Machine Learning Engineer Nanodegree

Capstone Proposal

David Havera
2/19/2018

Proposal

I have often said the only thing that I dislike about Kaggle competitions is that they have an end date. In September of 2017, I competed in the Zillow's Home Value Prediction competition that was hosted by Kaggle. This was my first competition and I was very proud to score within the top 60 percentile. My approach was very simple. First, I removed categorical variables. Second, I completed the following data preparation steps via a pipeline:

- Selector - Converted DataFrame to a NumPy array
- Imputer – Replaced missing values with median
- Standard Scaler - Not needed for decision trees, but needed for other algorithms
- SelectKBest - Selects the most important features
- PCA – Feature reduction into correlated components.
- Random Forest Regressor - Used to make predictions

I chose the Random Forest Regressor because of the following benefits.

- Limited parameter tuning
- Fast and versatile
- Great for feature selection because algorithm evaluates a lot of variations

Next, I I ran a grid search to optimize the pipeline parameters for KBest and PCA. Finally, I used a random search to tune the Random Forest Regressor because I discovered that it can be more effective than a grid search [1].

I would like to update the benchmark model to evaluate its performance against several other regression algorithms. Furthermore, I would like to create a template that I can use to approach supervised learning problems in the future. The template will have the following steps:

- Exploratory data analysis – I will include visualizations such as heat maps, correlation matrixes, and scatterplots
- Data Cleaning (scaling, imputing, etc) – I will include in a pipeline for the numeric features
- Feature Engineering – I will evaluate the performance with and without feature engineering
- Kfold cross validation
- Untuned/Tuned algorithm comparisons with box plots – Box plots will allow me to review the results of the kfold cross validation scores variability and average
- Algorithm tuning using grid and random search – I will tune all the algorithms to evaluate the kfold cross validation scores
- Learning curve analysis for top performing algorithms – Learning curves will be used to evaluate a model's bias and variance
- Stacking of top performing algorithms – Stacking can improve model performance by combining top performing models

This template is critical to me professionally because I have just been promoted to an Analytics Engineer at GE Aviation. I am responsible for using machine learning algorithms to deliver cost-out projects at GE Aviation. My cost-out target for 2018 is $150 million. To achieve this target, I analyze the manufacturing cost for engine parts to determine if there are ways to make the parts cheaper at our factories. The research for aircraft engine parts is difficult because GE Aviation has more than 40 pillar IT systems across 65 factories. The task of making better engines for a cheaper cost requires a lot of data munging to succeed. However, there is a huge opportunity in this area because the information has never been consolidated and reviewed by machine learning algorithms.

I manage an offshore team of data scientists, so it's critical that we are all on the same page as we progress toward our common goal. I need a standard machine learning template that I can implement as a starting point as we are assigned new productivity projects each quarter. The template that I create during my capstone will be operationalized with my overseas team and used to investigate areas of savings. The template will also ensure that we are evaluating the machine learning algorithms globally with the same practices such as using learning curves.

Domain Background

(approx. 1-2 paragraphs)

"Zestimates" are estimated home values based on 7.5 million statistical and machine learning models that analyze hundreds of data points on each property. And, by continually improving the median margin of error (from 14% at the onset to 5% today), Zillow has since become established as one of the largest, most trusted marketplaces for real estate information in the U.S. and a leading example of impactful machine learning.

A home is often the largest and most expensive purchase a person makes in his or her lifetime. Ensuring homeowners have a trusted way to monitor this asset is incredibly important. The Zestimate was created to give consumers as much information as possible about homes and the housing market, marking the first-time consumers had access to this type of home value information at no cost. Winning algorithms stand to impact the home values of 110M homes across the U.S." [2]

Problem Statement

(approx. 1 paragraph)

"Zillow Prize, a competition with a one million dollar grand prize, is challenging the data science community to help push the accuracy of the Zestimate even further.

In this competition, you are going to predict the logerror between their Zestimate and the actual sale price, given all the features of a home for the months in Fall 2017. The log error is defined as

$$logerror = log(Zestimate) - log(SalePrice)$$

and it is recorded in the transactions training data. Submissions are evaluated on **Mean Absolute Error** between the predicted log error and the actual log error." [2]

Datasets and Inputs

"You are provided with a full list of real estate properties in three counties (Los Angeles, Orange and Ventura, California) data in 2016.

The train data has all the transactions before October 15, 2016, plus some of the transactions after October 15, 2016.

File descriptions

properties_2016.csv - all the properties with their home features for 2016. Note: Some 2017 new properties don't have any data yet except for their parcelid's. Those data points should be populated when properties_2017.csv is available.

properties_2017.csv - all the properties with their home features for 2017 (released on 10/2/2017)

train_2016.csv - the training set with transactions from 1/1/2016 to 12/31/2016

train_2017.csv - the training set with transactions from 1/1/2017 to 9/15/2017 (released on 10/2/2017)

sample_submission.csv - a sample submission file in the correct format

Data fields - Please refer to zillow_data_dictionary.xlsx" [2]

Solution Statement

(approx. 1 paragraph)

The capstone model will enhance the benchmark model by including cross validation, box plots, and learning curves. Additionally, the Mean Absolute Error (MAE) will improve with the capstone model.

Benchmark Model

(approximately 1-2 paragraphs)

The benchmark model is included in the attachments and scored in the top 60% of the Kaggle competition. My goal is to make this benchmark model more robust as described in the proposal section.

Evaluation Metrics

(approx. 1-2 paragraphs)

This competition is based on Mean Absolute Error (MAE) of the forecasted versus actual sale price. The final model will have the lowest MAE and will be scored as a "Late Submission" on Kaggle's website. Additionally, final model will have the lowest MAE after kfold cross validation and will illustrate a learning curve with low variance and bias.

Project Design

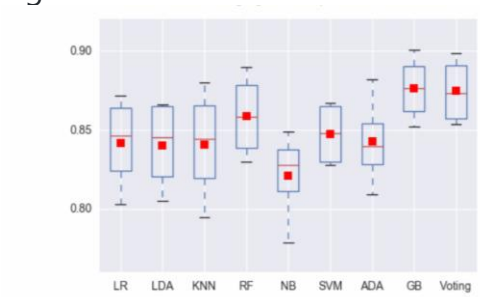
(approx. 1 page)

The key improvements to the benchmark models are as follows:

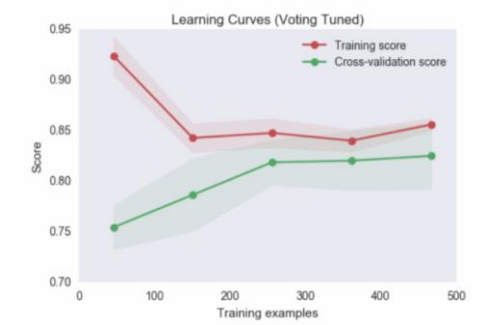
- Evaluation of multiple algorithms with kfold cross valuation as illustrated below:

```
In [23]: # base case
         # Spot-Check Algorithms
         LR_clf = LogisticRegression()
         LDA_clf = LinearDiscriminantAnalysis()
         KNN_clf = KNeighborsClassifier()
         RF_clf = RandomForestClassifier(random_state = 7)
         NB_clf = GaussianNB()
         SVC_clf = SVC()
         ADA_clf = AdaBoostClassifier(random_state = 7)
         GB_clf = GradientBoostingClassifier()

         models = []
         models.append(('LR', LR_clf))
         models.append(('LDA', LDA_clf))
         models.append(('KNN', KNN_clf))
         models.append(('RF',RF_clf))
         models.append(('NB', NB_clf))
         models.append(('SVM', SVC_clf))
         models.append(('ADA', ADA_clf))
         models.append(('GB', GB_clf))

         results = []
         names = []
         for name, model in models:
             cv_results = cross_val_score(model, X_train, y_train, cv=sss, scoring=scoring)
             results.append(cv_results)
             names.append(name)
             msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
             print(msg)
```

- Box plot illustration of the cross-validation scores to select top performing algorithms:



- Learning Curve Analysis of the top performing algorithms:



Bibliography

[1] A. E. Deeb, "Rants on Machine Learning," Medium, 22 Jun 2015. [Online]. Available: https://medium.com/rants-on-machine-learning/smarter-parameter-sweeps-or-why-grid-search-is-plain-stupid-c17d97a0e881. [Accessed 18 Febr 2018].

[2] "Zillow Prize: Zillow's Home Value Prediction (Zestimate)," Kaggle, [Online]. Available: https://www.kaggle.com/c/zillow-prize-1#description. [Accessed 18 Feb 2018].