AutoML - Project Report

Jagadeesha (163059002), Ebin Chacko (153079009),
Nagma Khan (153079030), Sanjeev kumar (16305R008),
Vivek Mishra (143079005)

Abstract

In today's world where we are trying to make sense out of large amount of data and there is an unmet need of skilled data analysts, there is a dire need for algorithms which help us in deciding a model along with its tuned parameters given data and the task at hand as input. There is an emerging field of research termed **AutoML** which addresses this problem. Auto-ML aims at automating the process of model selection, parameter tuning and other aspects of learning. We propose to build a model which returns a good **regression** model for a wide variety of datasets. If not anything at least these models can be used as a starting point for more research.

1 Introduction

Here we are building a class which returns a regression object which can be used on a test dataset for prediction. This model will give best accuracy out of the regression models considered in this approach. The overall model has roughly these modules:

1. Dataset Preprocessing
2. Feature encoder
3. Feature Selection
4. Data Splitter
5. Hyper Parameter Optimizaiton
6. Model Selector

   The regression models considered in this class are: *Support Vector Regression with RBF and Polynomial Kernel, Ridge Regression and Lasso Regression.*

2 Framework Description

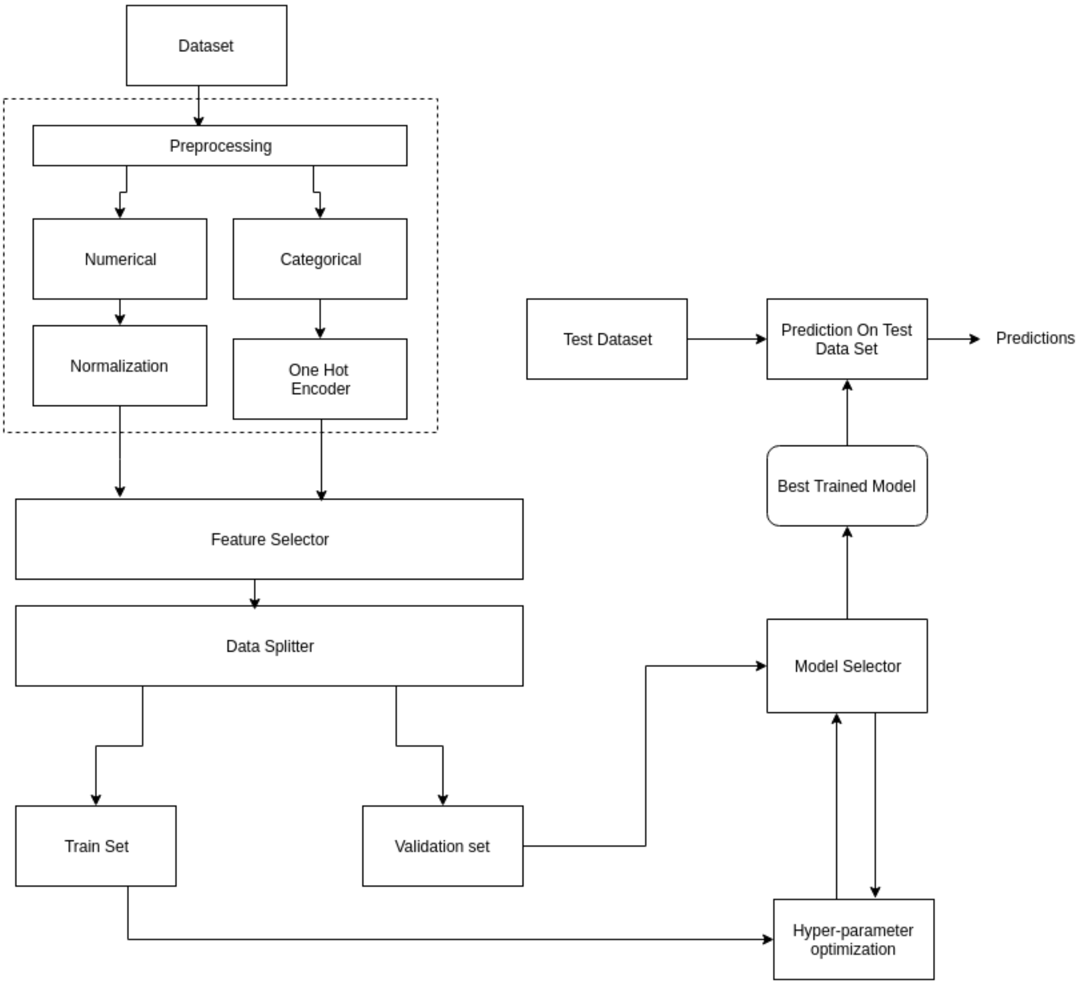

Figure 1: Framework Block Diagram

The approach will be along the lines of what has been described in [2].

We have implemented an *autoRegressor* class in python with all required methods related to preprocessing, feature extraction, Hyperparameter optimization and Model Selection.

2.1 Dataset Preprocessing

The dataset (both train and test) are stored in csv format and filename of the train dataset, test dataset as well as the label of the target column is specified as input to the constructor while creating the object of *autoRegressor* class.

The train dataset is split by the data splitter into training and validation sets in the ratio 8:2 i.e. 80 % of the dataset is to be used for training and rest 20% for validation.

2.2 Feature encoder

This part does one-hot encoding for categorical features and does feature normalization.

2.3 Feature extractor

For feature extraction we have used a linear model with L1 norm. This was selected because these models tend to have a sparse solutions, i.e, many of their estimated coefficients tend to have zero values. This can be used to reduce the dimensionality of the data and use this with another model.

2.4 Parameter tuning

The class *autoRegressor* has a *hyperOpt* method which does the parameter tuning. This module does hyperparameter tuning of all the regression models considered in this class i.e. SVR (RBF and Poly kernel), Ridge and Lasso on the training data set. The parameters tuned for Ridge and Lasso Regression is α, for SVR RBF is C and *gamma* and for SVR Poly is C and polynomial degree.

The parameter C, common to all SVM kernels, trades off misclassification of training examples against simplicity of the decision surface. A low C makes the decision surface smooth, while a high C aims at classifying all training examples correctly. *Gamma*, specific to the SVR RBF kernel, defines how much influence a single training example has. The larger gamma is, the closer other examples must be to be affected [9].

In case of Ridge regression $\alpha \geq 0$ is a complexity parameter that controls the amount of shrinkage: the larger the value of α, the greater the amount of shrinkage and thus the coefficients become more robust to collinearity. While in case of Lasso, the α parameter controls the degree of sparsity of the coefficients estimated [10].

Scoring method used is *mean absolute error* while deciding best parameters.

2.5 Model Selector

It compares the performance of different models on the validation set and returns the best model i.e. the model with least mean absolute error.

Their is also provision for getting the predictions of the best model on the test dataset.

3 Data Sets Description

Online News Popularity Dataset [8]

- Source: UCI machine learning repository [5]
- Data Description: This dataset summarizes a heterogeneous set of features about articles published by Mashable in a period of two years. The goal is to predict the number of shares in social networks (popularity) [6].
- Number of Attributes: 61
- Number of Train Data Samples used: 39,644
- Number of Test Data Samples used: 9000

House Prices Prediction [11]

- Source: Kaggle

- Data Description: This playground competition's dataset proves that much more influences price negotiations than the number of bedrooms or a white-picket fence. With 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa, this competition challenges you to predict the final price of each home [11].
- Number of Attributes: 80
- Number of Train Data Samples used: 1400
- Number of Test Data Samples used: 1400

4 Results

Online News Popularity Dataset

- Best Model : Support Vector Regression with RBF Kernel
- Pre-processing: Feature Reduction with Lasso
- Optimal Parameters: $C = 216.5$ and $gamma = 0.3753$
- Error on Test Dataset (Mean Absolute Error): 2252.5

House Prices Prediction

- Best Model : Ridge Regression
- Pre-processing: none
- Optimal Parameters: $\alpha = 2.967$
- Error on Test Dataset (Mean Absolute Error): 3700

5 Future Work

- More feature extraction techniques like Kernel PCA, Fast ICA, Polynomial Combination of Features can be considered.
- We can train a neural network and extract the features from the penultimate layer, this will lead to effective feature dimensionality reduction and proper learning.
- We can include more advanced Regression models or ensemble of models.

6 References

[1] I. Guyon et al., "Design of the 2015 ChaLearn AutoML challenge", 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, 2015, pp. 1-8.

[2] A. Thakur and A. Krohn-Grimberghe, "Autocompete: A framework for machine learning competitions", In AutoML Workshop, International Conference on Machine Learning 2015, 2015.

[3] M. Feurer et al., “Methods for Improving Bayesian Optimization for AutoML”, In AutoML Workshop, International Conference on Machine Learning 2015, 2015.

[4] A. Thakur , “AutoML Challenge: Rules for Selecting Neural Network Architectures for AutoML-GPU Challenge”, In AutoML Workshop, International Conference on Machine Learning 2016, 2016.

[5] https://archive.ics.uci.edu/ml/datasets.html

[6] https://archive.ics.uci.edu/ml/datasets/Online+News+Popularity

[7] http://automl.chalearn.org/

[8] K. Fernandes, P. Vinagre and P. Cortez. A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News. Proceedings of the 17th EPIA 2015 - Portuguese Conference on Artificial Intelligence, September, Coimbra, Portugal.

[9] http://scikit-learn.org/stable/modules/svm.html

[10] http://scikit-learn.org/stable/modules/linear_model.html

[11] https://www.kaggle.com/c/house-prices-advanced-regression-techniques