Revisiting the Connection Between Gut Microbiota and Depressive Disorder: An Evaluation and Reproduction of Machine Learning Approaches

Michael Wurth, CS:4980 at the University of Iowa

Abstract

Research into the human gut microbiome has increasingly revealed associations with various psychiatric disorders, suggesting a link between microbial composition and mental health outcomes. Angelova et al.[2] investigated these associations by training machine learning classifiers on metagenomic sequence read data, finding that the YOLOv8 (You Only Look Once) convolutional neural network (CNN) was most effective in classifying depressive states based on gut microbiome composition.

Our study aims to revisit and extend Angelova et al.'s findings by evaluating the performance of logistic regression, random forest, and support vector machines (SVMs). Additionally, we assess whether these models can achieve performance comparable to YOLOv8 under similar data conditions and parameter optimization.

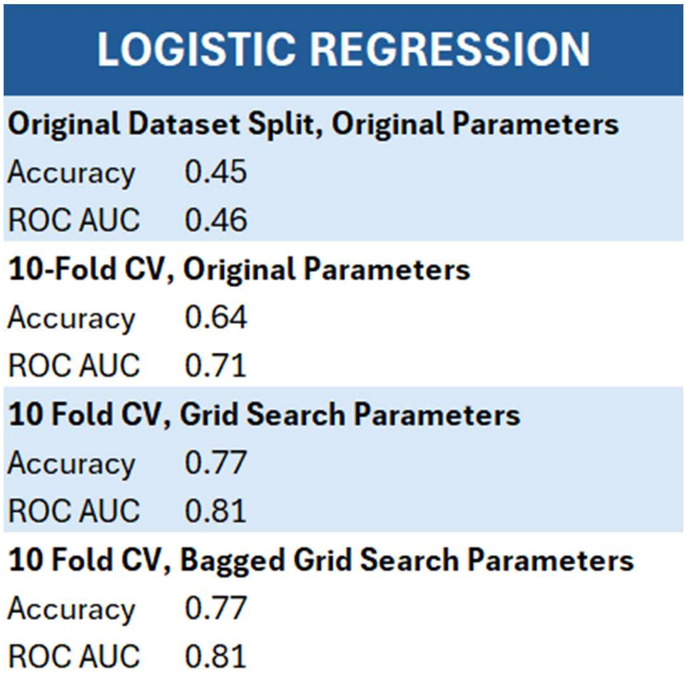
Using the datasets from the reference study, we re-examined model efficacy before and after parameter tuning and cross-validation. Logistic regression notably outperformed previous reports, achieving a cross-validated accuracy of 0.74 and a ROC AUC of 0.81. On their synthetically expanded dataset, random forest achieved near-perfect classification. SVM models attained perfect accuracy on this dataset, matching YOLOv8's performance as reported by Angelova et al. These findings suggest that simpler models can achieve competitive results on metagenomic data when conditions are controlled and classifier optimization is appropriately applied.

Key Words: [Depression, gut microbiome, machine learning, logistic regression, psychiatric disorder]

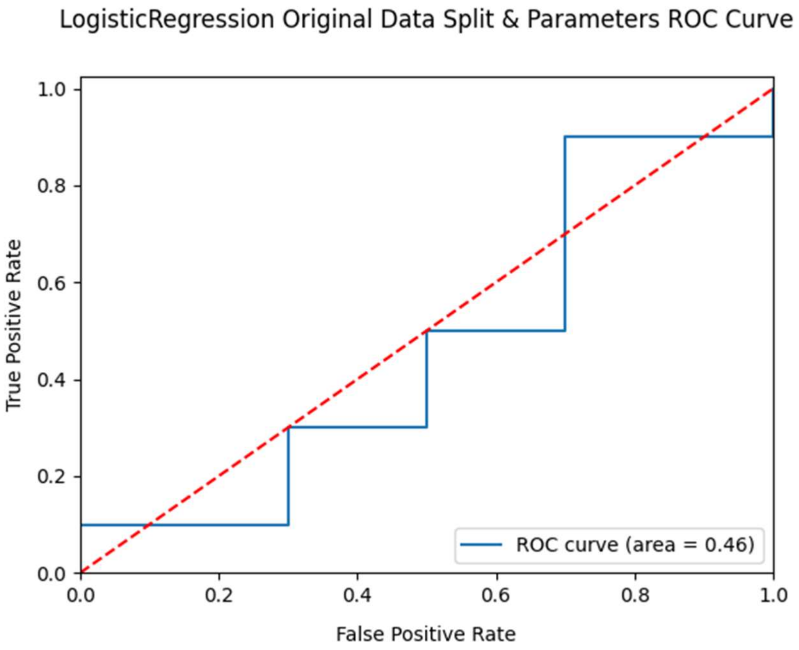
Results:

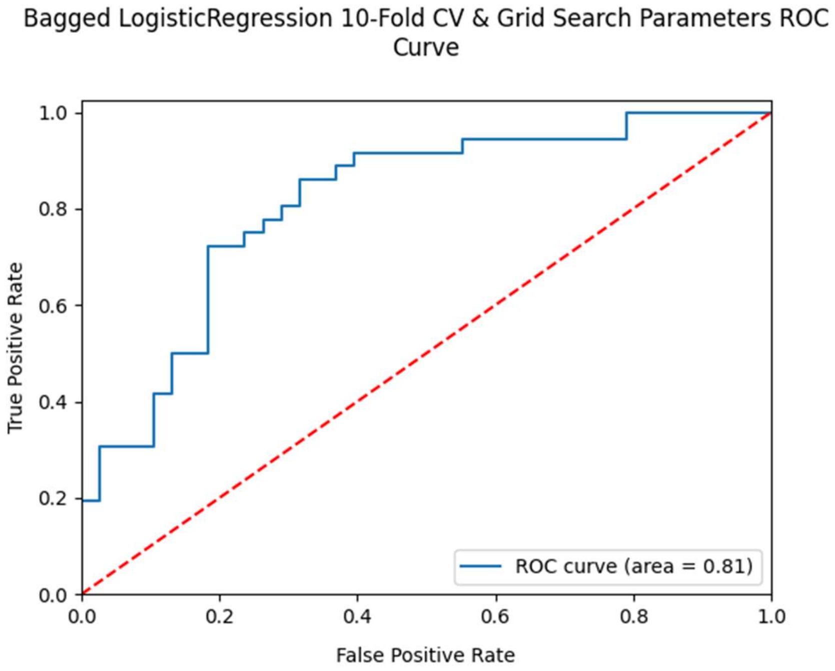
1. Classical Model Comparative Analysis:

1.a. Logistic Regression

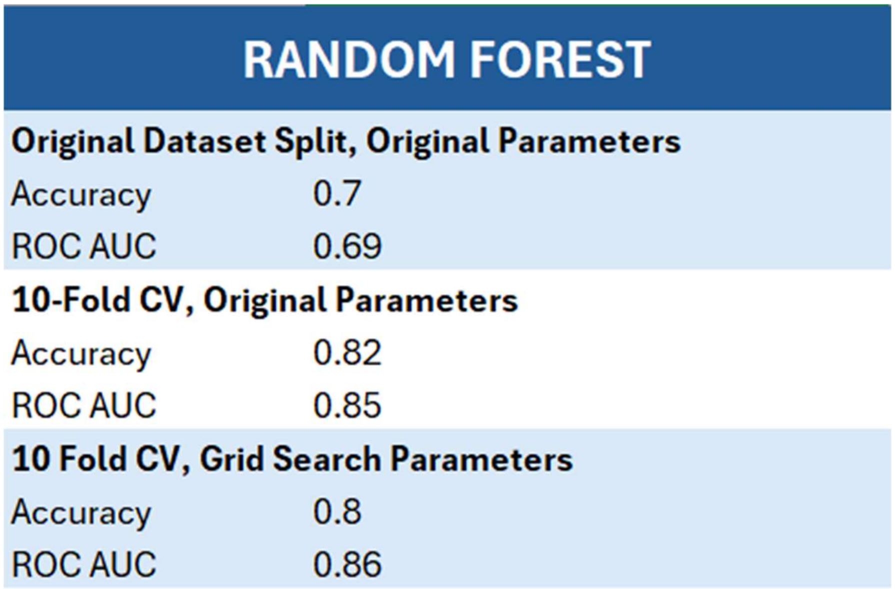


LOGISTIC REGRESSION	
Original Dataset Split, Original Parameters	
Accuracy	0.45
ROC AUC	0.46
10-Fold CV, Original Parameters	
Accuracy	0.64
ROC AUC	0.71
10 Fold CV, Grid Search Parameters	
Accuracy	0.77
ROC AUC	0.81
10 Fold CV, Bagged Grid Search Parameters	
Accuracy	0.77
ROC AUC	0.81

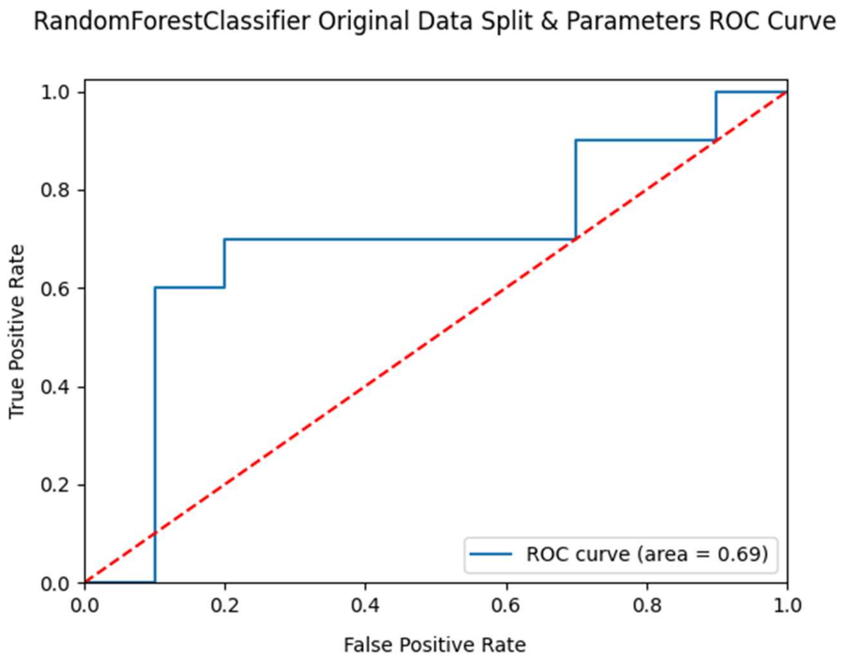


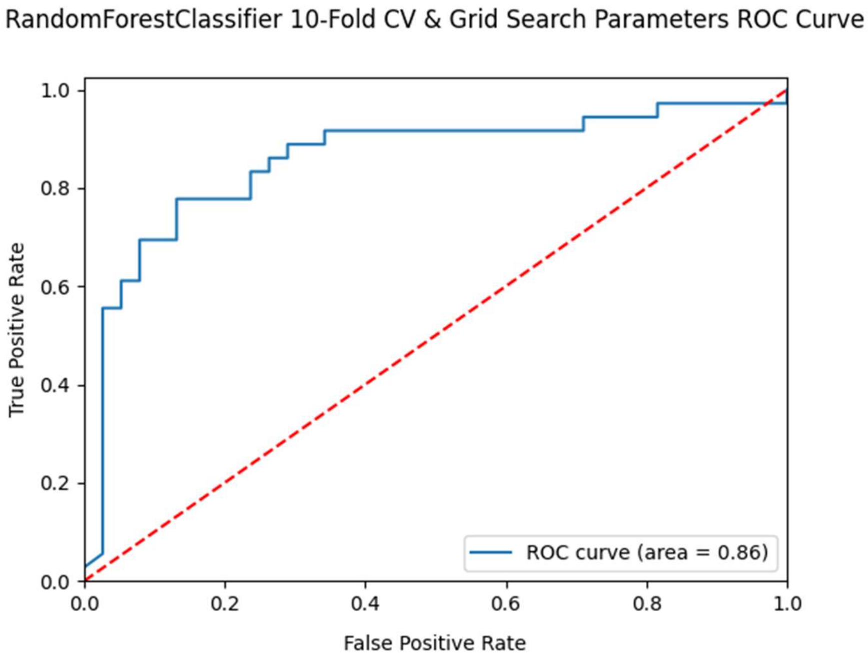


1.b. Random Forest



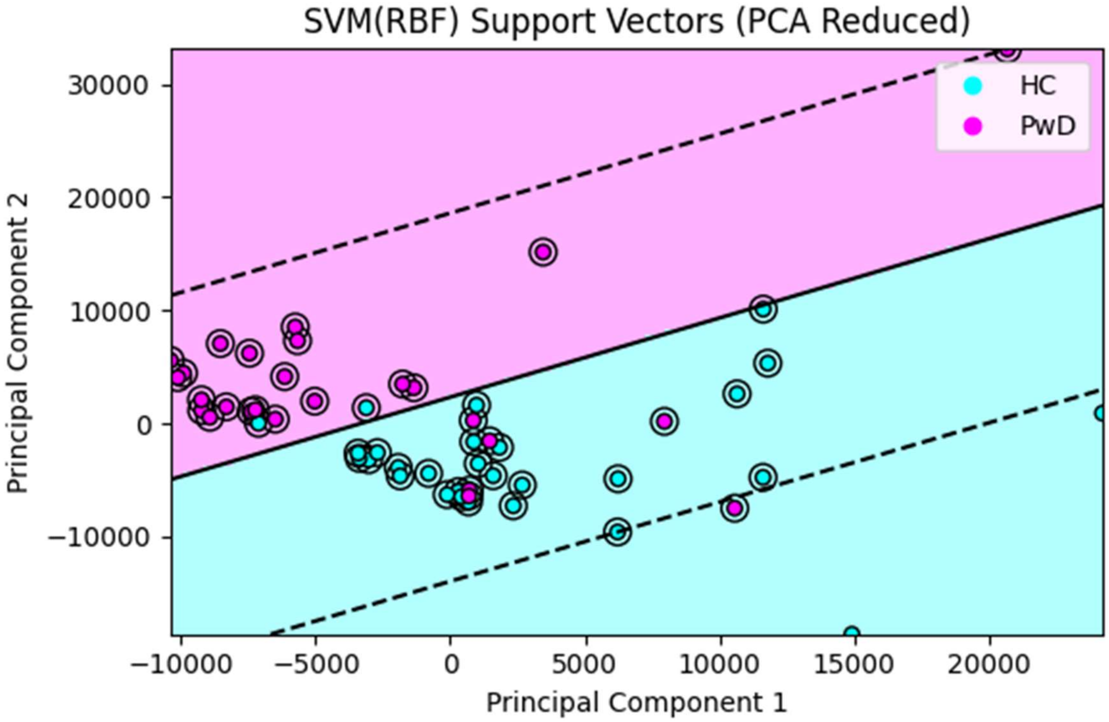
RANDOM FOREST	
Original Dataset Split, Original Parameters	
Accuracy	0.7
ROC AUC	0.69
10-Fold CV, Original Parameters	
Accuracy	0.82
ROC AUC	0.85
10 Fold CV, Grid Search Parameters	
Accuracy	0.8
ROC AUC	0.86

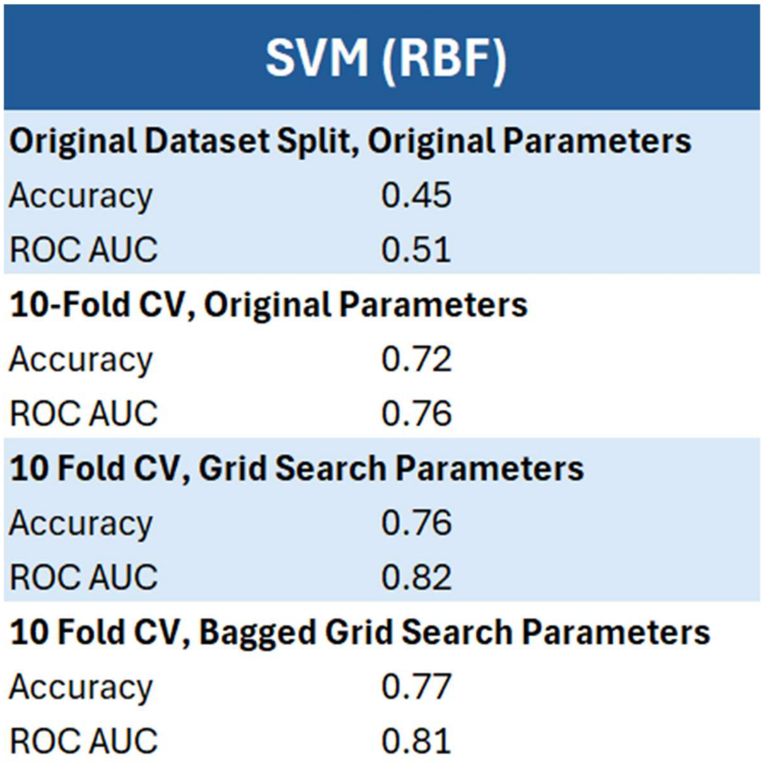

RandomForestClassifier Original Data Split & Parameters ROC Curve


RandomForestClassifier 10-Fold CV & Grid Search Parameters ROC Curve

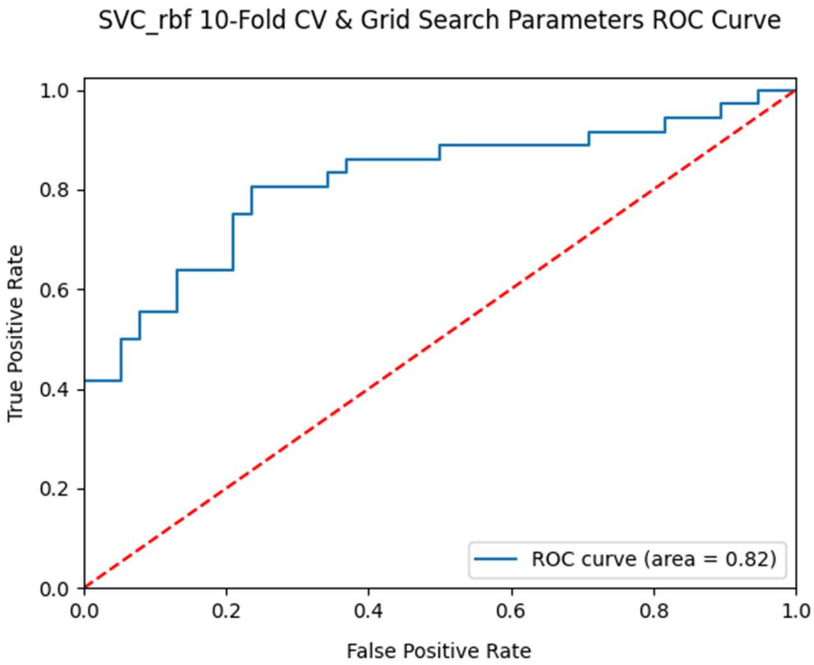
2. Support Vector Machine (SVM) Performance on Metagenomic Signature Data:

2.a. Radial Basis Function Kernel



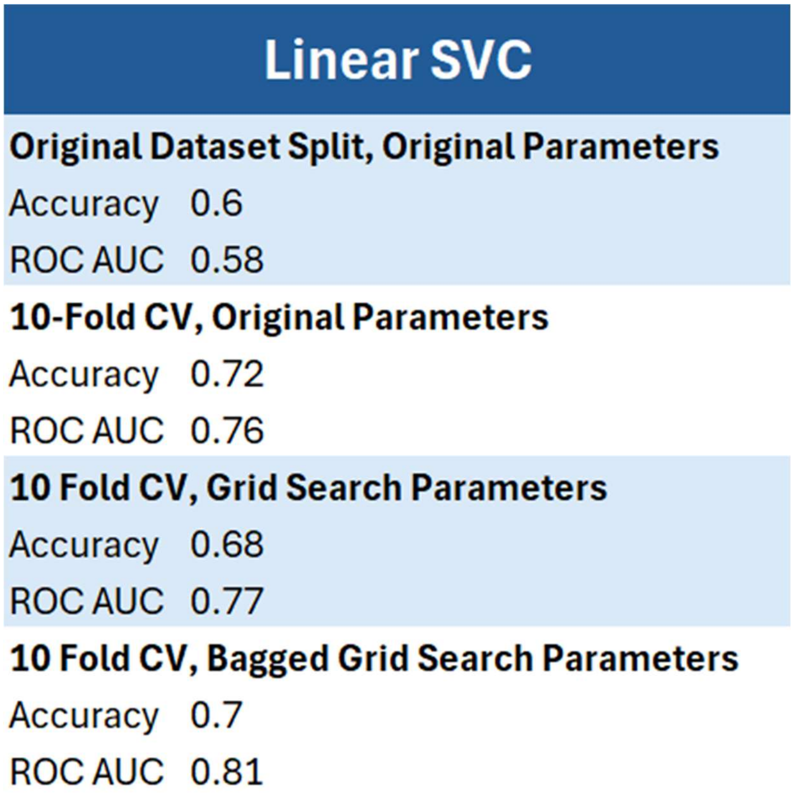


SVM (RBF)	
Original Dataset Split, Original Parameters	
Accuracy	0.45
ROC AUC	0.51
10-Fold CV, Original Parameters	
Accuracy	0.72
ROC AUC	0.76
10 Fold CV, Grid Search Parameters	
Accuracy	0.76
ROC AUC	0.82
10 Fold CV, Bagged Grid Search Parameters	
Accuracy	0.77
ROC AUC	0.81

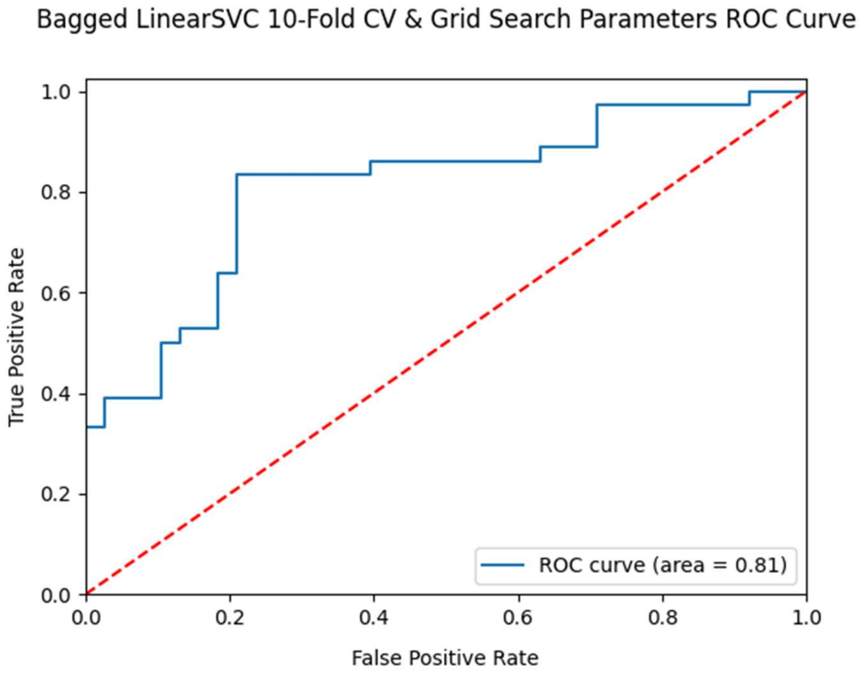


SVC_rbf 10-Fold CV & Grid Search Parameters ROC Curve

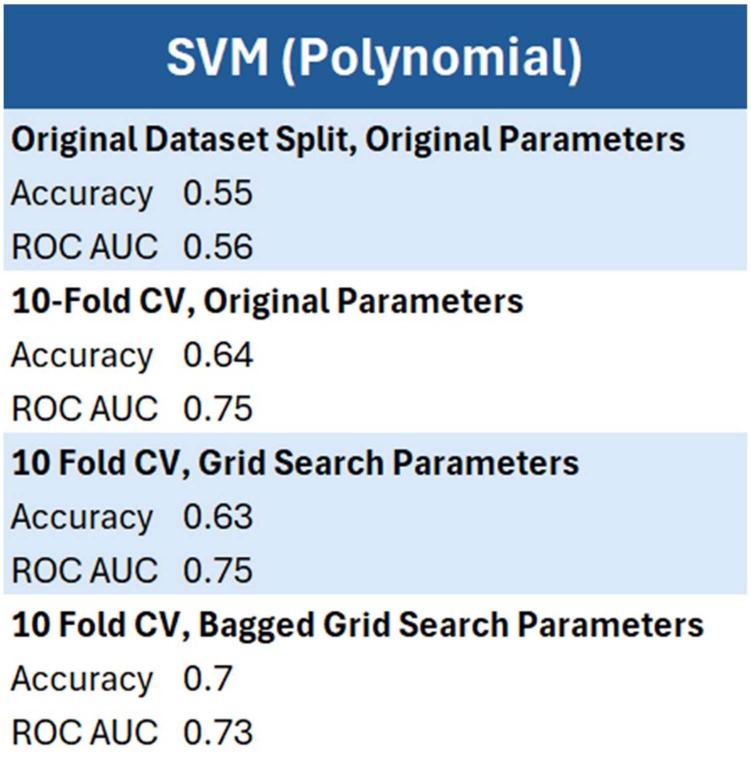
2.b. Linear SVC



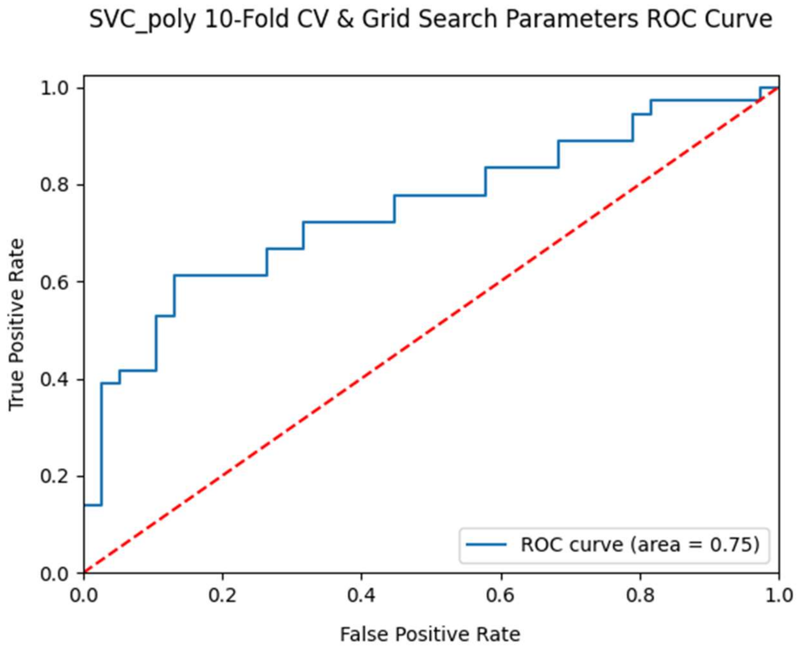
Linear SVC	
Original Dataset Split, Original Parameters	
Accuracy	0.6
ROC AUC	0.58
10-Fold CV, Original Parameters	
Accuracy	0.72
ROC AUC	0.76
10 Fold CV, Grid Search Parameters	
Accuracy	0.68
ROC AUC	0.77
10 Fold CV, Bagged Grid Search Parameters	
Accuracy	0.7
ROC AUC	0.81



2.c. Polynomial Kernel

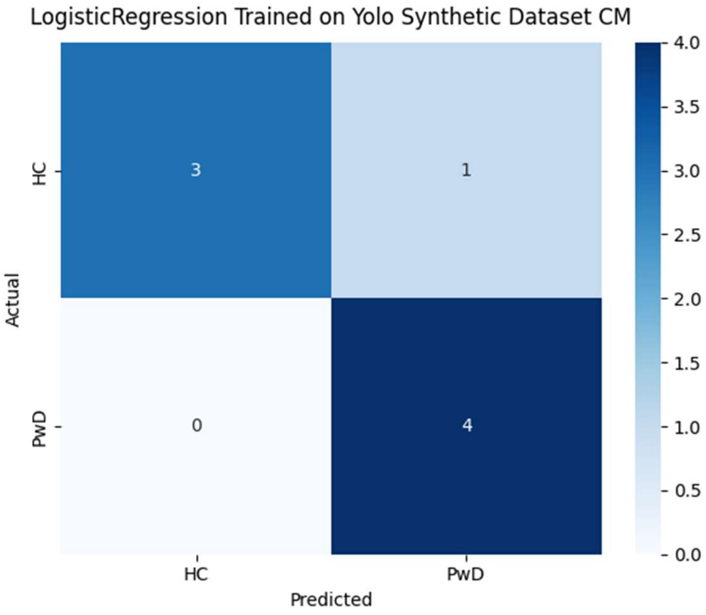


SVM (Polynomial)	
Original Dataset Split, Original Parameters	
Accuracy	0.55
ROC AUC	0.56
10-Fold CV, Original Parameters	
Accuracy	0.64
ROC AUC	0.75
10 Fold CV, Grid Search Parameters	
Accuracy	0.63
ROC AUC	0.75
10 Fold CV, Bagged Grid Search Parameters	
Accuracy	0.7
ROC AUC	0.73



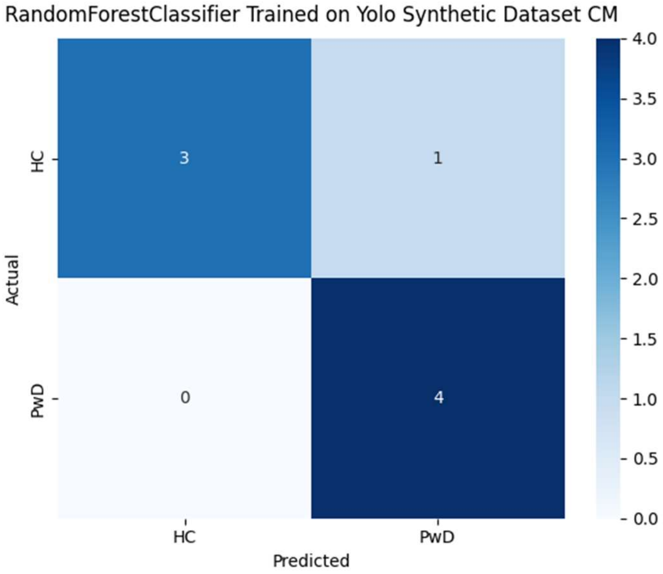
3. YOLO Comparison - Using Synthetic Data:

3.a. Logistic Regression



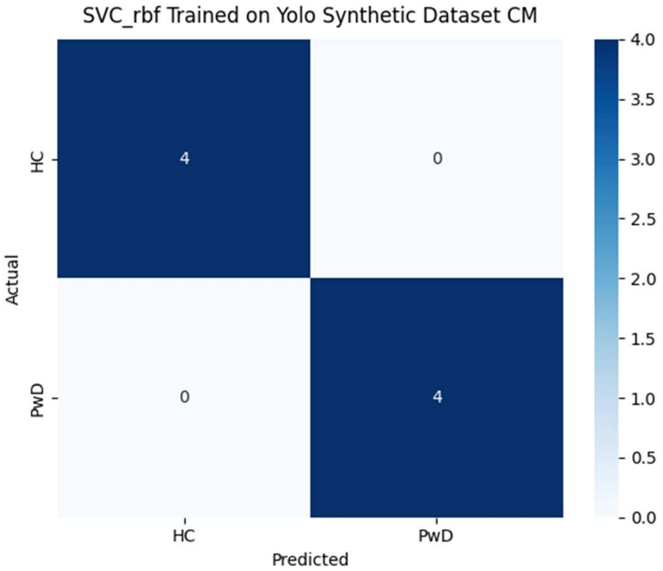
Logistic Regression performed promisingly when trained on the synthetic dataset, misclassifying only one data point. It predicted one individual had depression when they were in fact a healthy control.

3.b. Random Forest



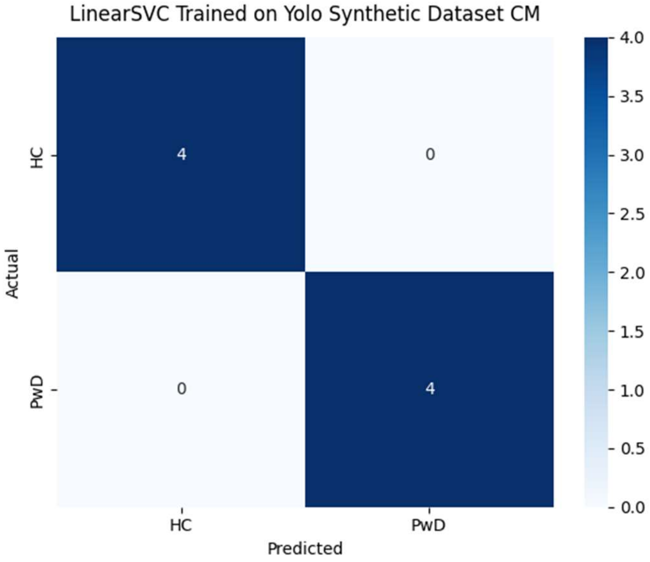
Random Forest also performed promisingly when trained on the synthetic dataset. It also failed to correctly classify one data point. Like logistic regression, it predicted one healthy control to have depression.

3.c. SVM (Radial Basis Function)



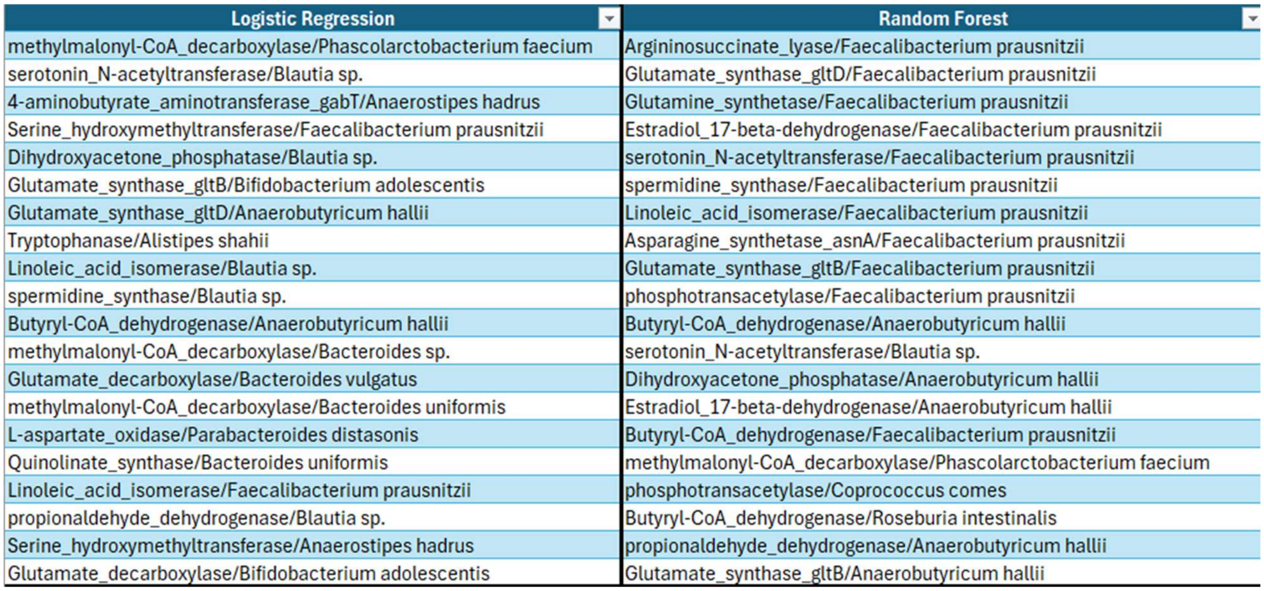
The radial basis function SVM performed astoundingly well, achieving 100% accuracy on the yolo holdout test set.

3.d. LinearSVC

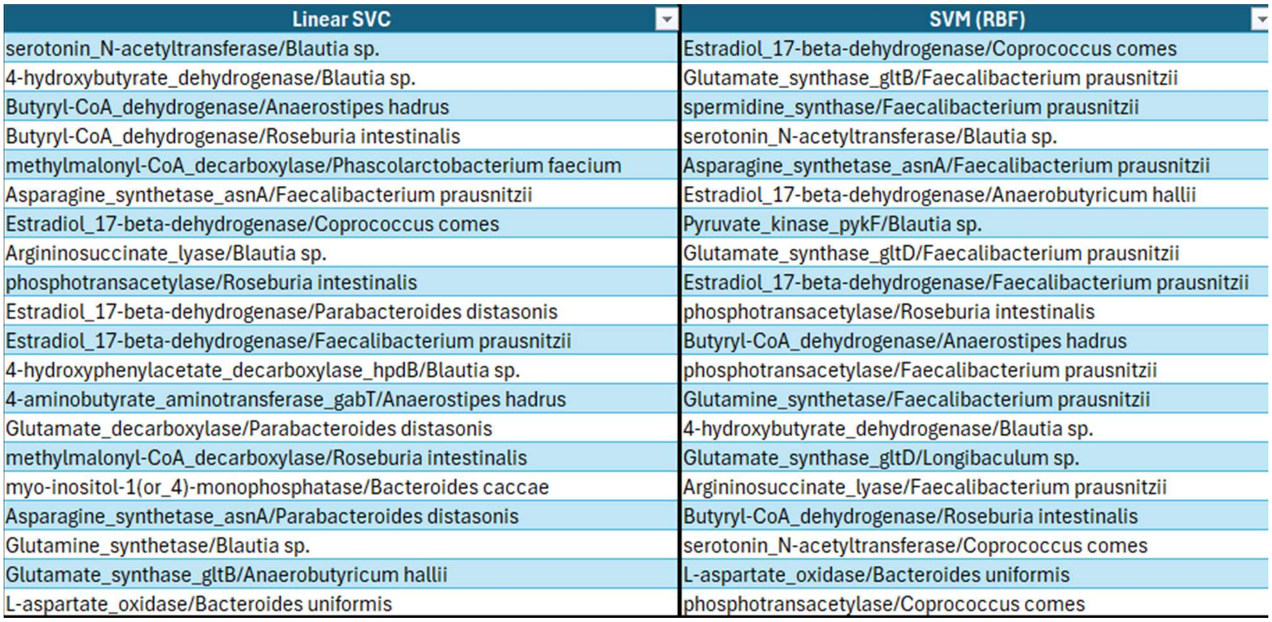


The LinearSVC model additionally predicted with perfect accuracy when trained on the yolo synthetic data.

4. Feature Importance Comparative Analysis:



Logistic Regression	Random Forest
methylmalonyl-CoA_decarboxylase/Phascolarctobacterium faecium	Argininosuccinate_lyase/Faecalibacterium prausnitzii
serotonin_N-acetyltransferase/Blautia sp.	Glutamate_synthase_gltD/Faecalibacterium prausnitzii
4-aminobutyrate_aminotransferase_gabT/Anaerostipes hadrus	Glutamine_synthetase/Faecalibacterium prausnitzii
Serine_hydroxymethyltransferase/Faecalibacterium prausnitzii	Estradiol_17-beta-dehydrogenase/Faecalibacterium prausnitzii
Dihydroxyacetone_phosphatase/Blautia sp.	serotonin_N-acetyltransferase/Faecalibacterium prausnitzii
Glutamate_synthase_gltB/Bifidobacterium adolescentis	spermidine_synthase/Faecalibacterium prausnitzii
Glutamate_synthase_gltD/Anaerobutyricum hallii	Linoleic_acid_isomerase/Faecalibacterium prausnitzii
Tryptophanase/Alistipes shahii	Asparagine_synthetase_asnA/Faecalibacterium prausnitzii
Linoleic_acid_isomerase/Blautia sp.	Glutamate_synthase_gltB/Faecalibacterium prausnitzii
spermidine_synthase/Blautia sp.	phosphotransacetylase/Faecalibacterium prausnitzii
Butyryl-CoA_dehydrogenase/Anaerobutyricum hallii	Butyryl-CoA_dehydrogenase/Anaerobutyricum hallii
methylmalonyl-CoA_decarboxylase/Bacteroides sp.	serotonin_N-acetyltransferase/Blautia sp.
Glutamate_decarboxylase/Bacteroides vulgatus	Dihydroxyacetone_phosphatase/Anaerobutyricum hallii
methylmalonyl-CoA_decarboxylase/Bacteroides uniformis	Estradiol_17-beta-dehydrogenase/Anaerobutyricum hallii
L-aspartate_oxidase/Parabacteroides distasonis	Butyryl-CoA_dehydrogenase/Faecalibacterium prausnitzii
Quinolinate_synthase/Bacteroides uniformis	methylmalonyl-CoA_decarboxylase/Phascolarctobacterium faecium
Linoleic_acid_isomerase/Faecalibacterium prausnitzii	phosphotransacetylase/Coprococcus comes
propionaldehyde_dehydrogenase/Blautia sp.	Butyryl-CoA_dehydrogenase/Roseburia intestinalis
Serine_hydroxymethyltransferase/Anaerostipes hadrus	propionaldehyde_dehydrogenase/Anaerobutyricum hallii
Glutamate_decarboxylase/Bifidobacterium adolescentis	Glutamate_synthase_gltB/Anaerobutyricum hallii



Linear SVC	SVM (RBF)
serotonin_N-acetyltransferase/Blautia sp.	Estradiol_17-beta-dehydrogenase/Coprococcus comes
4-hydroxybutyrate_dehydrogenase/Blautia sp.	Glutamate_synthase_gltB/Faecalibacterium prausnitzii
Butyryl-CoA_dehydrogenase/Anaerostipes hadrus	spermidine_synthase/Faecalibacterium prausnitzii
Butyryl-CoA_dehydrogenase/Roseburia intestinalis	serotonin_N-acetyltransferase/Blautia sp.
methylmalonyl-CoA_decarboxylase/Phascolarctobacterium faecium	Asparagine_synthetase_asnA/Faecalibacterium prausnitzii
Asparagine_synthetase_asnA/Faecalibacterium prausnitzii	Estradiol_17-beta-dehydrogenase/Anaerobutyricum hallii
Estradiol_17-beta-dehydrogenase/Coprococcus comes	Pyruvate_kinase_pykF/Blautia sp.
Argininosuccinate_lyase/Blautia sp.	Glutamate_synthase_gltD/Faecalibacterium prausnitzii
phosphotransacetylase/Roseburia intestinalis	Estradiol_17-beta-dehydrogenase/Faecalibacterium prausnitzii
Estradiol_17-beta-dehydrogenase/Parabacteroides distasonis	phosphotransacetylase/Roseburia intestinalis
Estradiol_17-beta-dehydrogenase/Faecalibacterium prausnitzii	Butyryl-CoA_dehydrogenase/Anaerostipes hadrus
4-hydroxyphenylacetate_decarboxylase_hpdB/Blautia sp.	phosphotransacetylase/Faecalibacterium prausnitzii
4-aminobutyrate_aminotransferase_gabT/Anaerostipes hadrus	Glutamine_synthetase/Faecalibacterium prausnitzii
Glutamate_decarboxylase/Parabacteroides distasonis	4-hydroxybutyrate_dehydrogenase/Blautia sp.
methylmalonyl-CoA_decarboxylase/Roseburia intestinalis	Glutamate_synthase_gltD/Longibaculum sp.
myo-inositol-1(or_4)-monophosphatase/Bacteroides caccae	Argininosuccinate_lyase/Faecalibacterium prausnitzii
Asparagine_synthetase_asnA/Parabacteroides distasonis	Butyryl-CoA_dehydrogenase/Roseburia intestinalis
Glutamine_synthetase/Blautia sp.	serotonin_N-acetyltransferase/Coprococcus comes
Glutamate_synthase_gltB/Anaerobutyricum hallii	L-aspartate_oxidase/Bacteroides uniformis
L-aspartate_oxidase/Bacteroides uniformis	phosphotransacetylase/Coprococcus comes

Discussion

Our first noteworthy finding is the drastic increase in the performance of logistic regression on metagenomic signature data when appropriately tuned and evaluated. Angelova et al. reported a modest accuracy of 0.55 for logistic regression on a single hold-out test with an elastic net penalty. Our results suggest that elastic net may not be an ideal penalty parameter for this type of data, as our cross-validation tests show logistic regression's accuracy can be increased by over 13% when employing optimized parameters that include an l1 penalty instead. Further, our tests indicate that even without modifying the elastic net parameter, 0.55 is unlikely to be a reliable measure of the true accuracy of an elastic net model on this data. By training the logistic regression model on 90% portions of the dataset iteratively over 10 folds and averaging the results, we find that this model's accuracy increases by nearly 19% when compared to the single 70% train and 30% hold-out test. On our final logistic regression test, an accuracy of 0.77 and ROC AUC of 0.81 was achieved by bagging optimized logistic regression models and evaluating them with a 10-fold cross validation on the classic_full dataset. This result represents an increase in accuracy of 0.32 and an increase in ROC AUC of 0.35 when compared to our reproduction of the evaluation procedure utilized in Angelova et al. These findings imply that logistic regression may be a more competent model for classifying depressive state in metagenomic signature data than initially indicated.

Random forest showed some promise to have a greater true performance when evaluated on a 10-fold cross-evaluation as well. Our 10-fold cross-validation test for random forest on the classic_full set was associated with an increase in accuracy of 0.12 and an increase in ROC AUC of 0.16 when compared with the single-split 70% train, 30% hold-out test. While our fully optimized and cross-validated evaluations of random forest accuracy only succeeded in matching those reported by Angelova et al. at 0.80, it is worth noting that our baseline reproduction of their methods produced an accuracy of only 0.7 when compared to their 0.8. This suggests firstly that we may not have accurately reproduced their study, but also that the true capacity of random forest and logistic regression may be higher even than reported in this paper. This discrepancy in baseline results is apparent for logistic regression as well- our baseline accuracy for logistic regression was measured to be 0.45 compared to their reported 0.55. We speculate this difference is attributable to our datasets' reduced feature width resulting from the inner join operation performed during our data preprocessing step. If the same increases in measurement of performance associated with our evaluation methods can be directly applied using Angelova et al.'s original datasets, random forest's revised evaluation of accuracy could be as high as 0.92, and logistic regression as high as 0.87.

In addition to attempting to reproduce and extend the performance of the classical models employed in Angelova et al., we sought to determine if support vector machines could classify depressive states competitively on this dataset. The results of our tests are promising, finding that a bagged support vector machine (SVM) using a radial basis function (RBF) kernel evaluated on a 10-fold cross-validation on the classic_full set achieved an accuracy of 0.77 and a ROC AUC of 0.81. Our tests on linear and polynomial kernels indicated these kernels may be less performant than the radial basis function for this type and distribution of data. The linear support vector classifier and support vector machine with the polynomial kernel evaluated at 0.70 accuracy, 0.81 ROC AUC, and 0.70 accuracy, 0.72 ROC AUC when using their optimal hyperparameters respectively.

Interestingly, the optimal gamma value found by grid search for the radial basis function kernel was 0.0001, indicating that a linear decision boundary may be preferred for this dataset's distribution. This implies that a radial basis function could be superfluous for classification on this data. This notion is further supported when comparing our optimized RBF SVM's performance metrics with those of logistic regression, another linear model. Both models' performances were found to be maximized at 0.77 accuracy and 0.81 ROC AUC in our work. It's noteworthy, however, that this implication is at odds with our finding that the linear support vector classifier was outperformed by the optimized RBF SVM and logistic regression model in our experiments. These seemingly contradictory results highlight an avenue for further investigation into linear support vector classifier and RBF SVM parameters.

A core aim of our work was to facilitate a controlled comparison of the YOLO (You Only Look Once) convolutional neural network (CNN) with classical models to determine if the additional preprocessing and computational overhead associated with training and evaluating the YOLOv8 CNN is warranted through improvements in classification performance. We additionally compared support vector machines to YOLOv8 following our findings above. To control for dataset size in our experiment, we trained an optimized logistic regression, random forest, RBF SVM, and linear support vector classifier on the 1866-instance synthetic dataset generated and employed by Angelova et al. to train the YOLOv8 CNN. We further controlled the conditions of our comparison by using the reference study's same 8 subject test set that was held out prior to generation of synthetic data points (yolo_test). Our experiments found that when trained on the synthetic set (yolo_train), our RBF SVM and Linear SVC were able to match YOLOv8's reported performance, classifying at 100% accuracy. The logistic regression and random forest models each misclassified a single data point.

We note here that YOLO's documentation recommends training on at least 10,000 instances per class. In the context of an original dataset containing 74 instances, generation of synthetic data is likely to be the most practical means to train YOLOv8. Under ideal conditions, additional metagenomic signature instances would be collected from real study participants through further recruitment. Understandably, this is likely to be a costly endeavor. Further experimentation may be illustrative here; These tests presently represent a single hold-out, and it's possible that a disproportionate number of outliers were represented in the test set. Additionally, an 8-sample test set is very small, which amplifies the impact of misclassification on accuracy. Further experimentation would benefit from iteratively sampling a test hold-out prior to synthesis of the training set.

Ultimately, our findings suggest that YOLOv8's dominance may be attributable largely to its expanded training set. It's plausible that with shuffled hold-out splits, logistic regression or random forest could match the performance of YOLOv8 on this data. On this specific hold-out test, our results indicate that for less overhead, a support vector machine may perform just as well as YOLOv8 for metagenomic signature data.

Finally, we evaluated our investigated models' assessments of feature importance in making classifications for logistic regression, random forest, SVM (RBF) and linear SVC. Interestingly, there was a wide diversity of results for this experiment, suggesting that the models may have employed different pathways to classify each individual. Despite some variety in ranking, there were some recurring themes. Estradiol 17-beta-dehydrogenase appeared to be a key enzyme, included in 3 of the 4 models' top 20 features. Additionally, this enzyme was present in 8 of the 80 features selected between all 4 models, representing 10% of the assessed important features overall. Butyryl-CoA dehydrogenase appears to be a key component of highly important features in classification as well, appearing at least once in every model's top 20 features. This enzyme also appeared 8 times between the aggregate 80 features, representing 10% of the selected enzymes.

Key species seem to include *Faecalibacterium prausnitzii*, *Blautia sp.*, and *Anaerobutyricum hallii*, appearing 23, 14, and 9 times in the 80 aggregated features respectively. Together, enzyme orthologs from these three species account for over half of all enzymes in the aggregate feature list.

Methods:

1. Dataset Preprocessing

Data for this study was obtained from the following sources:

1. **Kovtun et al.**[1]: Supplementary Table S6 included relative abundances of microbe/ortholog pairs at the genus and species levels. Although not directly used in this work's main analyses, this dataset offers a more comprehensive feature space that may support production of additional datasets for further research.
2. **NCBI Bioproject PRJNA762199**‘s summary provided supplementary demographic information. Each subject's age and sex also may be used in future research to expand the feature space of the datasets used in our work.
3. **Angelova et al.**[2]: Supplementary Table S1 contains raw sequence read counts of microbe/ortholog pairs at the species level. This is the primary dataset processed for use in this work. The provided Excel worksheet is composed of 9 sub-sheets in total, separated in the following table by their use in the study's classical model or YOLOv8 CNN evaluation:

Classical Model Sheets (rfel)		YOLOv8 CNN Sheets		
Test Sets	**Training Sets**	**Test Sets**	**Training Sets**	
			Real	**Synthetic**

HC_test_(rfel)
PwD_test_(rfel)
	HC_train_(rfel)
	PwD_train_(rfel)
		HC_test_(yolo)
		PwD_test_(yolo)
			HC_real_train_(yolo)
			PwD_real_train_(yolo)
				HC_syn_train_(yolo)
				PwD_syn_train_(yolo)

Supplementary Table S1 from Angelova et al.[2] was processed into training, test, and full sets for both the *rfel* (classical models of random forest & elastic net) and *yolo* sheets included in the excel worksheet. Python's *pandas* library was employed to load each .csv sheet into a *DataFrame*. Concatenation of datasets was performed iteratively to produce usable *DataFrames* for model training and testing. Within the *rfel* and *yolo* sets, HC (healthy control) and PwD (patient with depression) sheets were concatenated into single frames with the *join* parameter set to ‘inner' to prevent data leakage resulting from differing feature space sizes. The *train_(yolo)* sheets were further concatenated to combine the real and synthetic sets. For both *yolo* and *rfel*, an additional ‘full' dataset was produced by concatenating the training and test sets. The final resulting datasets after preprocessing are detailed in the following table:

Classical Model Sheets (rfel)			YOLOv8 CNN Sheets (yolo)		
Test Set	**Training Set**	**Full Set**	**Test Set**	**Training Set**	**Full Set**
classic_test	classic_train	classic_full	yolo_test	yolo_train	yolo_full

The yolo_test set represents an 8 subject set held out prior to the production of synthetic data in Angelova et al.

2. Logistic Regression and Random Forest Comparative Analysis

Following dataset preprocessing, a comparative analysis of logistic regression and random forest models was performed using Python's scikit-learn library. For each evaluation stage, performance metrics were calculated using functions from scikit-learn's metrics module, specifically accuracy_score, classification_report, roc_curve, auc, and confusion_matrix. All models and components in testing were initialized with a random_state parameter of 42 to ensure reproducibility of results.

2.1 Baseline Evaluation

Both logistic regression and random forest models were initialized with the hyperparameters reported in the reference study. For logistic regression, the solver parameter was set to 'saga' and the penalty parameter to 'elasticnet'. For the random forest model, scikit-learn's default settings were utilized.

The models were trained on the classic_train set and tested against classic_test. From this evaluation, preliminary performance metrics using the scikit-learn functions listed above were recorded.

2.2 10-Fold Cross Validation

Subsequently, scikit-learn's *KFold* module was employed to perform 10-fold cross-validation. This involved partitioning the classic_full dataset into ten equal folds, iteratively training the model on nine folds and validating it on the remaining fold. The performance metrics across all folds were computed and averaged to obtain a robust estimate of each model's generalizability.

2.3 Hyperparameter Optimization via Grid Search

A grid search with scikit-learn's *GridSearchCV* was performed to identify the optimal hyperparameters for both logistic regression and random forest classifiers. The grid search systematically explored sets of hyperparameter combinations, utilizing 5-fold cross-validation within each grid search iteration to evaluate performance based on the ROC AUC metric. The estimator that maximized ROC AUC was retrieved through *GridSearchCV*'s best_estimator_ attribute and was retrained and evaluated on a 10-fold cross validation on the classic_full set.

2.4 Ensemble Method: Bagging Logistic Regression

Lastly, bagging was applied to the optimized logistic regression model by wrapping it in scikit-learn's *BaggingClassifier* with the estimators parameter set to 105, max_samples set to 0.9, and max_features set to 0.85. This model's performance was again assessed through 10-fold cross validation on the classic_full set.

3. *Support Vector Machines (SVM) Analysis*

The performance of support vector machines (SVMs) with kernel types of radial basis function (RBF), polynomial, and linear (via *LinearSVC*) were investigated. Each support vector machine configuration was evaluated similarly to the procedure outlined above.

3.1 Baseline Evaluation

Using each model's respective scikit-learn default parameters, the three support vector machines were trained on the classic_train set and tested against classic_test. From this evaluation, preliminary performance metrics were recorded.

3.2 10-Fold Cross Validation

Performance metrics across all folds of the classic_full set were computed and averaged to obtain a robust estimate of each model's generalizability.

3.3 Hyperparameter Optimization via Grid Search

Scikit-learn's *GridSearchCV* was again employed, using a 5-fold cross-validation within each grid search iteration to evaluate performance based on the ROC AUC metric. For each SVM, the estimator that maximized ROC AUC was retrieved through *GridSearchCV*'s best_estimator_ attribute and was retrained and evaluated through 10-fold cross validation on the classic_full set.

3.4 Ensemble Method: Bagging

Lastly, bagging was applied to each optimized SVM by wrapping it in scikit-learn's *BaggingClassifier* with the estimators parameter set to 105, max_samples set to 0.9, and max_features set to 0.85. The models' performances were again assessed through 10-fold cross validation on the classic_full set.

4. Synthetic Dataset Comparison:

Logistic regression, random forest, SVM (RBF), and LinearSVC models were trained on the yolo_train set, each tuned with their respective optimal hyperparameters as determined by grid search in steps 2.3 and 3.3 outlined above.

Following training, the models were tested against the 8 subject yolo_test set. Key performance indicators from the *scikit-learn* metrics suite were again employed to measure model performance.

5. Feature Importance Analysis:

A variety of methods tailored to each algorithm were employed to extract the models' assessed importance of each feature. Resulting importances were ranked in descending order and filtered to their top 20 results.

5.a. Logistic Regression:

Model coefficients (i.e. the weights for each input feature) were extracted through the *LogisticRegression*'s *coef_* attribute.

5.b. Random Forest:

Feature importances were retrieved through the *RandomForestClassifier*'s *feature_importances_* attribute.

5.c. SVMs (RBF, LinearSVC):

Scikit-learn's *permutation_importance* function was employed to iteratively mutate each feature and record the extent to which each feature's mutations impacted model efficacy.

Conclusion:

Our findings indicate that simple classification models can prove to be more competitive on metagenomic signature data than previously reported, given optimized hyperparameters and a robust cross-validation approach. We also show that support vector machines with a radial basis function or linear kernel can match YOLOv8's reported performance in a controlled setting. These results suggest that YOLOv8's apparent superiority may stem primarily from its significantly larger training set rather than any fundamental advantage of convolutional neural network architecture.

Despite these findings, the small datasets and synthetic data generation may not capture the full complexity of true metagenomic distributions. Additionally, single-split test sets are prone to variance and may obfuscate the true performance of the models tested. The dataset sizes in our work highlight the need for expanded data collection and further iterative testing.

Looking ahead, deeper investigation into feature-importance rankings across diverse modeling strategies, plus larger, more generalized datasets would help solidify these findings. Future work should also consider deeper exploration of linear vs. non-linear decision boundaries for support vector machines. Ultimately, our work serves to advocate for simple and mid-complexity classifiers like logistic regression and support vector machines and their competence in predicting depressive states based on metagenomic signature data.

References:

Kovtun AS, Averina OV, Angelova IY, et al. Alterations of the Composition and Neurometabolic Profile of Human Gut Microbiota in Major Depressive Disorder. Biomedicines. 2022;10(9):2162. doi:10.3390/biomedicines10092162

Angelova IY, Kovtun AS, Averina OV, Koshenko TA, Danilenko VN. Unveiling the Connection Between Microbiota and Depressive Disorder Through Machine Learning. Int J Mol Sci. 2023;24(22):16459. doi:10.3390/ijms242216459