Name: SAHIL

Email address: sahiltinky94@gmail.com

Contact number: 9711308110

Anydesk address:

Date: 02nd Apr 2020

Self Case Study -1: Don’t Overfit! II

Overview

*** Write an overview of the case study that you are working on. ***(MINIMUM 200 words)*** ***

1. **Kaggle problem**: Don’t Overfit! II is a challenging problem where we must avoid models to be overfitted (or crooked way to learn) given very small amount of training samples.

   As per Kaggle say,” It was a competition that challenged mere mortals to model a 20,000x200 matrix of continuous variables using only 250 training samples… *without overfitting*. “

   Dataset can be download here: https://www.kaggle.com/c/dont-overfit-ii/overview
   Dimension of train.csv – 250 samples and 300 features and 1 class label and 1 Id: (250,302)

   Dimension of test.csv – 19750 samples and 300 features and 1 Id: (19750,301)

   So, with the small amount of train data given, we must do to task carefully to avoid overfitting easily.

What do we need to predict? We are predicting the binary target value (**binary classification**) associated with each row which contains 300 continuous feature values. Also without overfitting with the minimal set of training samples given.

2. **Evaluation**: As per Kaggle problem statement, the score will be evaluated based on **AUCROC** between predicted target and actual target.

Research-Papers/Solutions/Architectures/Kernels

*** Mention the urls of existing research-papers/solutions/kernels on your problem statement and in your own words write a detailed summary for each one of them. If needed you can include images or explain with your own diagrams. ***

1. Hyperparameter Tuning (Most Vote Blog: https://towardsdatascience.com/hyperparameter-tuning-c5619e7e6624)
    a. Ridge, Lasso, Elastic, LassoLars (Lasso model fit when Least Angle Regression), Bayesian Ridge regression, Logistic Regression and SGD classifier were used Machine Learning models and evaluate mean cross validate score and their standard deviation. The parameters keep as per the default values

    b. Found that logistic regression and SGD are the two top model better than others. Then they apply tuning strategy (Both GridSearchCV and RandomSearchCV) in these models only.

    c. Best CV score (not Kaggle Score) got using GridSearchCV in Logistic regression model: 0.789

2. Just Don’t Overfit (Blog: https://medium.com/analytics-vidhya/just-dont-overfit-e2fddd28eb29)

a. Used Standardization to fit in range between -1 and 1 in training data and transform in test data based on mean value and standard deviation evaluated from training data.

b. Used regression model LASSOCV (Least Absolute Shrinkage and Selection Operator Cross Validation). This LASSO model used to find the hyperplane which reduce the residual error with the additional shrinkage parameter λ with absolute of weights to avoid overfitting and they did cross validation with different value of λ to find the right hyperplane.

c. Test data score: 0.843

3. Don’t Overfit! – How to prevent Overfitting in your Deep Learning Models (Blog: https://nilsschlueter.de/blog/articles/dont-overfit-%E2%80%8A-%E2%80%8Ahow-to-prevent-overfitting-in-your-deep-learning%C2%A0models/)

   a. Used base model as MLP Deep Learning which contain two hidden layers: 128 units and 64 units. Since it’s a binary classifier, loss used binary cross entropy and used Adam optimizer. Kaggle score: 59%.

   b. Next approach, they create simplified model which contain one hidden layers 8 units and applying Dropout layer with 0.4 dropout rate. With Additionally, they put the earlycallback at val_loss with patience as 3. For this new model, Kaggle score achieved 80%.

4. How to not overfit (Most voted kernel: https://www.kaggle.com/artgor/how-to-not-overfit)

   a. Perform EDA: Plots on fee features, Correlation score among features, and did Basic modelling (Logistic Regression). Got Kaggle score: 0.7226

   b. Using ELI5 tools which give the weight importance for model. Observed and took top 32 importance and trained basic modelling again. Kaggle Score: 0.7486

c. Concluding, after performing various feature selections technique like Permutation importance, SHAP and SequentialFeatureSelector, its didn't improve very much.

d. Then they perform different various models with hyperparameter – Logistic regression, Gaussian Naïve Bayes, Adaboost, Extratrees, Random Forest, Gaussian Process classification, Support Vector Classificaion (SVC), kNN, Bernoulli Naïve Bayes, SGD. Blend with Logistic regression and SVC, Kaggle Score: 0.831

e. They tried feature engineering like Polynomial Features, Adding statistics, adding distance features by taking kNN with k=5 to calculate mean, max and min distance. And then perform several feature selection using by sklearn package like percentile, SelectKBest, RFE and apply model Logistic Regression and GLM. Still CV score remains below 80%.

First Cut Approach

*** Explain in steps about how you want to approach this problem and the initial experiments that you want to do. ***(MINIMUM 200 words)*** ***

1. **EDA**: Explore about the data.

    a. Plot PDF and CDF of few features and compare PDF's with other features and write observation about it.

    b. Plot Boxplot for all features and write observation about it.

    c. Check and Impute missing values

    d. Deal with imbalance data. (SMOTE: Synthetic Minority Oversampling Techniques)

2. Creating **feature engineering** by adding new statistics – mean of each sample, standard deviation of each sample, distance measure between each pair of samples. (added using: Euclidean and Manhattan distance)

3. Apply **Normalization** (and/or **Standardization**): There no rule of thumb which normalization work best. So, we will try with both and compare them

4. If number of features are large enough, we will try to used **feature selection**.

5. Apply and Comparison of **Machine Learning Models** (with hyperparameters and cross validation) and Plot Confusion Matrix on CV data (Since, we don't know about test's data target value, so we can observe with cv data).

    a. kNN: Starting with the simplest model and consider this as benchmark model. It's a distance-based model.

    b. Naïve Bayes: Try to do with probabilistic-based model

    c. Logistic Regression: Try to do to find hyper plane whether there exist linear separable between them.

    d. SVM: Try to find the hyper plane with nonlinear kernel RBF function to check if there exist nonlinear separable.

6. Try to perform even powerful Machine Learning (i.e. Ensemble Model): Random Forest and Gradient Boosting (XGBoost) and stacking Model.