Autism Spectrum Disorder Biolog Project

Classification Model for Autism Disorder

CPSC 6300 - Applied Data Science

Spring 2023

Gaura Sinha
EDA
Logistic Regression, LDA, Ridge CV, Random Forest Classifier, Linear SVC
Report: Summary of EDA

Zachary Schmitz
EDA
Report: Summary of Machine Learning Models

Dharmesh Gopinath
EDA
Report: Introduction and Conclusion

1. INTRODUCTION

Autism, also known as Autism Spectrum Disorder (ASD), is a neurodevelopmental disorder that affects communication, social interaction, and behavior. It is a spectrum disorder, meaning that the severity and symptoms can vary widely from person to person. There is no cure for autism, but early diagnosis and intervention can help individuals with autism lead fulfilling lives.

Our project involves training a model to diagnose patients on the Autistic Spectrum. This is done by analyzing the metabolic absorption rates of various biomarkers in two different groups: patients (Autism Spectrum individuals) and Control (Non-spectrum individuals). It is of interest if people on the spectrum have different metabolic reactions towards the given biomarkers compared to those not on the spectrum. By recording the different metabolic activities of various individuals and training a machine learning model to use this data to properly diagnose individuals on the spectrum, we can give an early diagnosis and support for patients on the autism spectrum.

It is important to diagnose autism early because early intervention can lead to better outcomes for individuals with autism spectrum disorder (ASD). Children who receive early diagnosis and intervention have a greater chance of developing important communication, social, and self-care skills, and may be able to better manage symptoms of ASD as they grow older. Early diagnosis allows for appropriate services and support to be put in place, such as specialized education and therapy programs that can be tailored to the child's individual needs. This can help the child to develop skills and abilities that will help them succeed in school and in life. Early diagnosis also allows for families to better understand their child's condition, which can lead to better communication and support. It can also help parents and caregivers to better manage the stresses and challenges of raising a child with autism.

In addition, early diagnosis can help to rule out other conditions that may be causing symptoms similar to those of ASD, which can lead to more accurate and effective treatment. Overall, early diagnosis of autism is important because it can lead to better outcomes for individuals with ASD and their families.

The data used in this project was collected by Dr. Luigi Boccuto for his studies on Autism Spectrum. The dataset is gathered from the experiments conducted on Lymphoblastoid Cell Lines (LCLs) taken from 4 cohorts using the Biolog Phenotype MicroArrays (PM) which are preconfigured 96-well microplates coated with different oxidizable carbon sources in various wells. Of these datasets, the *50*

ASD vs 50 Controls is the only dataset with a clearly marked set of Control observations and Patient observations. Each participant has contributed a total of 734 data points to their label (Patient/Control), corresponding to each feature or biomarker that they were tested for.

2. SUMMARY OF EDA

2.1 The Dataset

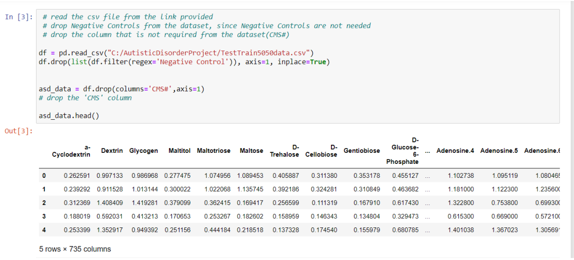
The dataset is gathered from the experiments conducted on Lymphoblastoid Cell Lines (LCLs) taken from 4 cohorts using the Biolog Phenotype MicroArrays (PM) which are preconfigured 96-well microplates coated with different oxidizable carbon sources in various wells. Of these datasets, the 50 ASD vs 50 Controls is the only dataset with a clearly marked set of Control observations and Patient observations. Each participant has contributed a total of 735 data points to their label (Patient/Control), corresponding to each feature or biomarker that they were tested for.

The dataset has 100 observations, each having 735 features spread out over 9 csv files. There are 100 unique observations that come with a label.

The data was collected over a period of time ranging from 2016 to 2017.

2.2 Cleaning and Preprocessing

The data for each patient was spread over various files and had to be collated. We transposed the data wherever needed for ease of use. We also dropped the identifier CMS# and also removed Negative control values so they do not interfere with the modeling.



In [3]:

```
# read the csv file from the link provided
# drop Negative Controls from the dataset, since Negative Controls are not needed
# drop the column that is not required from the dataset(CMS#)

df = pd.read_csv("C:/AutisticDisorderProject/TestTrain5050data.csv")
df.drop(list(df.filter(regex='Negative Control')), axis=1, inplace=True)


asd_data = df.drop(columns='CMS#',axis=1)
# drop the 'CMS' column

asd_data.head()
```

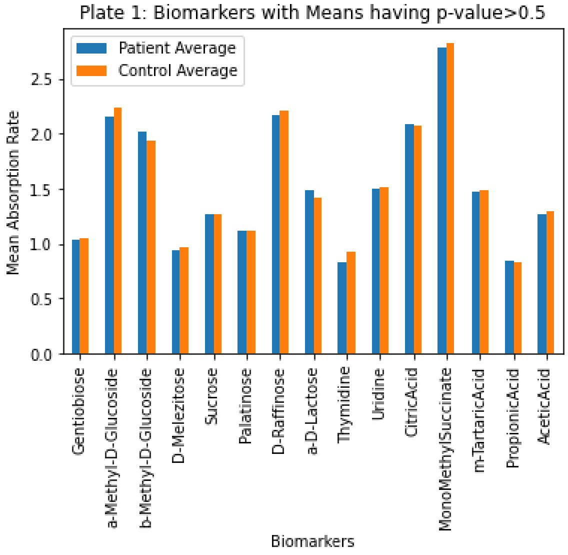
Out[3]:

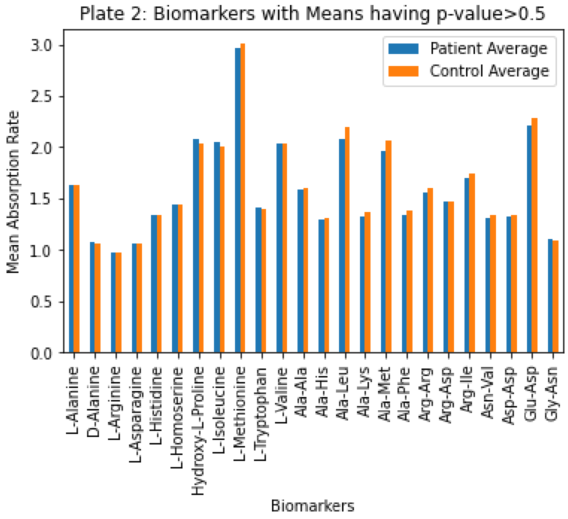
	a-Cyclodextrin	Dextrin	Glycogen	Maltitol	Maltotriose	Maltose	D-Trehalose	D-Cellobiose	Gentiobiose	D-Glucose-6-Phosphate	...	Adenosine.4	Adenosine.5	Adenosine.6
0	0.262591	0.997133	0.986968	0.277475	1.074956	1.089453	0.405887	0.311380	0.353178	0.455127	...	1.102738	1.095119	1.08046
1	0.239292	0.911528	1.013144	0.300022	1.022068	1.135745	0.392186	0.324281	0.310849	0.463682	...	1.181000	1.122300	1.23560
2	0.312369	1.408409	1.419281	0.379099	0.362415	0.169417	0.256599	0.111319	0.167910	0.617430	...	1.322800	0.753800	0.69930
3	0.188019	0.592031	0.413213	0.170653	0.253267	0.182602	0.158959	0.146343	0.134804	0.329473	...	0.615300	0.669000	0.57210
4	0.253399	1.352917	0.949392	0.251156	0.444184	0.218518	0.137328	0.174540	0.155979	0.680785	...	1.401038	1.367023	1.30569

5 rows × 735 columns

2.2 Visualization of Response

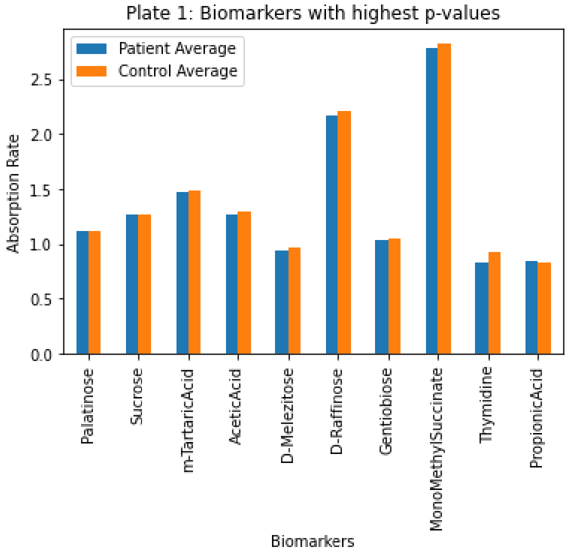
We used the Hypotheses Testing data for the averages of the biomarkers given to us in the p-value data, to plot features which had a p-value>0.5. We chose 0.5 to narrow down features with considerable differences in their means. We plotted the data for each of the 8 plates. Here are two of the 8 plots

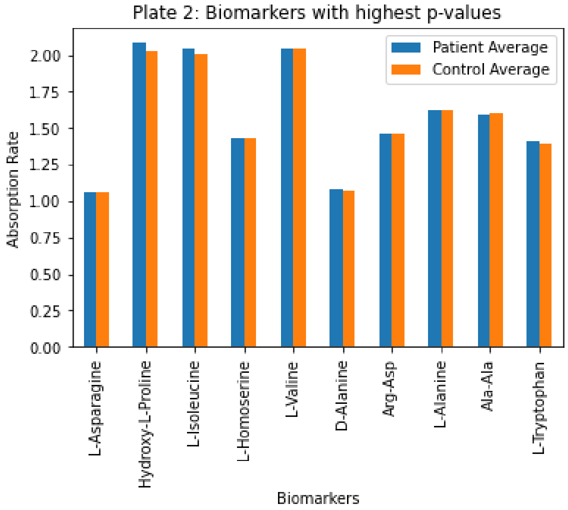


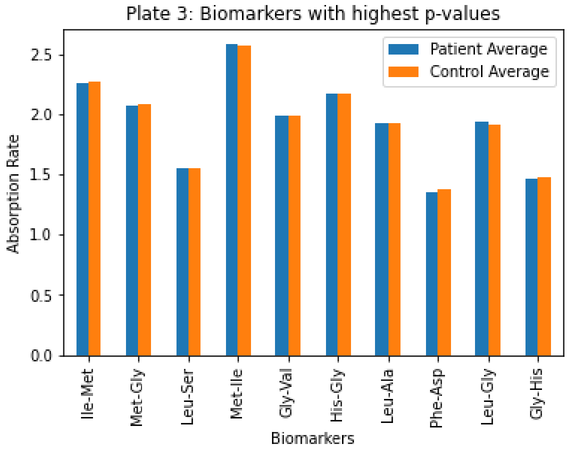


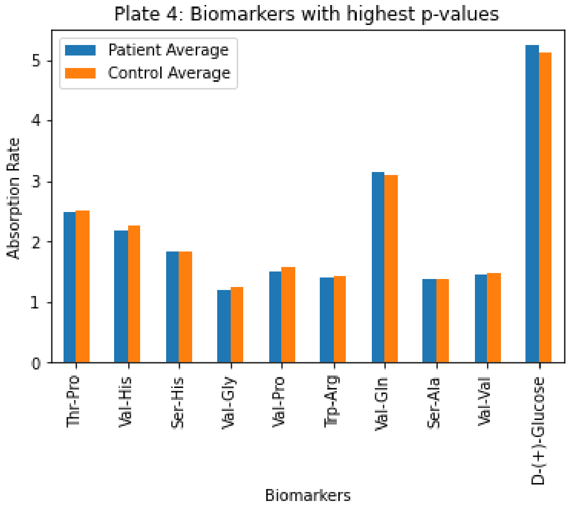
It seems to the human eye that the averages may not be the best parameter for a good choice of features. However we have greatly reduced the number of features we have to work with by this

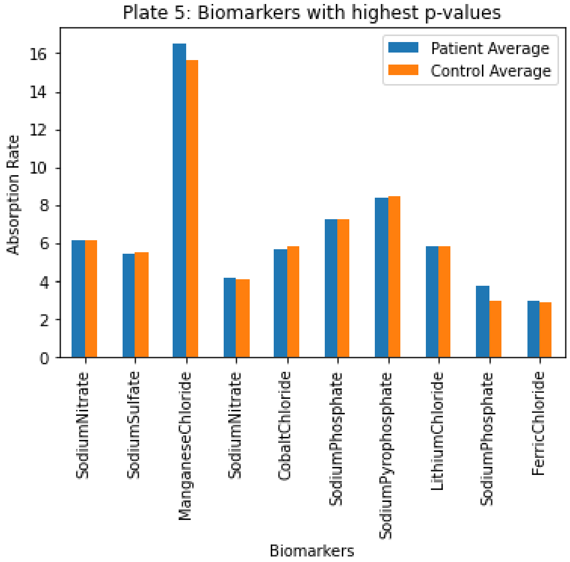
simple task. Ww went further to extract the features with top 10 p-values, ie, the greatest difference in the patient-control averages.
Here are the results for all the plates

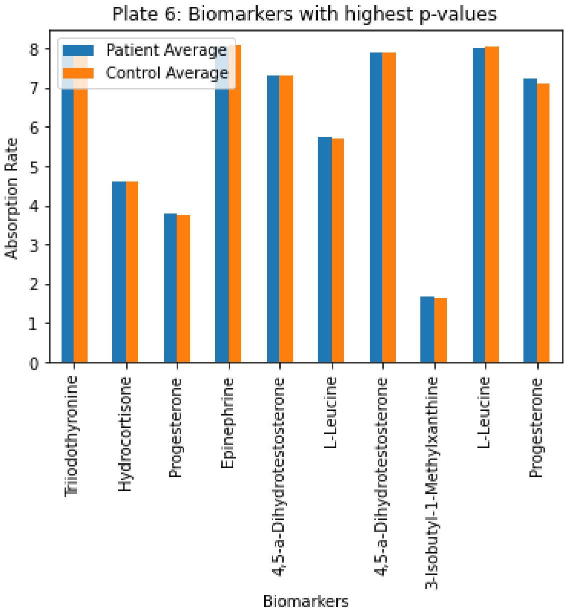


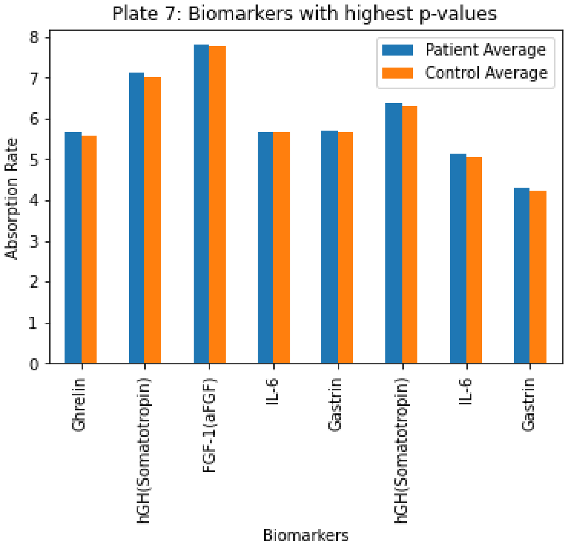


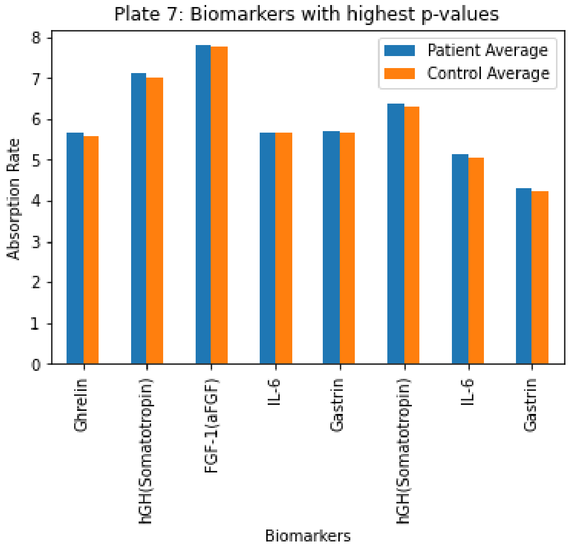








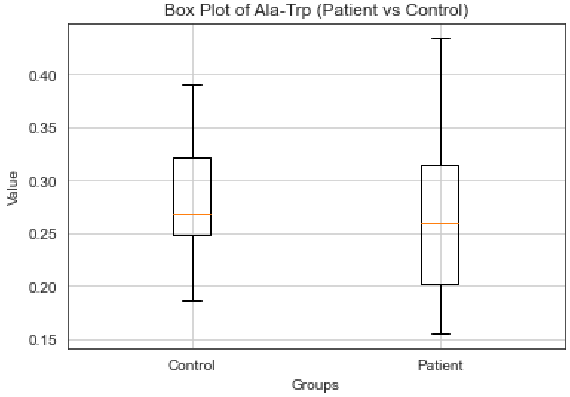


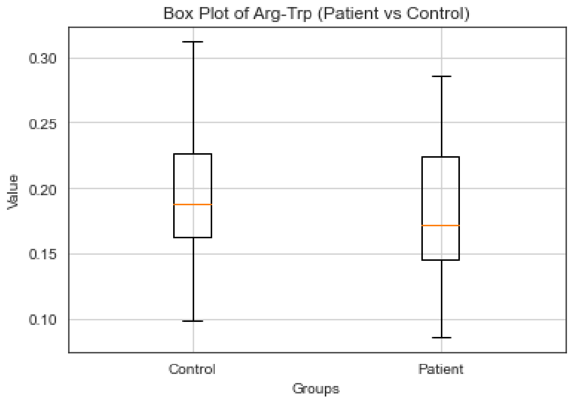


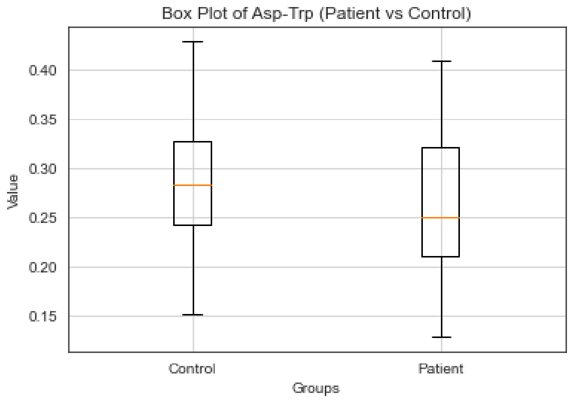
This is helpful to shortlist, but aggregation has its own pitfalls.

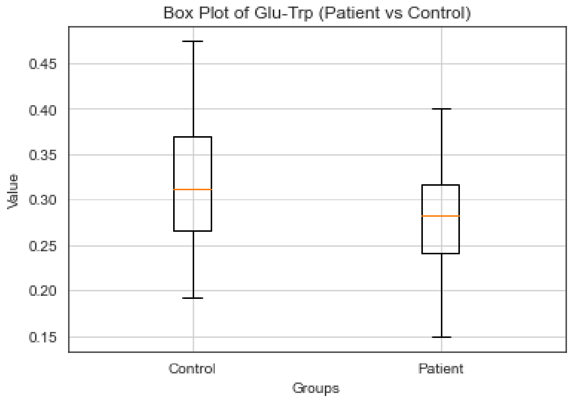
2.2 Visualization of Key Predictors

We specially took care to visualize the metabolic absorption rates of *Tryptophan*, a substance known to absorb differently in Patient cohort vs Control cohort.[1] We also plotted the data in box plots to minimize the effect of outliers.









We can clearly see a reduced absorption in the patient cohort for the substrates containing Tryptophan. We can say with certainty that these features are going to be important and have greater weight while training the classifier.

3. SUMMARY OF MACHINE LEARNING MODELS

We split our dataset into training and testing dataset and shortlisted our models to:

- Ridge CV Classifier
- Linear Discriminant Analysis
- Linear SVC
- Random Forest Classifier
- Logistic Regression

Metrics used to evaluate the classifiers were as follows:

Recall: Recall score is used to measure the model performance in terms of measuring the count of true positives in a correct manner out of all the actual positive values. This would be perhaps most important for our purposes, since we are mostly interested in True Positives. The recall is intuitively the ability of the classifier to find all the positive samples. The goal of a classification model such as this is to provide early intervention to help patients cope with Autism as quickly as possible. Hence we need to identify all Positives the best we can.

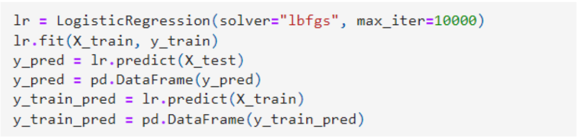
Precision: The precision is the ratio tp / (tp + fp) where tp is the number of true positives and fp the number of false positives. The precision is intuitively the ability of the classifier not to label as positive a sample that is negative. We can see why in a diagnosis we would not want a Negative diagnosis to be labeled as Positive

Accuracy: Accuracy is a measure of correctness of prediction of the model, and hence is included in the metrics to see how well the model can classify both positive and negative diagnosis.

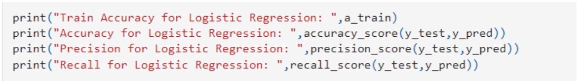
We also kept a check on **training accuracy** to keep an eye on overfitting.

3.1 Logistic Regression

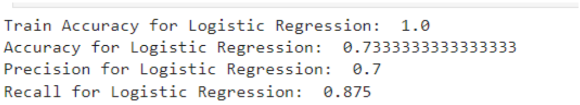
decided on using the logistic regression model because it represented the best fit we could find to represent our needed prediction, and serves our need for a classification model. Logistic regression is one of the most common algorithms used for classification models and it is a mathematical model used in statistics to estimate the probability of an event, given some previous data. Unlike decision trees or support vector machines, the logistic regression approach enables models to be quickly changed to reflect new data. It is possible to update using stochastic gradient descent. The predicted parameters (trained weights) give an inference about the importance of each feature. The direction of the association, i.e., positive or negative, is also given. So, we can use logistic regression to determine the features' relationship.



```
lr = LogisticRegression(solver="lbfgs", max_iter=10000)
lr.fit(X_train, y_train)
y_pred = lr.predict(X_test)
y_pred = pd.DataFrame(y_pred)
y_train_pred = lr.predict(X_train)
y_train_pred = pd.DataFrame(y_train_pred)
```



```
print("Train Accuracy for Logistic Regression: ",a_train)
print("Accuracy for Logistic Regression: ",accuracy_score(y_test,y_pred))
print("Precision for Logistic Regression: ",precision_score(y_test,y_pred))
print("Recall for Logistic Regression: ",recall_score(y_test,y_pred))
```



```
Train Accuracy for Logistic Regression:  1.0
Accuracy for Logistic Regression:  0.7333333333333333
Precision for Logistic Regression:  0.7
Recall for Logistic Regression:  0.875
```

The Logistic Regression model gave us fairly good results:

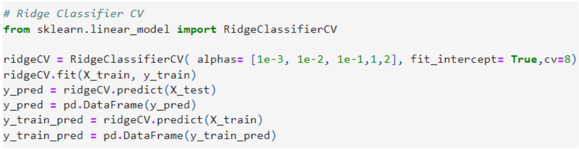
- Accuracy: 73.3%
- Precision: 70%
- Recall: 87.5%

3.2 Ridge CV Classifier

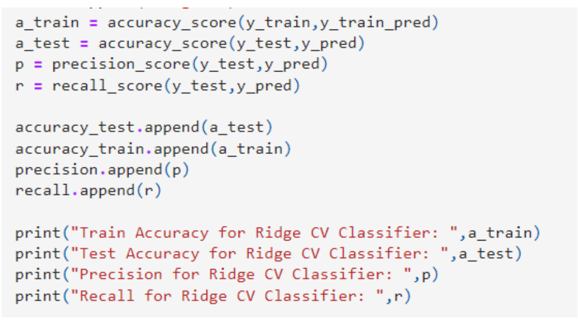
The Ridge Classifier CV (Cross-Validation) is an extension of the Ridge Classifier that performs cross-validation to tune the regularization parameter, alpha. Cross-validation is a technique used to evaluate the performance of a model by splitting the data into training and validation sets multiple times. By doing this, the model's performance can be assessed across different subsets of data, which helps to reduce the risk of overfitting.

The Ridge CV Classification model gave us the following results:

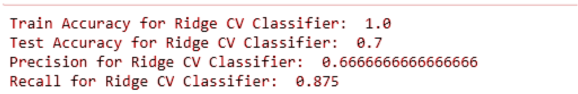
- Recall: 87.5%
- Accuracy: 70%
- Precision: 66.7%



```
# Ridge Classifier CV
from sklearn.linear_model import RidgeClassifierCV

ridgeCV = RidgeClassifierCV( alphas= [1e-3, 1e-2, 1e-1,1,2], fit_intercept= True,cv=8)
ridgeCV.fit(X_train, y_train)
y_pred = ridgeCV.predict(X_test)
y_pred = pd.DataFrame(y_pred)
y_train_pred = ridgeCV.predict(X_train)
y_train_pred = pd.DataFrame(y_train_pred)
```

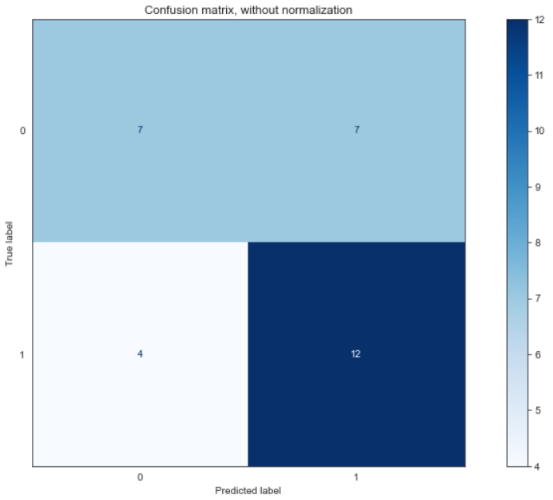


```
a_train = accuracy_score(y_train,y_train_pred)
a_test = accuracy_score(y_test,y_pred)
p = precision_score(y_test,y_pred)
r = recall_score(y_test,y_pred)

accuracy_test.append(a_test)
accuracy_train.append(a_train)
precision.append(p)
recall.append(r)

print("Train Accuracy for Ridge CV Classifier: ",a_train)
print("Test Accuracy for Ridge CV Classifier: ",a_test)
print("Precision for Ridge CV Classifier: ",p)
print("Recall for Ridge CV Classifier: ",r)
```



```
Train Accuracy for Ridge CV Classifier:  1.0
Test Accuracy for Ridge CV Classifier:  0.7
Precision for Ridge CV Classifier:  0.6666666666666666
Recall for Ridge CV Classifier:  0.875
```

We also obtained the following confusion matrix for our results:



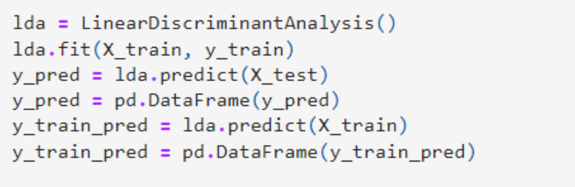
Upon applying the Ridge CV classification model, we were able to find a regularization parameter with solid results. With that parameter, we were able to create an instance of 70% test accuracy and 67% test precision, and while not great, those are still solid outcomes, leading the Ridge CV classification method to be a legitimate possible model.

3.3 Linear Discriminant Analysis

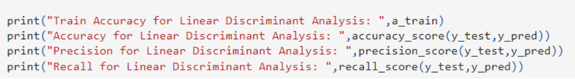
Linear Discriminant Analysis (LDA) is a supervised machine learning algorithm used for classification. It is commonly used for dimensionality reduction, feature extraction, and pattern recognition. The goal of LDA is to find a linear combination of features that best separates the classes in the data.

Using LDA we can attempt to simultaneously reduce dimensionality for the dataset, while exposing the critical components of separation. In this case, that would be finding the wells with similar variants of the same chemical component, and evaluating the resulting effects for each category of chemical. As described in the graph below, we can find that by applying LDA to our dataset, we can produce a normalization parameter with fairly significant results. The Linear Discriminant Analysis model gave us the following results:

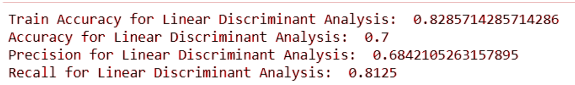
- Recall: 81.2%
- Accuracy: 82.9%
- Precision: 68.4%



```
lda = LinearDiscriminantAnalysis()
lda.fit(X_train, y_train)
y_pred = lda.predict(X_test)
y_pred = pd.DataFrame(y_pred)
y_train_pred = lda.predict(X_train)
y_train_pred = pd.DataFrame(y_train_pred)
```

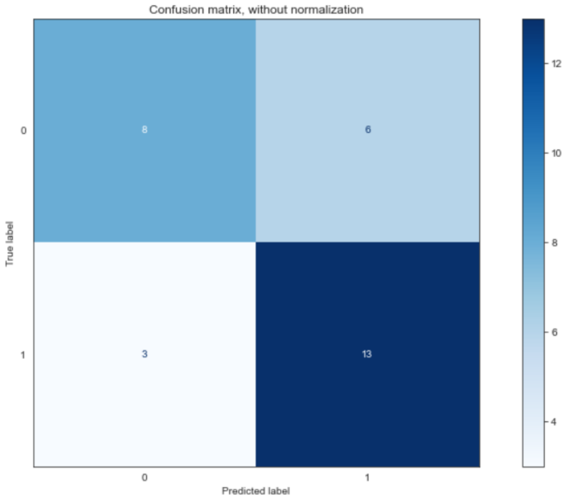


```
print("Train Accuracy for Linear Discriminant Analysis: ",a_train)
print("Accuracy for Linear Discriminant Analysis: ",accuracy_score(y_test,y_pred))
print("Precision for Linear Discriminant Analysis: ",precision_score(y_test,y_pred))
print("Recall for Linear Discriminant Analysis: ",recall_score(y_test,y_pred))
```



```
Train Accuracy for Linear Discriminant Analysis:  0.8285714285714286
Accuracy for Linear Discriminant Analysis:  0.7
Precision for Linear Discriminant Analysis:  0.6842105263157895
Recall for Linear Discriminant Analysis:  0.8125
```

We also obtained the following confusion matrix for our results:

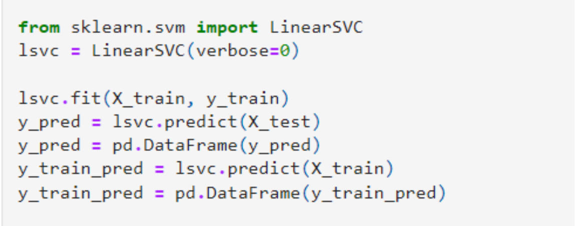


We achieved significant 82% Train Accuracy, with a near 70% Accuracy and a great 81.25% recall rate.

These results were encouraging because we clearly did not overfit, yet managed to maintain a good test accuracy.

3.4 Linear SVC

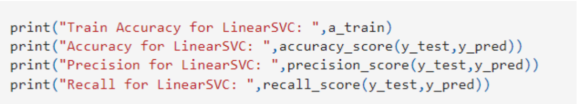
Linear Support Vector Classification (Linear SVC), commonly used for binary classification tasks, finds the optimal hyperplane that separates the two classes in the input data. Linear SVC is particularly useful for high-dimensional datasets where there are many features, as was the case with our dataset.



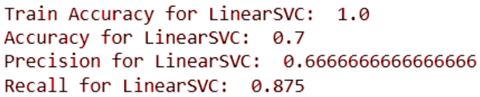
```
from sklearn.svm import LinearSVC
lsvc = LinearSVC(verbose=0)

lsvc.fit(X_train, y_train)
y_pred = lsvc.predict(X_test)
y_pred = pd.DataFrame(y_pred)
y_train_pred = lsvc.predict(X_train)
y_train_pred = pd.DataFrame(y_train_pred)
```

The Linear SVC model gave us the following results:

- Accuracy: 70%
- Precision: 66.7%
- Recall: 87.5%

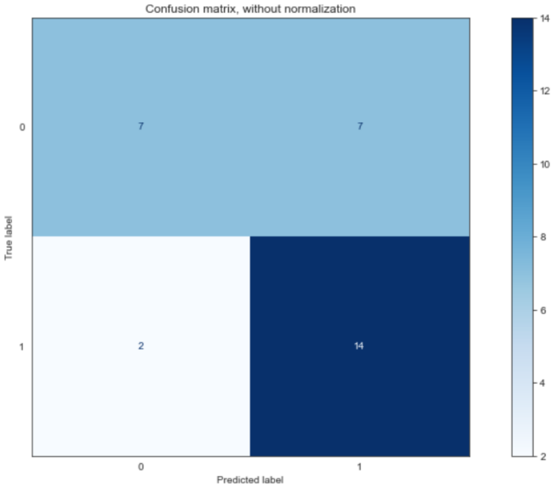


```
print("Train Accuracy for LinearSVC: ",a_train)
print("Accuracy for LinearSVC: ",accuracy_score(y_test,y_pred))
print("Precision for LinearSVC: ",precision_score(y_test,y_pred))
print("Recall for LinearSVC: ",recall_score(y_test,y_pred))
```



```
Train Accuracy for LinearSVC:  1.0
Accuracy for LinearSVC:  0.7
Precision for LinearSVC:  0.6666666666666666
Recall for LinearSVC:  0.875
```

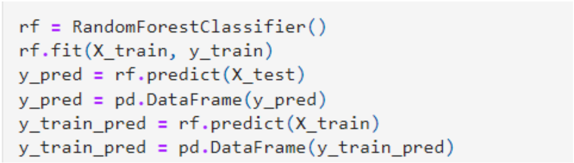
We also obtained the following confusion matrix for our results:



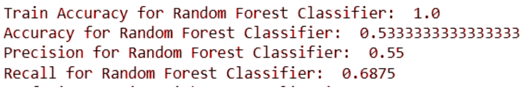
3.5 Random Forest Classifier

On the other hand, when using the Random Forest model, due to the shallow depth of tree variation, the model resulted in a fairly low accuracy and precision rate of around 55%. This indicates that this model is not a good model choice for our analysis. Random Forest Classifier model gave us the following results:

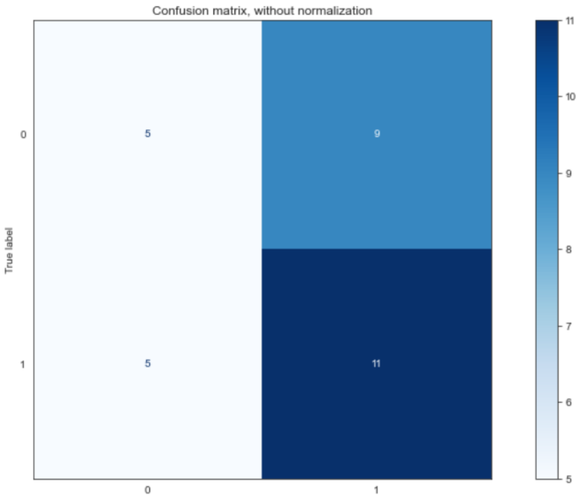
- Accuracy: 53%
- Precision: 55%
- Recall: 68.8%



```
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)
y_pred = pd.DataFrame(y_pred)
y_train_pred = rf.predict(X_train)
y_train_pred = pd.DataFrame(y_train_pred)
```

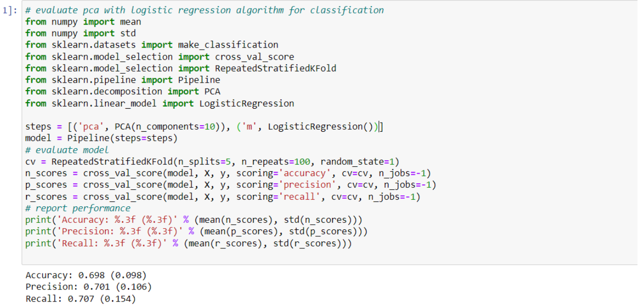


```
Train Accuracy for Random Forest Classifier:  1.0
Accuracy for Random Forest Classifier:  0.5333333333333333
Precision for Random Forest Classifier:  0.55
Recall for Random Forest Classifier:  0.6875
```



3.6 Dimensionality Reduction and PCA

Here we see that the recall is alright in all our cases, but the accuracy and precision not so much. We need to fine tune the model to get better results. One approach to getting better results would be to attempt some dimensionality reduction to get a better result.

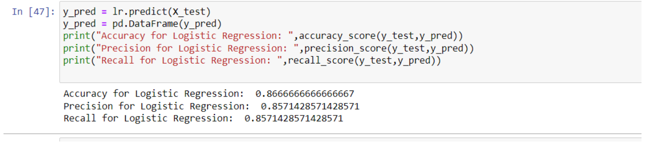


```
1]: # evaluate pca with logistic regression algorithm for classification
    from numpy import mean
    from numpy import std
    from sklearn.datasets import make_classification
    from sklearn.model_selection import cross_val_score
    from sklearn.model_selection import RepeatedStratifiedKFold
    from sklearn.pipeline import Pipeline
    from sklearn.decomposition import PCA
    from sklearn.linear_model import LogisticRegression

    steps = [('pca', PCA(n_components=10)), ('m', LogisticRegression())]
    model = Pipeline(steps=steps)
    # evaluate model
    cv = RepeatedStratifiedKFold(n_splits=5, n_repeats=100, random_state=1)
    n_scores = cross_val_score(model, X, y, scoring='accuracy', cv=cv, n_jobs=-1)
    p_scores = cross_val_score(model, X, y, scoring='precision', cv=cv, n_jobs=-1)
    r_scores = cross_val_score(model, X, y, scoring='recall', cv=cv, n_jobs=-1)
    # report performance
    print('Accuracy: %.3f (%.3f)' % (mean(n_scores), std(n_scores)))
    print('Precision: %.3f (%.3f)' % (mean(p_scores), std(p_scores)))
    print('Recall: %.3f (%.3f)' % (mean(r_scores), std(r_scores)))

    Accuracy: 0.698 (0.098)
    Precision: 0.701 (0.106)
    Recall: 0.707 (0.154)
```

With all three metrics, Accuracy, Precision and Recall hanging at around 70%, it became clear that PCA was not the solution to our problem and was ruled out.

3.7 Predictions of Model

We zeroed in on Logistic Regression as our best performing model. We used this model to predict a diagnosis on a validation dataset to get a classification and these are the metrics that we achieved.



```
In [47]: y_pred = lr.predict(X_test)
         y_pred = pd.DataFrame(y_pred)
         print("Accuracy for Logistic Regression: ",accuracy_score(y_test,y_pred))
         print("Precision for Logistic Regression: ",precision_score(y_test,y_pred))
         print("Recall for Logistic Regression: ",recall_score(y_test,y_pred))

         Accuracy for Logistic Regression:  0.8666666666666667
         Precision for Logistic Regression:  0.8571428571428571
         Recall for Logistic Regression:  0.8571428571428571
```

- Accuracy: 86.7%

- Precision: 85.7%
- Recall: 85.7%

These are excellent metrics and validate the choice of model we have made

4. SUMMARY AND CONCLUSION

Our chosen model has given promising results and with further refining will be of great use in the diagnosis of ASD(Autism Spectrum Disorder). The results of the analysis can provide valuable insights into the features and characteristics that are associated with ASD, but a diagnosis cannot be made solely based on these results. In order to diagnose ASD, a healthcare professional typically conducts a comprehensive evaluation that includes interviews with the individual and their family members, observation of the individual's behavior and communication, and a review of their medical and developmental history. Diagnostic tools such as the Autism Diagnostic Observation Schedule (ADOS) and the Autism Diagnostic Interview-Revised (ADI-R) may also be used to aid in the diagnosis of ASD.

Overall, while the results of the analysis can provide valuable information about the characteristics of individuals with ASD, a diagnosis should only be made by a trained healthcare professional using a comprehensive evaluation process.

The project can identify certain features of ASD. This can inform the work of experts by providing them with a list of key features to look for when evaluating patients for potential ASD. The data can also help validate the diagnosis of ASD, this can inform the work of experts by confirming the accuracy and relevance of the existing diagnostic criteria.

With more time and resources, we would have collected more labeled data and perhaps experimented with a greater number of biomarkers in order to boost our understanding of the relationship between the metabolic activity of these biomarkers in individuals with ASD. We would also work on thinning out the features as working with a large number of features is quite complicated at times. With further resources we could have also attempted to train a neural network and also incorporate Deep Learning algorithms into our project to further enhance the accuracy of our diagnosis.