxxx"

client_credentials_manager = SpotifyClientCredentials(client_id=cid, client_secret=secret)
sp = spotipy.Spotify(client_credentials_manager=client_credentials_manager)

artist_name = []
track_name = []
track_id = []
popularity = []

for i in range(1, 200, 50):
    track_results = sp.search(q='year:2019', type='track', limit=50, offset=i)
    for i, t1 in enumerate(track_results['tracks']['items']):
        # print(t)
        artist_name.append(t1['artists'][0]['name'])
        track_name.append(t1['name'])
        track_id.append(t1['id'])
        popularity.append(t1['popularity'])
```

Fig 3:Snapshot of pseudocode

B. Data Preprocessing

We first described the dataset in order to get a picture of what data we would be working on [3]. We searched for any missing values[28], [29]. Our data has no missing values. We dropped the attributes such as 'Artist Name', 'Track ID', 'Track Name' and 'URI',that were not in float or int format. Just like missing values, our data might also contain values that diverge heavily from the big majority of the other data. These data points are called outliers. To find them, we can check the distribution of our single variables by means of a box plot or we can make a scatter plot of our data to identify data points that don't lie in the expected area of the plot [28], [29]. We visualize the data using plots. We visualize the data using plots. This helps us know what the attributes wish to convey about the output variable.

Authentication snippet:

import spotipy

from spotipy.oauth2 **import** SpotifyClientCredentials

cid ="xx "

secret = "xx "

client credentialsmanager = SpotifyClientCredentials(clientid = cid.clientsecret=secret)

sp = spotipy.Spotify(clientcredentialsmanager = clientcredentials manager)

All this information was saved onto a dataframe and then written onto a csv file for further work. We extracted data from the years 2015-2019. This data would serve as our training data.

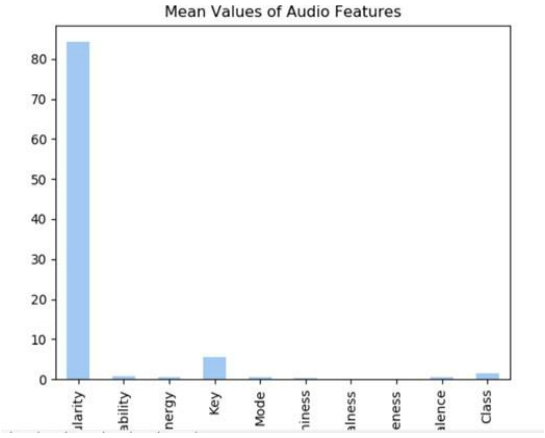

Fig 4: Mean of each audio feature to know the exact mean value of each

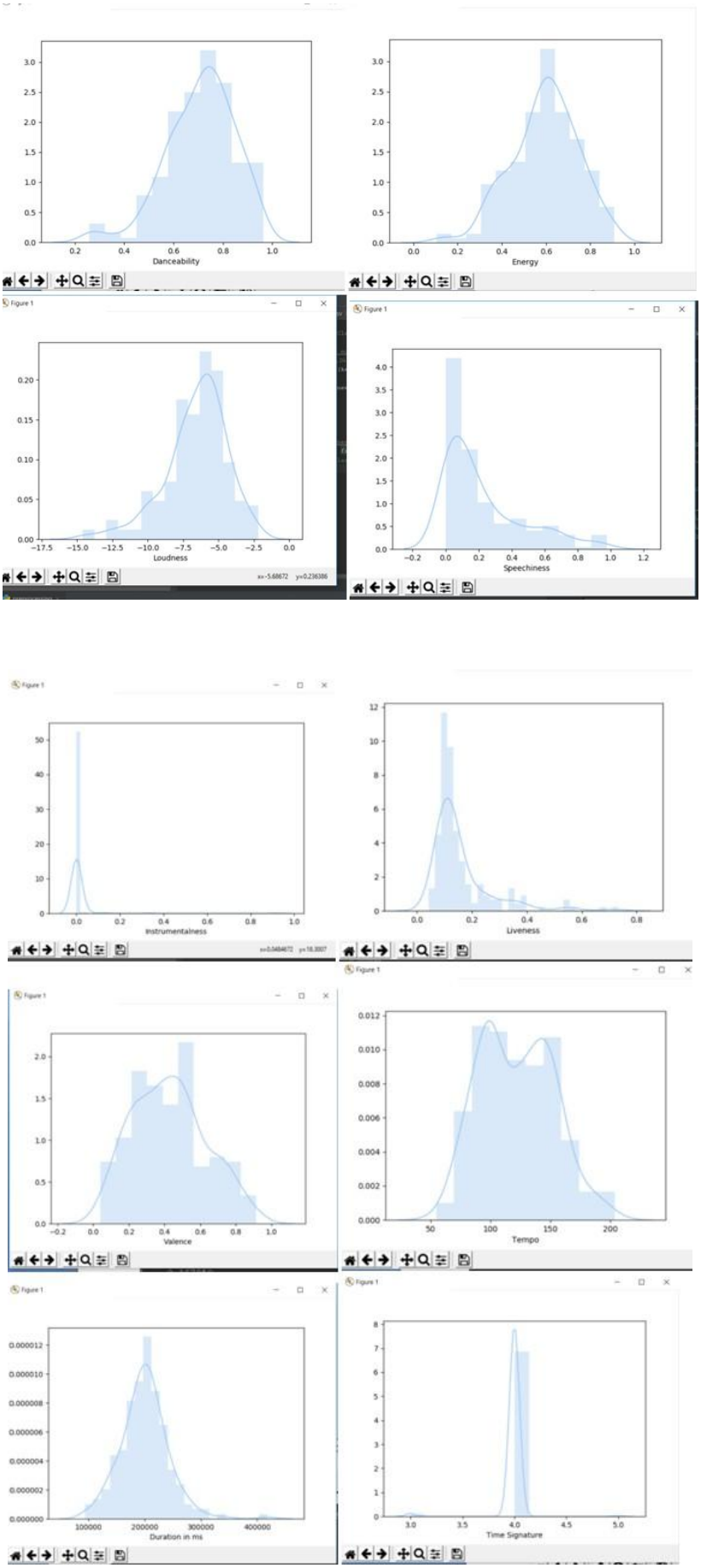

Fig 5: Distribution of all attributes

Key (attribute) values were present in the form of numbers which had to be mapped to their equivalent nodes like C, C major, D etc. The keys were mapped using the map function.

C. Training

We split the preprocessed data into training and testing sets. Training set constituted about 75 percent of the data while 25 percent of the data was present in the test set. Training set was used to derive a model while the test set was used to evaluate the model.

We compared all the regression models like linear regression, ridge regression and random forest regression in order to predict the popularity of a song. The algorithm showing the best R2 value was chosen. However for our data, all the models showed an R2 value of 0.04 hence we chose the linear regression model as it is the simplest of all. For classification we compared various algorithms like logistic regression, SVM, Random Forest, KNN etc in terms of accuracy to determine which algorithm would best suit our data. Model with higher accuracy was chosen. We performed K-fold cross validation in order to determine our algorithm.

We can try classifying without having to perform regression as well. Following is the alternative to what we have done above: Instead of performing regression and then classification, we tried to predict the classes using all the audio features and popularity. Since it is a 14-dimensional array, we reduced the dimensions of the array such that it would be easier for the fit, predict and score functions to perform their job. Using PCA, we reduced the dimension of a 14D array to a 2D array and passed it as input to our model.

We observe that there is no difference in the accuracy predicted using both the methods.

In one of the papers, handling similar kind of data and problem statement, it is mentioned that direct classification of data without performing regression can lead to loss in some important information and hence can affect accuracy [13]. In our case the accuracy remained the same using both the methods.

D. Testing

Once the model was trained, we evaluated the performance of the model using the test set. For linear regression, we used the mean square error and R2 scores to determine performance, while we used accuracy, precision and recall as scores for classification algorithm.[24] If the performance of the model was good enough for the test data, we saved the models onto two pickle files,[26] one for regression and one for classification using pickle module of python. This model will now be used on new incoming data.

E. Retraining

We have used python scheduler in order to extract data weekly. We use the saved models onto the data. If we find any change in the distribution or the accuracy scores of the model, we retrain it [14],[15], [19], [21].

We combine the data that we used to model previously and the new data that we extracted in order to train our model. If we do not detect any significant change in the accuracy or if the distribution [30] of the incoming data is similar to that of the data that was used to train the model, we do not retrain the model. For retraining we use an AutoML package called TPOT. TPOT selects the appropriate the algorithm for the dataset, tunes the hyperparameters and produces efficient and

Similarly a value of 0 of the mode attribute had to be mapped to ‘minor’ while a value of 1 had to be mapped to ‘major’.

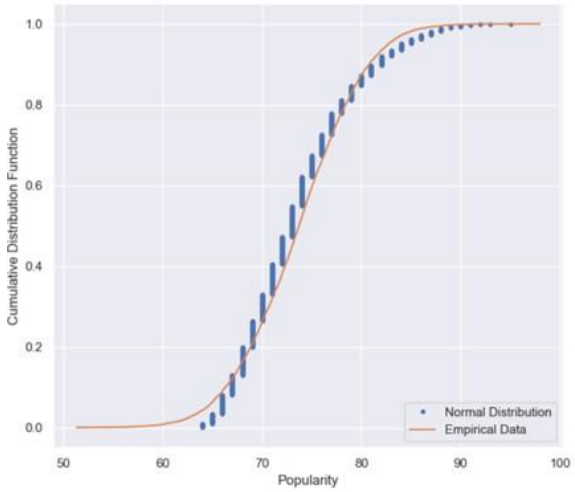

Fig 6: Distribution of the data w.r.t Normal distributionfaster results.

VI. RESULT AND DISCUSSION

The first part of our project required us to train and test our model using the ML library Scikit-learn[31]. Using traditional ML libraries gave us an idea of how the user needs to check for inconsistencies in data either using plots or mathematical functions. The feature selection and the model selection was done by us. The number of models from which we chose our best model, based on the accuracy score, was very limited due to our limited knowledge of models. We randomly selected the hyperparameter and didn’t tune it as we got the result we expected. However if we hadn’t got expected results, we would have had to tune our hyperparameters, giving it random values each time we wanted it to perform better.

Using TPOT, an AutoML library for retraining and traditional ML library for initial training has given us sufficient reasons and conditions in which either of these libraries could be used. If the data is small and not complex, such that data preprocessing and model selection can be easily done by the user, then traditional ML libraries could be used. It is used when the number of resources to perform highly complex calculations in limited.

On the other hand, when the data becomes complex and large such that data pre processing, model selection and hyper parameter tuning become a burden, then AutoML libraries could be used. However it must be ensured that the system one is working on can support highly complex calculations.

Limitations of using TPOT:

However efficient TPOT module might be, one does need to clean the data before sending it as an input to the TPOT classifier or regressor object. For example it does not accept missing values.

TPOT can take a lot of time when it comes to handling large datasets. The module is under constant development and hence can also lead to importing of future warnings. TPOT tends to slow down as the number of

data points increase. It consumes lot of resources and hence can't be used on basic laptops for large data.

Advantages of using TPOT: Despite cleaning the data, all other data preprocessing steps such feature selection, feature engineering etc., algorithm selection and hyperparameter tuning are performed by the module itself. It gives the ba- sic possible pipeline for our dataset. It chooses from a range of algorithms that the user could be unaware of.

Although TPOT takes in lot of resources, it selects the most appropriate algorithm in minimum amount of time. The burden of selecting the right algorithm and tune the hyperparameters is taken off our shoulders.

The parameters generations, population size and verbosity in TPOT regressor and classifier functions depict the number of iterations to run the pipeline optimization process, number of individuals to retain in the Gaussian process population every generation and the amount of information communicated by TPOT, respectively. When verbosity is set to 2, it shows the optimization procedure through a progress bar as seen above. Since the number of generations set by us is 5, it performs the optimization process in 5 iterations. It shows the best score for that pipeline in terms of negated mean squared error(for regression) and accuracy(for classification). The scoring function can be changed as per the need of the user and type of data. The scoring function needs to be mentioned while creating the object. TPOT will evaluate population size + generations x offspring size pipelines in total. Offspring size is the number of offsprings produced in each generation. By default it is set to population size. TPOT (Fig 14) chooses ExtraTreeRegressor with a cv score of -13.51(negated mean squared error) as the best algorithm for regression and ExtraTreeClassi- fier with a cv score of 1.0 (accuracy) for classification. It can also be seen that the models are selected with specified hyperparameters. Fig 7 shows the TPOT output for our data. We also compared the training scores of the models chosen by us using our interpretation of the data and the model chosen by AutoML.

Fig 7: TPOT output

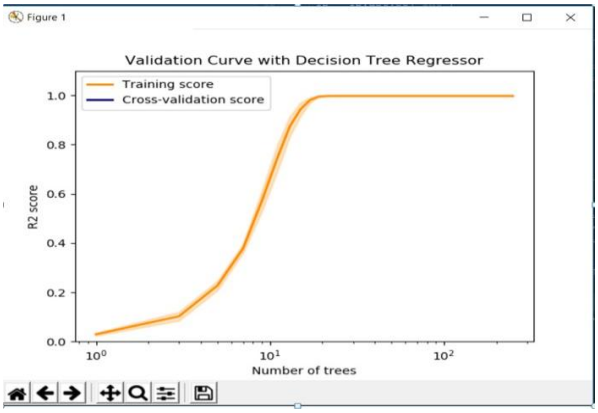

Fig 8: Training score of model chosen by TPOT

The scoring method used here is R2. As we can see, the R2 is almost 0 when it comes to Linear Regression in Fig 9, a model chosen by us. However the model chosen by TPOT gives a better score by tuning its hyperparameters in Fig 8. X-axis depicts the value of the hyperparameter (in this case, number of trees).

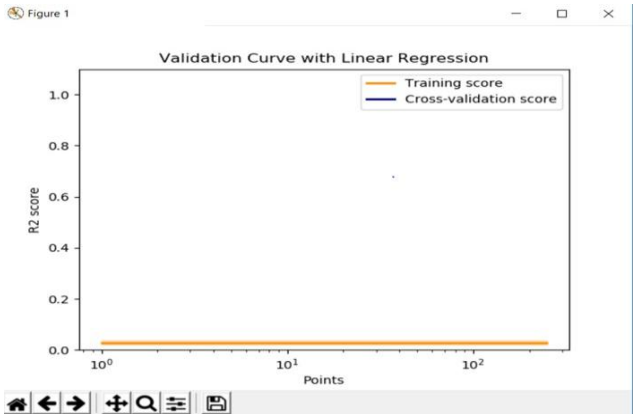

Fig 9: Training score of model chosen by us

We also tried to plot the time taken by our AutoML code and traditional ML code with respect to the number of iterations as shown in Fig 10. For traditional ML, processing and training is not done in one go. One has to go through each plot produced, analyze the output of every feature, select the feature and then training. The process of training also requires the user to experiment with different hyperparameter values. Hence it may take days to come up with an ideal algorithm. The plot for traditional ML is on the algorithm that we have selected for our data by performing all the required steps beforehand.

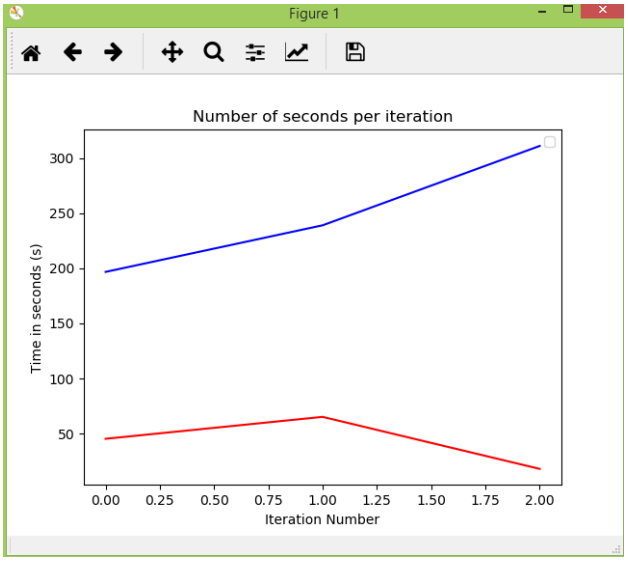

Fig 10: Plot of Traditional ML (red line) and TPOT (blue line) taken for a different dataset

So what library does one choose for retraining? It depends on the kind of application being developed and the compatibility of that library with different OS. We chose TPOT as it was Windows compatible and it could work with our data. There are many AutoML libraries that could be used and are efficient. One can also use traditional ML libraries for retraining, however AutoML does make the process of retraining easier as the user does not involve himself with selecting an algorithm especially when the user has limited knowledge of the models existing.

VII. CONCLUSION

The research started off by using traditional ML process. The hyperparameter tuning and selection of the model was manually done by us. Graphically, AutoML process took more time than the traditional ML process. However these graphs did not take the human intervention into account. The time a process takes depends on the type and size of a dataset. AutoML processes may take longer than traditional ML processes or may even take less time compared to traditional ML processes. However AutoML chose a more accurate model than what we chose using traditional ML process. Hence AutoML improves the accuracy of results and chooses the most appropriate model for the data. It also removes human intervention that is normally required in traditional ML processes.

This research was just an attempt to see how we could integrate the idea of retraining with AutoML. Retraining is essential in cases where the data and the nature of data changes frequently. Retraining can help the model be up to date and not become obsolete. There are many web services such as AWS, Azure [14], [15], [16], [27] etc that provide retraining of models as a service. AutoML packages are constantly under development and hence can't be used very often. However they reduce the burden of selecting algorithm and tuning hyperparameters. It becomes useful especially in case of retraining, where the users may not always be able to select the right type of algorithm for their datasets.

This research contributes just a little bit to the overall objective of the field of AutoML. Our work can empower the common man to use ML as a tool. By removing the burden of retraining and selecting algorithms from the user, we can increase the pool of people who can use ML in their daily lives, especially the ones in non-STEM fields.

We have used only accuracy as the metrics to evaluate our model when retraining. However accuracy alone cannot determine how well our model can classify the dataset. Any future work may involve using other metrics such as precision, recall, and f1. The reason why we didn't choose to use other metrics as measures was because we would have had to create new TPOT classifier objects every time we required scoring the model using different metrics. It would be time consuming and inefficient to create new objects for all the metrics present. Accuracy is the default metric when creating a TPOT classifier object. However, one can use other metrics in order to compare and create a model. Any further improvements shall include using all the metrics to capture how well the model does the classification.

REFERENCES

1. Andrew NG. "CS229 Lecture Notes, Stanford" [Online]. Available: http://cs229.stanford.edu/notes/cs229-notes1.pdf
2. Dumitru Iulian Nastac and Paul Dan Cristea, "A Modified Adaptive Retraining Procedure for Data Forecasting", IEEE 2012
3. Isabelle Guyon, Andre Elisseeff. "An introduction to variable and feature selection", http://www.jmlr.org/papers/volume3/guyon03a/guyon03a.pdf
4. Nastac, Iulian. (2004). "An Adaptive Retraining Technique to Predict the Critical Process Variables", June 2004.
5. James Pham, Edric Kyauk and Edwin Park,"Predicting Song Popularity" [Online]. Available: http://cs229.stanford.edu/proj2015/140_report.pdf
6. Matthias Feurer,Aaron Klein,Katharina Eggensperger, Jost Tobias Sprin- genberg, Manuel Blum and Frank Hutter, "Efficient and Robust Automated Machine Learning", NIPS, 2015
7. Pavel Brazdil and Christophe Giraud-Carrier, "Metalearning and Algorithm Selection: progress, state of the art and introduction to the 2018 Special Issue Cross Mark", 2018.
8. J. Bergstra, D. Yamins and D. D. Cox, "Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures", 2013
9. Lars Kottho, Chris Thornton, Holger H. Hoos, Frank Hutter and KevinLeyton-Brown, "Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA", JMLR, 2017
10. Randon S Olson and Ryan J. Urbanowicz, "Automating Biomedical Data Science Through Tree Based Pipeline Optimization", Springer 2016
11. Randal S. Olson, Nathan Bartley, Ryan J. Urbanowicz, and Jason H. Moore (2016), "Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science".
12. Matthias Feurer, Katharina Eggensperger, Stefan Falkner, Marius Lindauer and Frank Hutter, "Practical Automated Machine Learning for the AutoML Challenge 2018"
13. Jun Lu, "Machine learning modeling for time series problem: Predicting flight ticket prices", 2017
14. Retrain an Azure Machine Learning Studio Model [Online], Available: https://docs.microsoft.com/en-us/azure/machine-learning/studio/retrain-machine-learning-model
15. Retraining and updating Azure Machine Learning models with Azure Data [Online], Available: https://azure.microsoft.com/es-es/blog/retraining-and-updating-azure-machine-learning-models-with-azure-data-factory/
16. Automated Machine Learning, Wikipedia[Online]. Available:
17. http://en.wikipedia.org/wiki/Automated machine learning

18. Automated Machine Learning, Data Robot [Online]. Available: https://www.datarobot.com/wiki/automated-machine-learning/
19. Build, Deploy and Retrain a Predictive Machine Learning [Online]. Available: https://cloud.ibm.com/docs/tutorials?topic=solution-tutorials-create-deploy-retrain-machine-learning-model
20. TPOT Automated Learning in Python [Online]. Available: https://towardsdatascience.com/tpot-automated-machine-learning-in-python-4c063b3e5de9
21. Keeping your Machine Learning Models up to date, using IBM Watson Data Lab[Online]. Available: https://medium.com/ibm-watson-data-lab/keeping-your-machine-learning-models-up-to-date-f1ead546591b
22. 21.Automated Machine Learning, AutoML.org [Online]. Available: https://www.automl.org/automl/
23. Machine Learning Definition from Expert System[Online]. Available:
24. https://www.expertsystem.com/machine-learning-definition/
25. Google Crash course on Machine Learning[Online]. Available: http://www.developers.google.com/machine-learning/crash-course/classification/accuracy
26. Machine Learning Mastery, Python step by step [Online]. Available: https://machinelearningmastery.com/machine-learning-in-python-step-by-step/
27. Python Programming, Serializing using Pickle [Online]. Available: https://www.pythonprogramming.net/python-pickle-module-save-objects-serialization/
28. Amazon Web Services Documentation[Online]. Available: https://docs.aws.amazon.com/machine-learning/latest/dg/when-to-use-machine-learning.html
29. Data Camp, Feature Selection Python [Online]. Available: https://www.datacamp.com/community/tutorials/feature-selection-python
30. Data Camp, Exploratory Data Analysis Python 30. Statistics Solution, Mann Whitney U Test [Online]. Available: https://www.statisticssolutions.com/Mann-Whitney-U-Test
31. Lars Buitinck *et al*. ECML PKDD Workshop: Languages for Data Mining and Machine Learning 2013, Scikit Documentation.
32. Tomi Gelo, "Spotify Data Project Part 1 - from Data Retrieval to First Insights"[Online]. Available: https://tgel0.github.io/blog/spotify-data-project-part-1-from-data-retrieval-to-first-insights/

AUTHORS PROFILE

Akanksha Kavikondala is a first semester student in National University of Singapore, School of Computing doing her masters in Information Systems. She graduated from Manipal Institute of Technology, MAHE with a degree in Bachelor of Technology in Computer and Communication Engineering in 2019.

Vivek Muppalla is currently working as a software engineer at Larsen and Toubro Technology Services (LTTS), Bangalore, India. He graduated from Manipal Institute of Technology, MAHE with a degree in Bachelor of Technology in Computer and Communication Engineering in 2019.

Krishna Prakasha K received the B.E.,M.Tech. degree from Viswesvaraya Technological University, Belagavi. and Ph.D. degree in network security from MAHE, Manipal..He is associated with the Department of Information and Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, where he is currently an Assistant Professor (Senior) . He has more than 25 publications in national and international conferences/journals. His research interests include information security, network security, algorithms, real time systems, and wireless sensor networks.

Vasundhara Acharya Vasundhara Acharya received the B.E. degree in information science and engineering from the N.M.A.M. Institute of Technology, Nitte, and the M.Tech. degree in software engineering from the Manipal Institute of Technology (MIT), Manipal Academy of Higher Education (MAHE), Manipal. She plans to pursue her Ph.D in Bioinformatics. She is currently an Assistant Professor with the Department of Computer Science and Engineering, MIT, MAHE. Her current interests include information security, medical image processing, and artificial intelligence. She has been serving as reviewer for various International Journals. She is the Review Board Member of the International Journal of GEOMATE, Japan, and an Active Reviewer of Medical & Biological Engineering & Computing. She is serving as an Editorial Board Member for Information and Computer Security, Enpress publisher.