Predicting druggable proteins

In18-S8-CS4742- Bioinformatics

Group: Perceptron
Members: 180572D P.Sarveswarasarma,
180573G T.Sathulakjan,
180196D V.J.V Godfrey,
180630F T.Thakshayan

PROJECT LINK :

Q1.

- **Amino Acid Composition (AAC):** It is a feature that predicts which proteins will be druggable. AAC is determined by counting the number of times each amino acid appears in the protein sequence. This frequency of individual amino acids can be used as a feature, and it has been proven to be effective in predicting druggable proteins. Researchers used it to predict druggable proteins with an accuracy of 80% in a study conducted in 2017. This accuracy outperformed other approaches used to forecast druggable proteins by a wide margin.

- **Aggregated Position-Specific Amino Acid Composition** is referred to as APAAC. It is a characteristic that predicts which proteins will be druggable. The number of times each amino acid appears at each location in the protein sequence is used to calculate APAAC. The machine learning method then makes use of the frequency of each amino acid at each place as a feature.
  It has been demonstrated that the APAAC characteristic is useful for predicting druggable proteins. Researchers employed APAAC to predict druggable proteins with an accuracy of 85% in a 2018 study. This accuracy greatly outperformed those of other approaches used to forecast druggable proteins.

- **CTD** referred to the Comprehensive Protein Characterization Database. Protein sequences, structures, functions, and interactions are all included in the extensive collection of protein data known as CTD. To learn more about the proteins in their dataset, the paper's authors used CTD.
  CTD is a useful tool for scientists trying to comprehend proteins. It gives an in-depth review of protein data that can be utilized to research protein relationships, structure, and function. Additionally useful for drug discovery is CTD. It can be utilized to determine which proteins are druggable and to create fresh medications that target these proteins.

- **Dipeptide Composition is referred to as DPC**. Dipeptide Composition is a characteristic that is used to forecast which proteins will be druggable. It is determined By counting the instances of each dipeptide in the protein sequence. The machine learning method then makes use of the frequency of each dipeptide as a feature.

- **Reduced sequences (RS):** This feature is used to forecast druggable proteins. The protein sequence is broken down into a smaller number of features to calculate RS. A genetic algorithm is the most typical method of reducing a protein sequence. A genetic algorithm is a search algorithm that can be used to find the ideal solution to a problem. In the case of RS, the challenge is to determine the most effective method for reducing the protein sequence while preserving the data required to predict druggable proteins.

The genetic algorithm randomly creates a population of RS. Then a machine learning algorithm is then used to assess each RS in that population. The best-performing RS are then used to produce a new population of RS. This process is repeated until the optimal RS is found

Few of the RS features are,

RSsecond: It is a feature for predicting druggable proteins. It can be calculated by simulating the protein's secondary structure.

RSDHP: RSDHP is an acronym for Reduced Sequence Distribution of D, H, and P. It is a characteristic that is employed to forecast proteins that will be druggable. It can be calculated by simulating the distribution of D, H, and P amino acids within the protein.

RSacid. RSacid is an acronym for Reduced Sequence Acidity. This characteristic is employed to forecast which proteins will be druggable. It can be calculated by using the protein's acidity as a model.

RSpolar: RSpolar is an acronym for Reduced Sequence Polarity. It is a characteristic that predicts which proteins will be druggable. It can be calculated by simulating the protein's polarity.

RScharge: RScharge is an acronym for Reduced Sequence Charge. It is a characteristic that predicts which proteins will be druggable. Then the charge of the protein is used to calculate it.

Q2)

Four different Machine learning models were used, and the effectiveness of those machine learning models is assessed through cross-validation and multiple evaluation metrics. The best model is chosen by taking AUC as an evaluation metric. The detailed process involves the following steps,

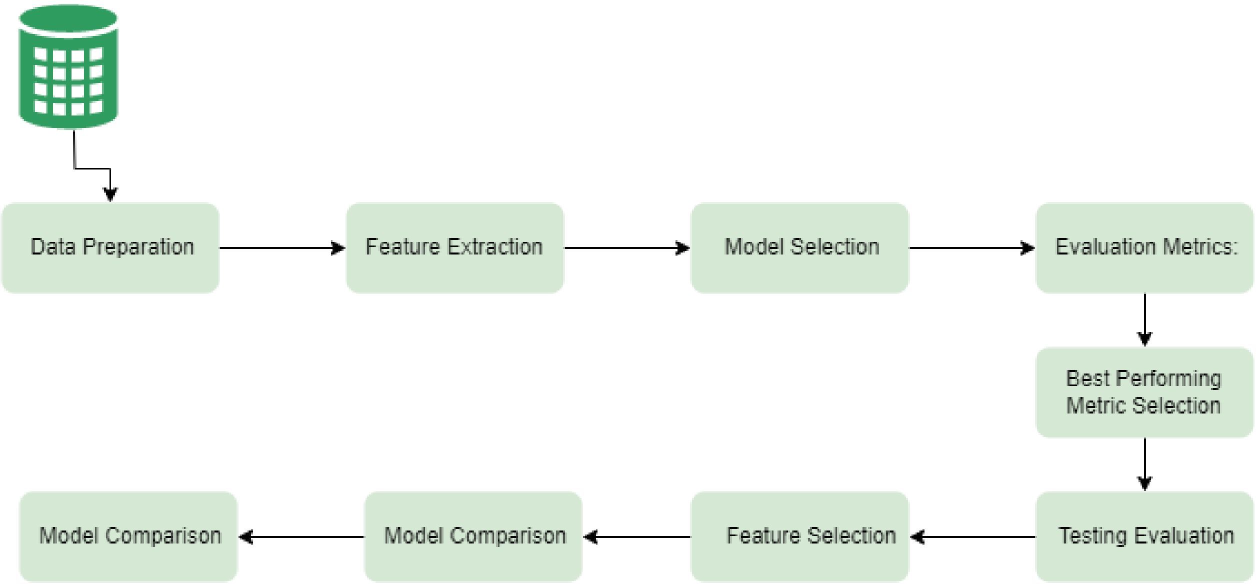


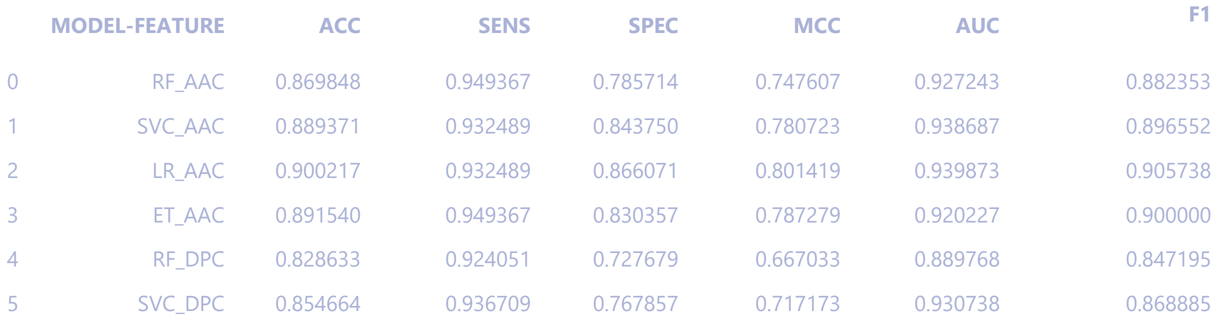
Figure 1 Process diagram

Data Preparation: The positive and negative testing and training data samples are combined to create a single dataset containing both positive and negative samples.

Feature Extraction: The features were obtained from the FASTA files through feature extraction. These features capture important patterns within the primary amino acid components. The extracted features were rescaled using Min-Max Scaling to ensure consistency and comparability.

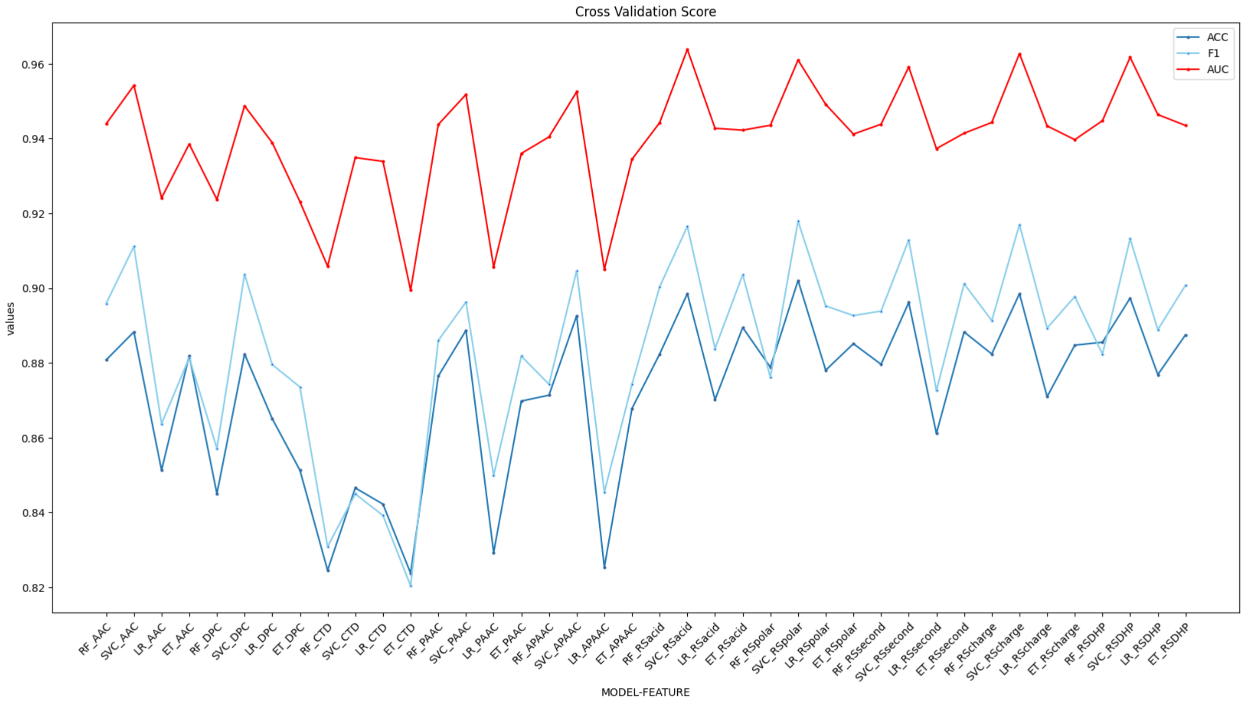
Model Selection: Four machine learning models, Random Forest, Support Vector Machines (SVM), Logistic Regression, and Extra Trees were chosen for evaluation. These models represent a variety of algorithmic approaches that can be used for different approaches on the given dataset.

Cross-Validation: To assess the performance of each model, 5-fold cross-validation is employed. Then all four models were then trained and evaluated using each fold serving as the test set once while the remaining folds act as the training set.



	MODEL-FEATURE	ACC	SENS	SPEC	MCC	AUC	F1
0	RF_AAC	0.869848	0.949367	0.785714	0.747607	0.927243	0.882353
1	SVC_AAC	0.889371	0.932489	0.843750	0.780723	0.938687	0.896552
2	LR_AAC	0.900217	0.932489	0.866071	0.801419	0.939873	0.905738
3	ET_AAC	0.891540	0.949367	0.830357	0.787279	0.920227	0.900000
4	RF_DPC	0.828633	0.924051	0.727679	0.667033	0.889768	0.847195
5	SVC_DPC	0.854664	0.936709	0.767857	0.717173	0.930738	0.868885

	MODEL-FEATURE	ACC	SENS	SPEC	MCC	AUC	F1
6	LR_DPC	0.880694	0.894515	0.866071	0.761232	0.937255	0.885177
7	ET_DPC	0.815618	0.919831	0.705357	0.642485	0.885407	0.836852
8	RF_CTD	0.759219	0.843882	0.669643	0.522713	0.839634	0.782779
9	SVC_CTD	0.839479	0.860759	0.816964	0.678799	0.905553	0.846473
10	LR_CTD	0.839479	0.848101	0.830357	0.678650	0.897340	0.844538
11	ET_CTD	0.767896	0.856540	0.674107	0.541083	0.837656	0.791423
12	RF_PAAC	0.874187	0.945148	0.799107	0.754559	0.903726	0.885375
13	SVC_PAAC	0.882863	0.949367	0.812500	0.771296	0.943942	0.892857
14	LR_PAAC	0.800434	0.936709	0.656250	0.620768	0.911449	0.828358
15	ET_PAAC	0.874187	0.962025	0.781250	0.758576	0.902925	0.887160
16	RF_APAAC	0.872017	0.945148	0.794643	0.750600	0.903688	0.883629
17	SVC_APAAC	0.878525	0.953586	0.799107	0.764284	0.939525	0.889764
18	LR_APAAC	0.761388	0.924051	0.589286	0.547614	0.894063	0.799270
19	ET_APAAC	0.859002	0.936709	0.776786	0.725059	0.894976	0.872299
20	RF_RSacid	0.869848	0.940928	0.794643	0.745755	0.928298	0.881423
21	SVC_RSacid	0.891540	0.932489	0.848214	0.784839	0.938611	0.898374
22	LR_RSacid	0.898048	0.924051	0.870536	0.796569	0.936200	0.903093
23	ET_RSacid	0.876356	0.945148	0.803571	0.758527	0.913888	0.887129
24	RF_RSpolar	0.869848	0.953586	0.781250	0.748628	0.930521	0.882812
25	SVC_RSpolar	0.885033	0.932489	0.834821	0.772523	0.938574	0.892929
26	LR_RSpolar	0.895879	0.932489	0.857143	0.793106	0.933186	0.902041
27	ET_RSpolar	0.887202	0.953586	0.816964	0.780087	0.917317	0.896825
28	RF_RSsecond	0.872017	0.949367	0.790179	0.751533	0.928854	0.884086
29	SVC_RSsecond	0.893709	0.940928	0.843750	0.790009	0.941588	0.901010
30	LR_RSsecond	0.898048	0.919831	0.875000	0.796318	0.943660	0.902692
31	ET_RSsecond	0.876356	0.936709	0.812500	0.756948	0.919812	0.886228
32	RF_RScharge	0.859002	0.936709	0.776786	0.725059	0.924729	0.872299
33	SVC_RScharge	0.889371	0.936709	0.839286	0.781266	0.938348	0.896970
34	LR_RScharge	0.898048	0.924051	0.870536	0.796569	0.935692	0.903093
35	ET_RScharge	0.869848	0.940928	0.794643	0.745755	0.912476	0.881423
36	RF_RSDHP	0.880694	0.945148	0.812500	0.766492	0.928129	0.890656
37	SVC_RSDHP	0.878525	0.932489	0.821429	0.760304	0.936709	0.887550
38	LR_RSDHP	0.895879	0.924051	0.866071	0.792364	0.933130	0.901235
39	ET_RSDHP	0.880694	0.940928	0.816964	0.765721	0.920170	0.890220



Evaluation Metrics and Metric Selection: Models are assessed using a variety of performance metrics throughout each cross-validation iteration. Accuracy, sensitivity, specificity, Matthew's correlation coefficient (MCC), area under the receiver operating characteristic curve (AUC), and F1 score are among the metrics that were recorded. These metrics explain how well the models handle imbalanced data, correctly classify instances, and perform. The best-performing metric was selected based on the AUC used for testing.

Testing Evaluation: The selected model is assessed using the selected metrics and tested on the test dataset. Accuracy, sensitivity, specificity, MCC, AUC, and F1 score were measured.

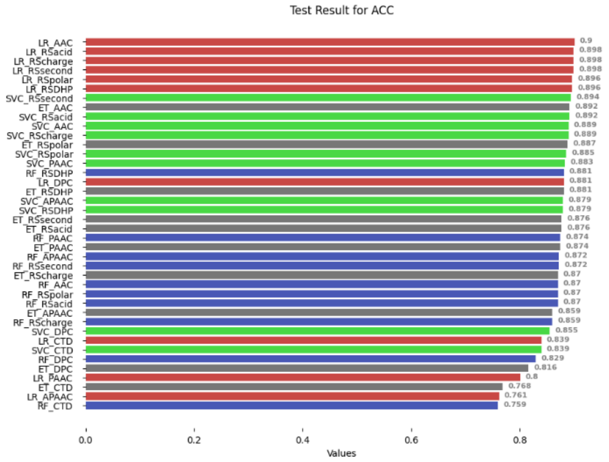


Figure 2 Accuracy of Test data

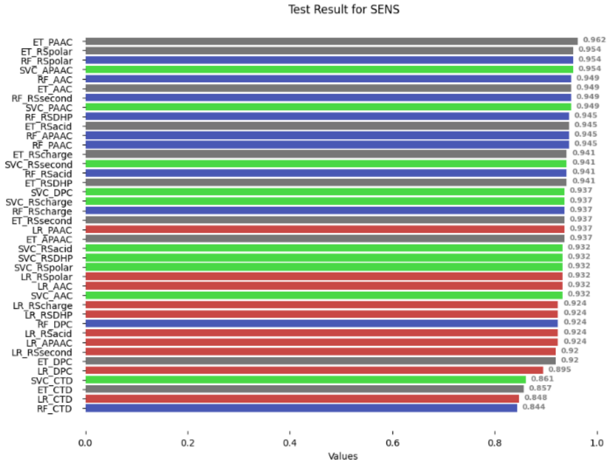


Figure 3 Sensitivity of Test data

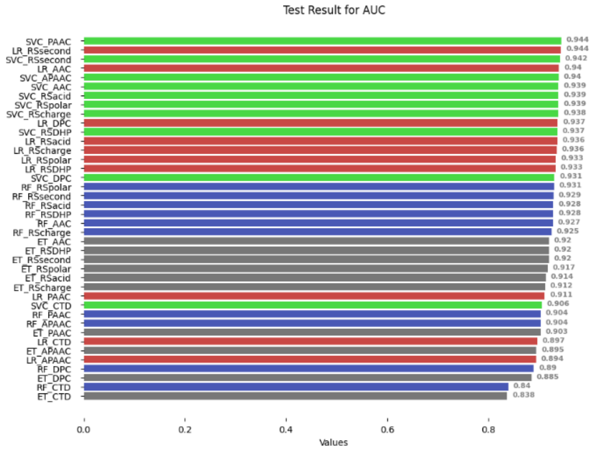


Figure 4 AUC of Test data

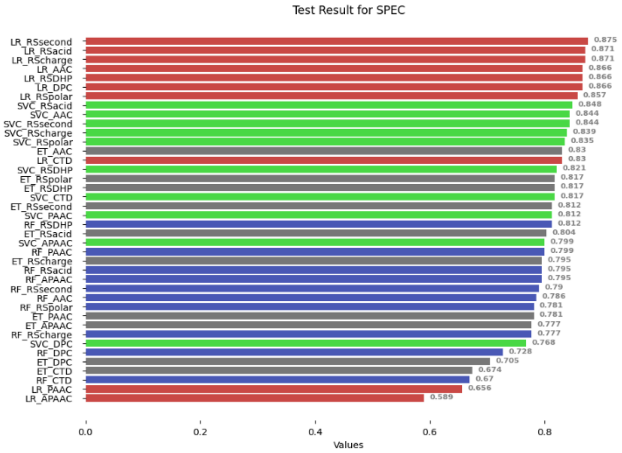


Figure 5 Specificity of Test data

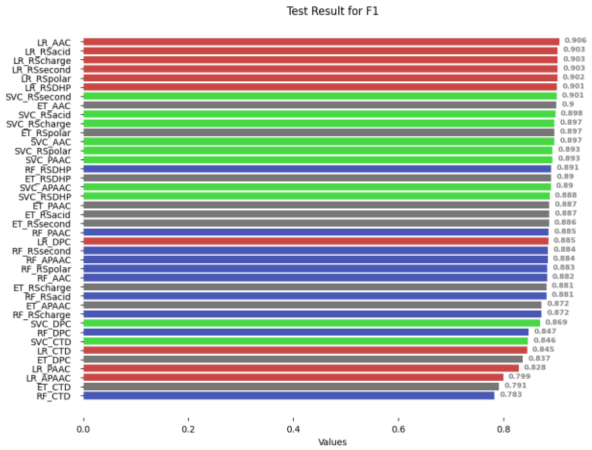


Figure 6 F1 measure of Test data

Model Comparison: By comparing the performance metrics of the various models and choosing the one that performs best in terms of the AUC evaluation metric, SVM outperformed the other models based on the most of the features.

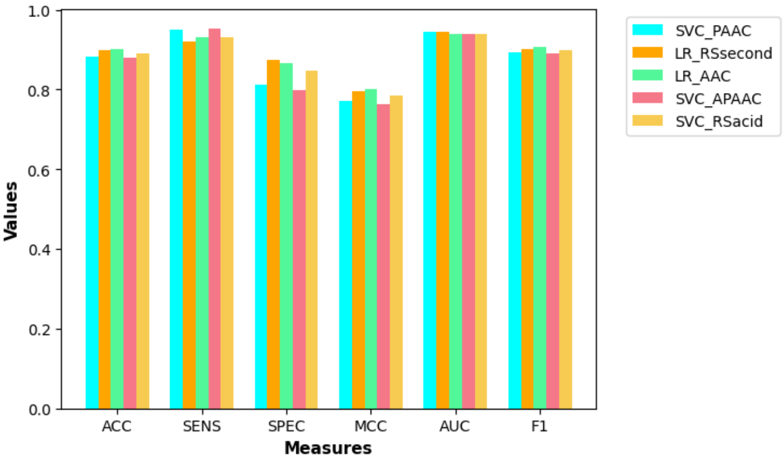


Figure 7 Best performed model of top 5 feature and their respective classifiers

In conclusion, the described procedure entails combining positive and negative samples, training and assessing four different machine learning models using cross-validation, recording various evaluation metrics, choosing the best-performing metric, evaluating the performance of the chosen model on a testing dataset, choosing five features, and concluding that SVM outperformed the other models.

Q3)

It seems like you have described a multi-step process for building a stratified classifier with **hyperparameter tuning, stacking, and a meta-learner**. Here's a breakdown of the steps:

1. Stratified Classifier and 10-fold Cross-Validation:
   - To ensure that each fold has a comparable distribution of classes, the dataset is partitioned into folds using stratified sampling.
   - Using the optimized hyperparameters received from Optuna (a hyperparameter optimization library), a classifier model is trained on each fold.
   - By computing the evaluation metric (such as accuracy or F1 score) on the validation set for each fold, cross_val_score estimates the performance of each model.

2. Hyperparameter Tuning with Optuna:
   - The model's hyperparameters are optimized using Optuna.
   - Cross_val_score, which measures the effectiveness of various hyperparameter configurations using cross-validation, is used to optimize the hyperparameters.
   - Based on the evaluation metric, the top hyperparameters are chosen and applied in the following steps.

3. Stacking Classifier:
   - In a stacking ensemble, the tuned models from the previous step are joined.
   - A meta-learner uses the predictions from each individual model as features.
   - Another classifier that uses the stacked predictions as input to learn how to create final predictions is the meta learner.

4. Meta Learner Training:
   - The meta-learner is trained using the stacked predictions from the previous step as input and the true labels as the target.
   - The meta-learner learns to make predictions based on the combined predictions of the individual models.
5. Testing:
   - Finally, the trained meta-learner is used to make predictions on the test samples.
   - The performance of the model is evaluated on the test set using appropriate evaluation metrics.

Overall, this method makes use of hyperparameter tunning, classifier stacking, and a meta-learner to create a more potent and reliable classifier. It combinations the outcomes of various models to enhance performance as a whole.

Q4)

After hyperparameter tuning, an ensemble model becomes more generalized compares to individual models due to its ability to combine the strengths and reduced the weaknesses of different models.

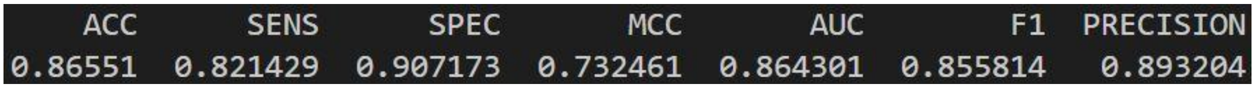
Ensemble models are composed of multiple hyperparameter-tuned models, which were optimized on different subsets of the feature. This process involved adjusting various settings, such as the number of estimators, regularization parameters, or maximum depths, to find the best configuration for each model. Due to hyperparameter tuning, each base model becomes more optimized and capable of capturing different aspects or patterns in the data.

By combining these models. The ensemble model can able to capture a more thorough understanding of the data and produce more precise predictions by combining the predictions of these models.

The ensemble model then utilizes a particular aggregation technique, to combine the predictions from the individual models. A more generalized model is produced, as a result of this combination's ability to reduce overfitting and smooth out individual model biases. Ensemble models also became less sensitive to noise and outliers compared to individual models. Ensemble models perform better by enhancing the individual base models and utilizing their combined knowledge. In comparison to individual models in Q2, the ensemble model in Q3 has improved in robustness, ability to generalize well to new data, and predictive performance.

MODEL_FEATURE	ACC	SENS	SPEC	MCC	AUC	F1	PRECISION
SVC_AAC	0.882863	0.932489	0.830357	0.768440	0.937519	0.873239	0.920792
SVC_DPC	0.744035	0.898734	0.580357	0.507676	0.834925	0.687831	0.844156
LR_CTD	0.885033	0.907173	0.861607	0.770218	0.921414	0.879271	0.897674
RF_PAAC	0.872017	0.932489	0.808036	0.748175	0.911656	0.859857	0.918782
RF_APAAC	0.863341	0.928270	0.794643	0.731351	0.908077	0.849642	0.912821
ET_RSacid	0.865510	0.928270	0.799107	0.735372	0.928157	0.852381	0.913265
SVC_RSpolar	0.819957	0.932489	0.700893	0.653800	0.886377	0.790932	0.907514
SVC_RSsecond	0.895879	0.932489	0.857143	0.793106	0.937735	0.888889	0.923077
RF_RScharge	0.878525	0.945148	0.808036	0.762505	0.931058	0.866029	0.932990
SVC_RSDHP	0.837310	0.852321	0.821429	0.674321	0.885925	0.830700	0.840183

Table 1 All Scores after tuning



```
    ACC      SENS      SPEC       MCC       AUC        F1  PRECISION
0.86551  0.821429  0.907173  0.732461  0.864301  0.855814   0.893204
```

Table 2 After Ensembling the score of meta-learner.

****