Ensemble Modeling for Performance Improve
Introduction
Feature Engineering
Modelling
Ensembling
Conclusion
Bayesian Optimization Methods for Datasets with Small-Numbered Features

Chapter 1: Ensemble Modeling for Performance Improve

using sklearn and xgboost packages in Python on ADCTL dataset.

Yusuf Baran Tanrıverdi[1]

University of Cassino and Southern Lazio[1],

May, 2023

MAIA

Ensemble Modeling for Performance Improve
Introduction
Feature Engineering
Modelling
Ensembling
Conclusion
Bayesian Optimization Methods for Datasets with Small-Numbered Features

Introduction

- The objective is to improve the performance of 11 different regressor models.
- The initial individual models yield AUC scores ranging from 80
- Various ensemble techniques are explored to enhance the overall performance.
- The motivation is to create a generalizable pipeline for all three datasets provided.

Challenges

- The regressors can be unsuitable for classification purposes - i.e. giving a better performance with logits.
- They can be highly correlated.
- Feature reduction thresholds must be carefully selected.
- One should be cautious with dataset split (i.e. train, validation and test).

Principal Component Analysis

I used PCA preprocessing provided by the library. This reduces the number of features to number of samples by solving a SVD problem on the training dataset.

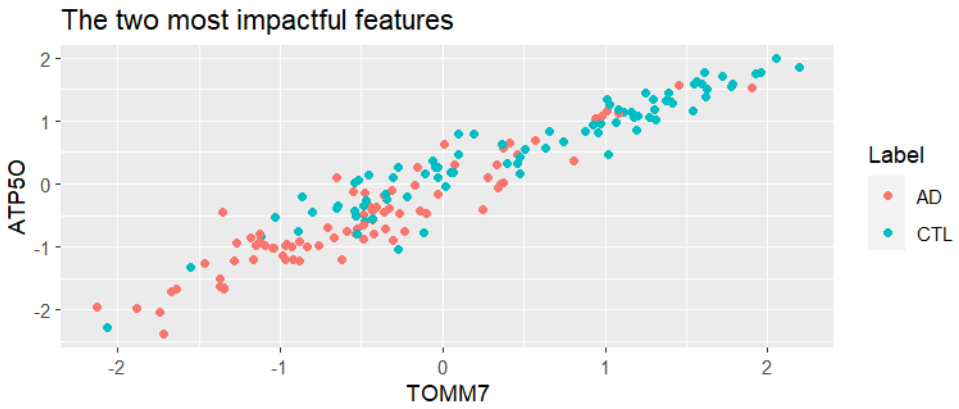

Figure: ADCTLDataset: Principal Component Analysis

Train Control with k-Fold CV

Ensemble Modeling for Performance Improve
Introduction
Feature Engineering
Modelling
Ensembling
Conclusion
Bayesian Optimization Methods for Datasets with Small-Numbered Features

I used 15-fold and 5-repeat cross-validation to get training performance.

```
# Create a cross-validation strategy
cv = RepeatedStratifiedKFold(n_splits=15, n_repeats=5,
    random_state=107)
```

I provide the results of the training next slide. Note that, this is the result of a 70 % of training set, split in a stratified fashion.

```

# Set the random seed for reproducibility
np.random.seed(123)
# Calculate the sample size.
# Perform random sampling with class distribution preservation.
train_set, val_set = train_test_split(train_val_set, test_size
    =0.30, stratify=y)
```

Diverse Classifiers in sklearn/xgboost and CV-Test Results

Model	Accuracy	AUC	MCC	MSE	F1	Recall	AP
Bayesian Ridge	0.649	0.933	0.378	0.136	0.427	0.317	0.952
MLP	0.838	0.837	0.700	0.162	0.835	0.849	0.811
LR	0.878	0.880	0.776	0.122	0.875	0.856	0.811
RF	0.661	0.661	0.339	0.339	0.677	0.647	0.743
SVC RBF	0.866	0.866	0.753	0.134	0.855	0.982	0.790
KN	0.820	0.819	0.659	0.180	0.837	0.790	0.746
SVC Linear	0.743	0.747	0.539	0.258	0.836	0.876	0.790
GBM	0.800	0.801	0.620	0.200	0.758	0.780	0.846
PLS	0.490	0.666	0.000	0.243	0.467	0.743	0.839
SVC Poly.	0.873	0.873	0.759	0.127	0.873	0.892	0.846
XGB Linear	0.870	0.871	0.756	0.130	0.868	0.839	0.839

Now, we have 11*75= 875 different models and we want to somehow utilise all the classifiers. This is what ensembling will do.

Ensembling via sklearn.ensemble

Ensemble Modeling for Performance Improve
Introduction
Feature Engineering
Modelling
Ensembling
Conclusion
Bayesian Optimization Methods for Datasets with Small-Numbered Features

```
# Separate the features and target variable
X_stacked = val_preds.copy()
y_stacked = val_set.iloc[:, -1]
# Split the stacked data into training and testing sets
X_train_stacked, X_test_stacked, y_train_stacked, y_test_stacked =
    train_test_split(
    X_stacked, y_stacked, test_size=0.5, random_state=42
)
# Initialize the meta-model (Gradient Boosting classifier was the
    best empirically)
meta_model = GradientBoostingClassifier()
# Train the meta-model
meta_model.fit(X_train_stacked, y_train_stacked)
# Evaluate the meta-model on the test set
meta_val_test_preds = meta_model.predict(X_test_stacked)
```

Ensemble Modeling for Performance Improve
Introduction
Feature Engineering
Modelling
Ensembling
Conclusion
Bayesian Optimization Methods for Datasets with Small-Numbered Features

Conclusion

Here are the results with meta-modeling with Gradient Boosting Classifier. We see metrics are well-balanced comparing the training results for the classifiers.

Table: Meta-Modelling Validation Results

Model	Accuracy	AUC	MCC	MSE	F1	Recall	AP
Meta-GBM	0.88	0.883	0.757	0.12	0.857	0.9	0.776

Ensemble Modeling for Performance Improve
Introduction
Feature Engineering
Modelling
Ensembling
Conclusion
Bayesian Optimization Methods for Datasets with Small-Numbered Features

Chapter 2:Bayesian Optimization Methods for Datasets with Small-Numbered Features

using sklearn and skopt packages in Python on ADMCI dataset

Yusuf Baran Tanrıverdi[1]

University of Cassino and Southern Lazio[1],

May, 2023

MAIA

Bayesian Optimization

- Bayesian Optimization is a powerful technique for the global optimization of expensive black-box functions.
- It uses a probabilistic surrogate model, such as Gaussian Process, to model the unknown function.
- By iteratively selecting promising points to evaluate, it aims to find the global optimum efficiently.
- The surrogate model guides the search by balancing exploration and exploitation.

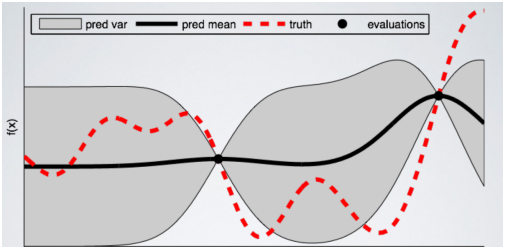

Figure: Illustration of Bayesian Optimization

Search Spaces

Ensemble Modeling for Performance Improve
Introduction
Feature Engineering
Modelling
Ensembling
Conclusion
Bayesian Optimization Methods for Datasets with Small-Numbered Features

I used conventional ranges for search spaces for each model's grid. One example is shown below.

```
# Create a search space for Gradient Boosting Classifier
    {
        'learning_rate': Real(0.01, 0.2),
        'n_estimators': Integer(100, 1000),
        'max_depth': Integer(1, 10),
        'min_samples_split': Integer(2, 20),
        'min_samples_leaf': Integer(1, 10),
        'max_features': Categorical(['auto', 'sqrt', 'log2'])
    },
```

Then I obtained the best models as in the next slide. Note that, when I get results, I applied CV-fold as in chapter one to fit the models again.

Obtain Best Models

Ensemble Modeling for Performance Improve
Introduction
Feature Engineering
Modelling
Ensembling
Conclusion
Bayesian Optimization Methods for Datasets with Small-Numbered Features

```
best_models = []
# Perform Bayesian search
for i, (model_name, model) in enumerate(models):
    bayes_search = BayesSearchCV(
        model, search_spaces[i], cv=5, scoring=make_scorer(
    custom_matthews_corrcoef), n_jobs=-1
    )
    bayes_search.fit(train_set.iloc[:, :-1], train_set["Label"])

    best_model = bayes_search.best_estimator_
    best_params = bayes_search.best_params_

    best_models.append((model_name, best_model, best_params))

best_models = np.asarray(best_models)[:, 1]
```

Cross-Fold Validation Results of Optimized Models

Ensemble Modeling for Performance Improve
Introduction
Feature Engineering
Modelling
Ensembling
Conclusion
Bayesian Optimization Methods for Datasets with Small-Numbered Features

Model	Accuracy	AUC	MCC	MSE	F1	Recall	AP
BayesianRidge	0.500	0.806	0.074	0.190	0.079	0.049	0.857
MLP	0.667	0.665	0.342	0.333	0.68	0.699	0.667
LR	0.732	0.732	0.478	0.268	0.737	0.742	0.718
PLS	0.55	0.775	0.198	0.225	0.233	0.155	0.835
XGB	0.728	0.728	0.476	0.272	0.739	0.764	0.711
RF	0.678	0.678	0.370	0.322	0.691	0.714	0.669
SVC RBF	0.708	0.707	0.429	0.292	0.726	0.754	0.697
KN	0.693	0.688	0.413	0.307	0.736	0.829	0.663
SVC Poly.	0.727	0.726	0.469	0.746	0.201	0.779	0.708
GBM	0.71	0.712	0.443	0.290	0.727	0.772	0.695
SVC Linear	0.725	0.725	0.464	0.737	0.137	0.748	0.711

Next, I applied ensemble modeling again over the models except for Bayesian Ridge (due to very low correlation). This time Random Forest Classifier is the base. There, I use Random-Search optimization for the meta-model. Then, I obtain the best estimator following the snippet below.

Ensemble Modeling for Performance Improve
Introduction
Feature Engineering
Modelling
Ensembling
Conclusion
Bayesian Optimization Methods for Datasets with Small-Numbered Features

Meta-modelling with Stacked Data using RF

```
base_model = RandomForestClassifier()

# Perform random search
meta_model = RandomizedSearchCV(base_model, param_grid, n_iter=10,
     scoring=cv_metrics, refit='matthews_corr', cv=5, random_state
    =42)

# Train the meta-model
meta_model.fit(X_train_stacked, y_train_stacked)

optimized_gb_model = meta_model.best_estimator_
```

The results are much more balanced in terms of metrics that we observe as criteria for overfitting!

Table: Meta-Modelling Validation Results

Model	Accuracy	AUC	MCC	MSE	F1	Recall	AP
Meta-RF	0.730	0.720	0.461	0.269	0.666	0.583	0.646

Ensemble Modeling for Performance Improve
Introduction
Feature Engineering
Modelling
Ensembling
Conclusion
Bayesian Optimization Methods for Datasets with Small-Numbered Features

Chapter 3: A Pipeline: Preprocessing, Bayesian Search, Cross-Fold Validation and Meta-Modelling

using sklearn and skopt packages in Python on MCICTL dataset

Yusuf Baran Tanrıverdi[1]

University of Cassino and Southern Lazio[1],

May, 2023

MAIA

Cross-Fold Validation Results of Optimized Models

Model	Accuracy	AUC	MCC	MSE	F1	Recall	AP
BayesianRidge	0.583	0.862	0.246	0.173	0.318	0.226	0.890
MLP	0.833	0.828	0.684	0.167	0.845	0.894	0.794
LR	0.852	0.848	0.721	0.148	0.854	0.878	0.825
PLS	0.577	0.906	0.24	0.141	0.275	0.193	0.925
XGB	0.82	0.82	0.663	0.18	0.84	0.881	0.786
RF	0.712	0.709	0.435	0.288	0.731	0.779	0.681
SVC RBF	0.842	0.837	0.705	0.158	0.857	0.931	0.790
KN	0.827	0.826	0.67	0.173	0.828	0.833	0.799
SVC Poly.	0.827	0.822	0.673	0.173	0.84	0.9	0.782
GBM	0.697	0.697	0.414	0.303	0.701	0.721	0.68
SVC Linear	0.843	0.840	0.706	0.157	0.850	0.883	0.801

The excluded models were marked purple.

Model	Accuracy	AUC	MCC	MSE	F1	Recall	AP
Meta-RF	0.846	0.851	0.702	0.154	0.846	0.917	0.759

Ensemble Modeling for Performance Improve
Introduction
Feature Engineering
Modelling
Ensembling
Conclusion
Bayesian Optimization Methods for Datasets with Small-Numbered Features

Conclusion

- Objective: Improve the performance of 11 different regression models on 3 datasets.
- Initial AUC scores 1) 66% 93% 2) 66% 80% 3) 69% 90% .
- Initial MCC scores 1) 33% 77% 2) 7% 48% 3) 25% 72% .
- Challenges: Small dataset with many features, high correlation among models, high numbers of hyperparameters.
- I could stabilize AUC scores to 1) 0.88 2) 0.72 3) 0.85 while corresponding MCC scores are 1) 0.76 2) 0.46 3) 0.7.
- Ensembling the models using meta-modeling significantly improved the overall performance, achieving balanced metrics. This approach provides a generalizable pipeline for similar datasets, demonstrating the effectiveness of ensemble techniques in enhancing performance.

Thank you for your attention! :)