Universität Siegen

OpenML Regression Bot: Streamlining Regression Analysis with Automated Model Selection and Evaluation

By **Perugu, Pavan Kumar, 1599044**
Under the supervision of **Prof. Dr.-Ing. Joeran Beel**
31. March 2023

www.uni-siegen.de

https://github.com/PavanKumarPerugu/OpenML_Regression_Bot--OpenMLRB

Table of Contents

Abstract

The present study introduces an Open Regressor Bot designed to automate regression tasks on the OpenML platform, which previously lacked in automated model selection for regression tasks. The Bot uses an AutoML technique to run on almost every OpenML regression task and obtain required metrics with a set of specified hyperparameters. Users can store and upload results on the local disk and OpenML platform, respectively. The study highlights the potential of the Open Regressor Bot in overcoming the limitations of OpenML Regression Tasks Automated Model Selection and making the running of Regression Tasks on OpenML more convenient and user-friendly.

Key Words: ML, OpenML, ISG, Open Regression Bot, SHp, AutoML, OpMLRB, OpMLRTAMS, OpenML Automation

Glossary: ML - Machine Learning, ISG - Intelligence Systems Group, OpMLRB - OpenML Regression Bot, SHp - Specified Hyperparameters, OpMLRTAMS - OpenML Regression Tasks Automated Model Selection, RTs - Regression Tasks, MP - Model Performance, SM - Specified Metrics, THp - Tuned Hyperparameters, MAE - Mean_Absolute_Error, MSE - Mean_Squared_Error, AMS – Automated Model Selection.

1

Introduction

1.1. Background: OpenML is an open-source ML platform that provides access to thousands of datasets, enabling users to create ML model pipelines and evaluate their model performance against specified metrics. It is also an excellent tool for showcasing ML skills and has been utilized by professionals in the field. In this regard a team of people developed Sklearn-Bot motivated by automating ML and Parameter Importance in Classification Tasks (CTs) on OpenML. It achieves this by specifying Tuned Hyperparameters (THp) within a standard range of performance metrics to make CTs more accessible in OpenML analysis.

1.2. Research Problem: Despite the popularity of RTs, no such comfortability in AMS for RTs analysis on OpenML has been developed until now. No one addressed this area for all these years.

1.3. Research Goal / Objective / Hypothesis / Benefits: The ISG proposed creating an OpMLRB for OpenML Regression Tasks. This bot takes the ML Regression Algorithm name and Regression Task ID of OpenML as input through argparse,

and applies a set of tuned and configured Hyperparameters based on Algorithm and Task ID selection for better performance.

1.4. Research Tasks: The development of the entire bot occurred in three sub-tasks.

i. The first sub-task involved the development of the Run_on_Task(OpMLRB) module**.**

ii. The second sub-task involved the Automated ML codes Package, which was used to create the Configuration Spaces module that defined SHp for various ML algorithms.

iii. The third sub-task is involved with several sub-functions, such as parse_args(), pipeline() and del_outputdir(), which are called by the Run_on_Task module. This module was designed to build pipelines and enable them running on OpenML Tasks.

1.5. Contribution: Development the 3 complete individual modules with an unique functionality for each one. This is for the easiness in understanding the architecture to develop further in future. Furthermore, making sure that the results are being stored in local drive and to the OpenML Server at the same time.

2

Background

In ML, the performance of the prediction model depends on several factors such as mentioned beside. Hence to make model prediction more accurate professionals in the field of ML have to develop a much better pipeline with which they can do appropriate data pre-processing and so the less computation requirement will be needed as same as time to train the model in achieving it’s best performance.

Pipeline

2.1. Data Collection,
2.2. Data Preparation,
2.3. Feature Selection,
2.4. Model Selection,
2.5. Training the Model,
2.6. Model Evaluation,
2.7. Model Optimization,
2.8. Model Deployment,
2.9.Model Maintenance.

3

Related Work

In this process of automating ML a team of researches developed a Bot with which they can do the Prediction Model analysis of any classification task to get aware of the Hyper parameter Importance by tuning them in achieving more accurate model predictions. Finally, this way every ML algorithm will have certain final Hyper parameters(according to AutoML Project) with which one can achieve the best prediction model with less computation effort and time as well. However, with help of this Bot professionals on OpenML(ML Platform where one can build a potential portfolio and showcase their achievements in the respective field) discovered a new usage of it over time that the Sk-learn Bot made an easy path way to run classification tasks in no time by an autoML technique called Automated Model Selection for their further research improvements for the close observation of other factors by standardising some.

4

Methodology:

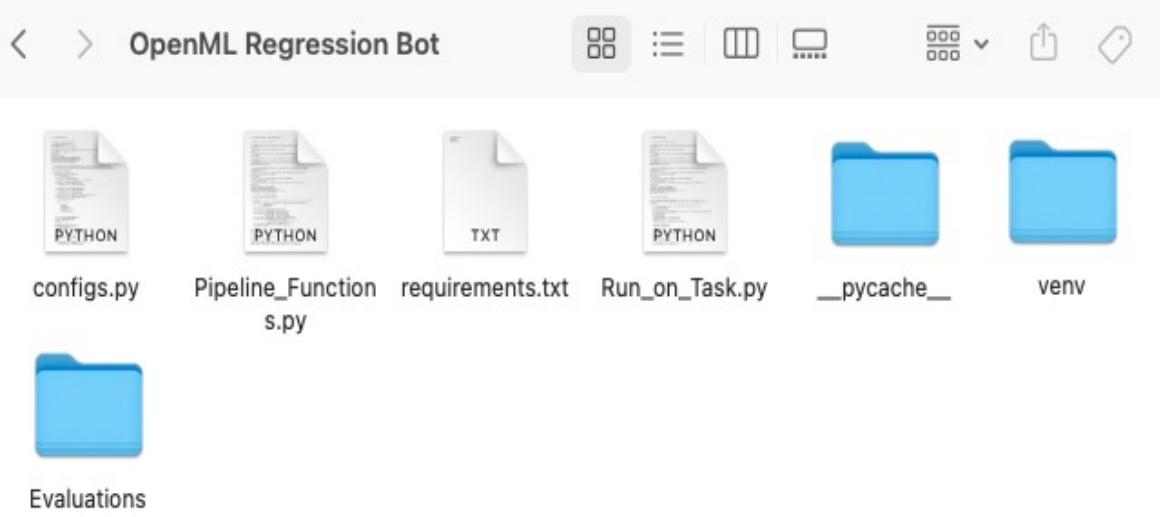

Fig. 1. OpenML Regression Bot Directory where the results have been stored

With the help of existing literature (ref. to Chapter No. 3), we built a pipeline which takes Task ID and Regression Algorithm names as inputs and run on on an OpenML RT with a specific Task ID to analyse the model performance based on certain Metrics (such as Mean_Absolute_Error (MAE), Mean_Squared_Error (MSE)) .

The complete code has been segregated into 3 distinct Modules, each serving a specific purpose, in order to facilitate comprehension of the Bot's overall functionality and to enable future enhancements with ease. The 3-Optimal Modules with unique functionalities as follows:

4.1. Run_on_Task / First Module: the initial module of a Bot, Run_on_Task.py, which receives configurational parameters and executes the Run_on_Task function to obtain the model's performance metrics. The process consists of two stages: importing necessary modules and packages and executing the main Bot run function. The main Bot run function calls predefined functions from Pipeline_Functions.py and configs.py to pre-process and process data, train the model, and evaluate model performance using metrics like MAE and MSE. The Bot systematically executes the pipeline to ensure optimal model performance and accurate logging of data, with steps like initializing the logger and invoking the parse_args() function. Finally, the run_model_on_task() method is used to train the ML Regression Model and calculate metrics, which are printed on the console and logged on OpenML.

4.2. configs / Second Module: The second module, configs.py, creates configuration spaces that contain Hyperparameters, which impact the Model's performance. These Hyperparameters are passed as configuration spaces by importing the argparse package from the sklearn package, and each ML Algorithm has SHp that affect its model performance, resulting in unique configuration spaces and parameters for each Regression algorithm.

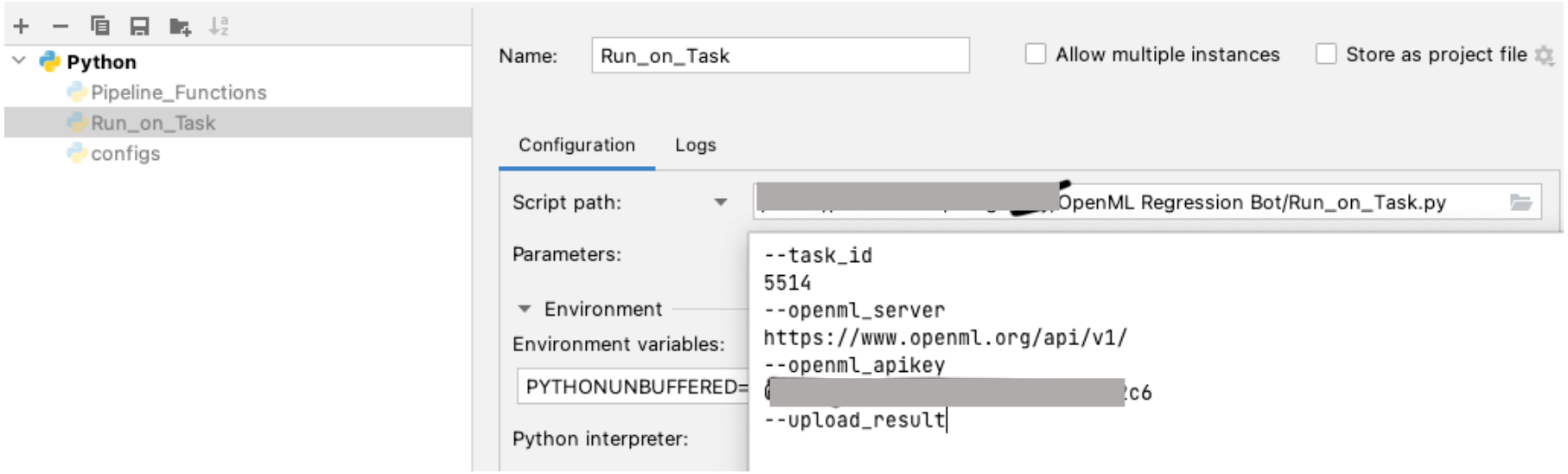

Fig. 4.2. User pre-defined configurational parameters in PyCharm IDE

4.3. Pipeline_Functions / Third Module: The third module, Pipeline_Functions.py, defines functions for specific tasks such as pre-processing data, creating the final pipeline, and deleting locally logged results. These functions can be executed efficiently by passing the necessary configuration parameters to the Bot, making it convenient for the user. The results can be immediately published to the OpenML server, streamlining the workflow and saving time.

5

Results & Discussions

5.1. Results: The Bot uploads the obtained results, including MAE, MSE, etc., to the OpenML server.

i. Results:

(a). for the selected Decision_Tree Regressor Algorithm:

INFO:openml.config:Executed Task 5514 with Flow id:19343 INFO:root:Task 5514 - bodyfat; Accuracy: 0.56
mean_absolute_error: 0.5612923076923078 mean_squared_error: 2.218947692307692
r2_score: 0.9648830358321332
upload_result= True
INFO:root:Starting [post] request for the URL https://www.openml.org/api/v1/run/ INFO:root:1.8374112s taken for [post] request for the URL https://www.openml.org/api/v1/run/
OpenML Run ========== Uploader Name: None Metric.....: None
Run ID.....: 10591952
Run URL.....: https://www.openml.org/r/10591952
Task ID.....: 5514
Task Type....: None
Task URL.....: https://www.openml.org/t/5514
Flow ID.....: 19343
Flow Name....: sklearn.pipeline.Pipeline(columntransformer=sklearn.compose._column_tra
nsformer.ColumnTransformer(numeric=sklearn.pipeline.Pipeline(simpleimp
uter=sklearn.impute._base.SimpleImputer,standardscaler=sklearn.preprocess
ing._data.StandardScaler),nominal=sklearn.pipeline.Pipeline(onehotencoder
=sklearn.preprocessing._encoders.OneHotEncoder)),decisiontreeregressor=s klearn.tree._classes.DecisionTreeRegressor)
Flow URL.....: https://www.openml.org/f/19343
Setup ID.....: None
Setup String.: Python_3.9.13. Sklearn_1.2.2. NumPy_1.21.5. SciPy_1.10.1. Dataset ID...: 560
Dataset URL..: https://www.openml.org/d/560
Results have been Uploaded to Openml while stored locally in Evaluations folder .

(a). for the selected Random_Forest Regressor Algorithm:

mean_absolute_error: 1.4074446102326155 mean_squared_error: 4.537086745615215
r2_score: 0.9178312996369519
upload_result= True
INFO:openml.config:Executed Task 5514 with Flow id:19351 INFO:root:Task 5514 - bodyfat; Accuracy: 1.41 INFO:root:Starting [post] request for the URL https://www.openml.org/api/v1/run/
INFO:root:1.9338472s taken for [post] request for the URL https://www.openml.org/api/v1/run/
OpenML Run
==========
Uploader Name: None
Metric.....: None
Run ID.....: 10591953
Run URL.....: https://www.openml.org/r/10591953 Task ID.....: 5514
Task Type....: None
Task URL.....: https://www.openml.org/t/5514
Flow ID.....: 19351
Flow Name....: sklearn.pipeline.Pipeline(columntransformer=sklearn.compose._column_tran
sformer.ColumnTransformer(numeric=sklearn.pipeline.Pipeline(simpleimput
er=sklearn.impute._base.SimpleImputer,standardscaler=sklearn.preprocessin
g._data.StandardScaler),nominal=sklearn.pipeline.Pipeline(onehotencoder=s
klearn.preprocessing._encoders.OneHotEncoder)),randomforestregressor=**sk learn.ensemble._forest.RandomForestRegressor**)
Flow URL.....: https://www.openml.org/f/19351
Setup ID.....: None
Setup String.: Python_3.9.13. Sklearn_1.2.2. NumPy_1.21.5. SciPy_1.10.1. Dataset ID...: 560
Dataset URL..: https://www.openml.org/d/560
Results have been Uploaded to Openml while stored locally in Evaluations folder .

Ref. to the uploaded results on OpenML Server:

Fig. 5.1. One example run for the ref. to the stored runs on OpenML

ii. Observations: From the obtained results, it is observable that OpMLRB has successfully performed the the run on specified Task ID and logged the results in a specified local directory, OpenML server as it intend to. In addition, it is clearly witnessed that on this specific Task ID; the pipeline with Decision_Tree Algorithm was the better choice.

5.2. Discussions: The OpenML Regression Bot was successful in automatically selecting the appropriate Regression Model, creating a Local Directory to store results and uploading obtained results to the OpenML server. This allowed for the analysis of model performance using observed metrics such as MAE and MSE.

6
Conclusion

The Open Regressor Bot has successfully implemented for OpenML Regression Tasks, with the ability to select Regression Models, store results and upload them to the OpenML server. It is recommended for OpenML users for Regression Tasks Analysis and Automated Model Selection.

Logged results to the local directory 

Uploaded results to the OpenML Server 

7
Summary

The OpenML platform provides a great environment for experimentation and testing in Machine Learning. However, the scarcity of available Regression Tasks on OpenML has never got equal importance from researchers to analyse these types of tasks using Automated Model Selection, like with OpenML Classification tasks. Therefore, the Open Regressor Bot was developed to help researchers more easily analyse Regression Tasks using a pre-defined ML pipeline, allowing for greater flexibility and customization when selecting the appropriate algorithm by it's name and configuration space for the task at hand.

8

Future Work and Limitations

In the scope of running a simulation for OpenML Regression Tasks, the Bot encountered an error for ARDRegression, while it was performing great with Regression Algorithms such as Decision_Tree, Random_Forest etc.. The occured error in ARDRegression case:

Traceback (most recent call last):
validate_parameter_constraints(
File "/Users/pavankumarperugu/Roy/OpenML Regression Bot/venv/lib/python3.9/site-packages/sklearn/utils/_param_validation.py", line 97, in validate_parameter_constraints
raise InvalidParameterError(sklearn.utils._param_validation.InvalidParameterError: The 'fit_intercept' parameter of ARDRegression must be an instance of 'bool', an instance of 'numpy.bool_' or an instance of 'int'. Got 'True' instead.

Acknowledgements

I am grateful to Prof. Dr.-Ing. Jöran Beel for giving me the opportunity to work on 'An OpenML bot for Regression' and guiding me throughout the project. I also express my sincere thanks to Mr. Lennart, Purucker, for his guidance and technical support in introducing me to Machine Learning and the scope of work. Their help and support were invaluable from the beginning to the end of this StudentArbrit.

References

[1]. PavanKumarPerugu, “PavanKumarPerugu/OpenML-Regression-Bot,” *GitHub*, Mar. 16, 2023. https://github.com/PavanKumarPerugu/OpenML-Regression-Bot (accessed Mar. 16, 2023).

[2]. “OpenML,” *Openml.org*, 2023. https://www.openml.org/ (accessed Mar. 16, 2023).

[3]. “ISG Siegen – Intelligent Systems,” *Beel.org*, 2022. https://isg.beel.org/ (accessed Mar. 16, 2023).

[4]. “BEEL, Joeran (Head of Group) – ISG Siegen,” *Beel.org*, 2023. https://isg.beel.org/people/joeran-beel/ (accessed Mar. 16, 2023).

[5]. F. Hutter, “Automated configuration of algorithms for solving hard computational problems,” 2017. https://www.semanticscholar.org/paper/Automated-configuration-of-algorithms-for-solving-Hutter/08c4fdd974d874c87ea87faa6b404a7b8eb72c73 (accessed Mar. 16, 2023).

[6]. F. Hutter, “AutoML | Wrapping Up AutoML-Conf 2022 and Introducing the 2023 Edition,” *Automl.org*, 2022. https://www.automl.org/automl-blog/ (accessed Mar. 16, 2023).

Queries Please..!

Thank You..!

Perugu, Pavan Kumar
1599044