AUTISM SPECTRUM DISORDER PREDICTION

Machine Learning Project

PROBLEM STATEMENT

According to a 2021 study published in the Indian Journal of Pediatrics, autism affects approximately 1 in 68 children in India. This is about 1 in 100 children under the age of 10, and is about 10 times higher than the 1.3% reported in India's 2011 census. The study also found that nearly 1 in 8 children in India has at least one neurodevelopmental condition

OBJECTIVE

The main objective of the project is to propose a method for the early detection of autism spectrum disorder (ASD) using Machine Learning techniques with the use of historical data (to be more specific attributes). Thereby helping earlier detection of autism, faster processing, and better life of individuals.

ABOUT THE DATASET

- ‘Autism Screening Adult’ dataset is used
- This dataset is composed of survey results for more than **800** people who filled an app form.
- Predict the likelihood of a person having autism using survey and demographic variables.
- Dataset Source : UCI repository

FEATURES OF THE DATASET

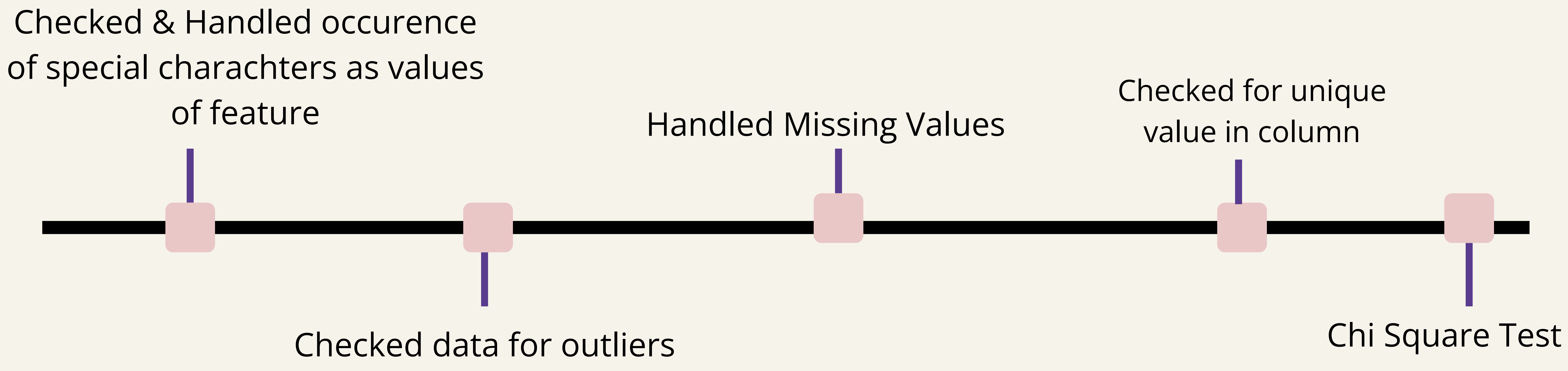
A1_Score to A10_Score (representing the 10 Autism Quotient Test Questions) (0,1)
Age
Gender (F, M)
Ethnicity
Age_desc
Jaundice (Yes, No)
Autism (family history) (Yes, No)
Country of residence
Used app before (Yes, No)
Result (AQ screening test score) Number
Class/ASD (target variable) Yes=1,No=0

Literature Review

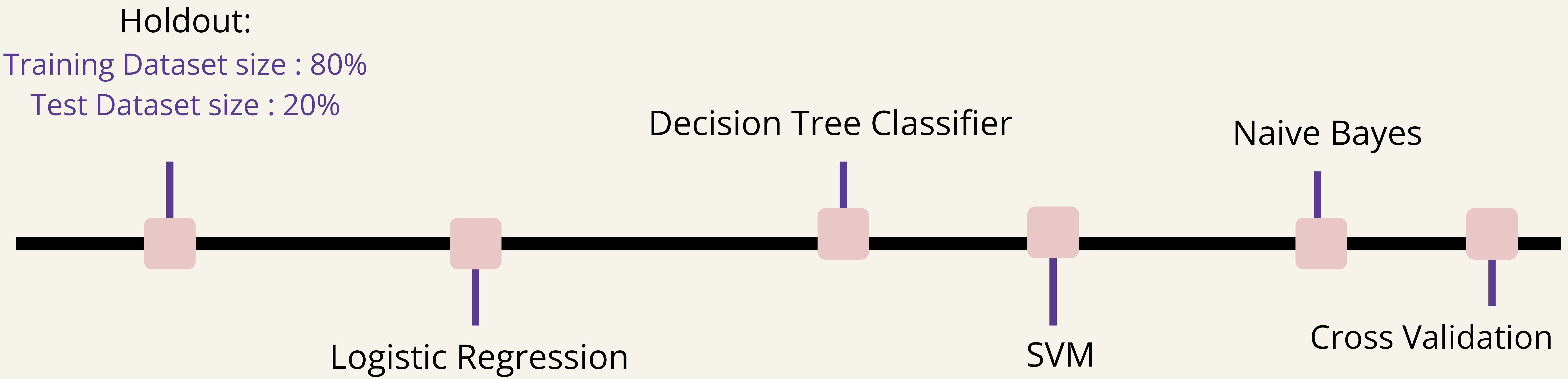
Paper title	Background	Materials & Methods	Result	Conclusion
Analysis and Detection of Autism Spectrum Disorder Using Machine Learning Techniques	Detection of Autism Spectrum Disorder was attempted using various machine learning and deep learning techniques	Dataset collected from the UCI Repository Data Pre-processing Apply ML Algorithms SVM, LR, NB, and CNN Evaluation of ASD and Non - ASD Classes	CNN based model was able to achieve highest accuracy result than all the other considered model building techniques ie 99.53	These results suggest that CNN based model can be implemented for detection of Autism Spectrum Disorder instead of the other conventional machine learning classifier
Predicting Autism Spectrum Disorder Using Machine Learning Classifiers	Potential of Machine Learning in providing faster, more accessible, and accurate diagnoses for ASD	Data Collection Data Understanding Pre-Processing Logistic Regression, Random Forest, MLP and XGBoost performance metric	(SVM) emerged as the most suitable model boasting the highest accuracy at 92%, precision at 0.845, recall at 0.865, and an F1-score of 0.853.	Handling medical datasets posed unique challenges, making the identification of the most potent classifier a significant achievement in our study.

IMPLEMENTATION

FLOWCHART


Checked & Handled occurence of special charachters as values of feature
Checked data for outliers
Handled Missing Values
Checked for unique value in column
Chi Square Test

FLOWCHART


Holdout:
Training Dataset size : 80%
Test Dataset size : 20%
Logistic Regression
Decision Tree Classifier
SVM
Naive Bayes
Cross Validation

LIBRARIES USED

1

Matplotlib, Seaborn:
For Data Visulisation

2

Pandas:
For Reading the dataset into a dataframe

3

scipy:
For Chi Square Test

4

sklearn:
For applying and evaluating models

CHECKED FOR MISSING VALUES

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 800 entries, 0 to 799
Data columns (total 22 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   ID               800 non-null    int64  
 1   A1_Score         800 non-null    int64  
 2   A2_Score         800 non-null    int64  
 3   A3_Score         800 non-null    int64  
 4   A4_Score         800 non-null    int64  
 5   A5_Score         800 non-null    int64  
 6   A6_Score         800 non-null    int64  
 7   A7_Score         800 non-null    int64  
 8   A8_Score         800 non-null    int64  
 9   A9_Score         800 non-null    int64  
 10  A10_Score        800 non-null    int64  
 11  age              800 non-null    float64
 12  gender           800 non-null    object 
 13  ethnicity        800 non-null    object 
 14  jaundice         800 non-null    object 
 15  austim           800 non-null    object 
 16  contry_of_res    800 non-null    object 
 17  used_app_before  800 non-null    object 
 18  result           800 non-null    float64
 19  age_desc         800 non-null    object 
 20  relation         800 non-null    object 
 21  Class/ASD        800 non-null    int64  
dtypes: float64(2), int64(12), object(8)
memory usage: 137.6+ KB
```

EXTRACTING THE CATEGORICAL COLUMNS AND NUMERICAL COLUMNS IN SEPARATE LIST FOR EASE OF EDA

```
num_cols, cat_cols = get_num_cat_cols(train_df)
```

```
Numerical columns
 ['ID', 'A1_Score', 'A2_Score', 'A3_Score', 'A4_Score', 'A5_Score', 'A6_Score', 'A7_Score', 'A8_Score', 'A9_Score', 'A10_Score', 'age', 'result', 'Class/ASD']
Categorical columns
 ['gender', 'ethnicity', 'jaundice', 'austim', 'contry_of_res', 'used_app_before', 'age_desc', 'relation']
```

CHECKED FOR UNIQUE VALUES IN CATEGORICAL COLUMNS

```
---------------------------------------------
Column Name - gender
---------------------------------------------
gender
m    530
f    270
Name: count, dtype: int64
---------------------------------------------
```

```
Column Name - age_desc
---------------------------------------------
age_desc
18 and more    800
Name: count, dtype: int64
```

Column - age_desc contains single value across all rows, so dropped this column

HANDLING SPECIAL CHARACHTER IN COLUMN

```
-----------------------------------------
Column Name - ethnicity
-----------------------------------------
ethnicity
White-European     257
?                  203
Middle Eastern      97
Asian               67
Black               47
South Asian         34
Pasifika            32
Others              29
Latino              17
Hispanic             9
Turkish              5
others               3
Name: count, dtype: int64
```

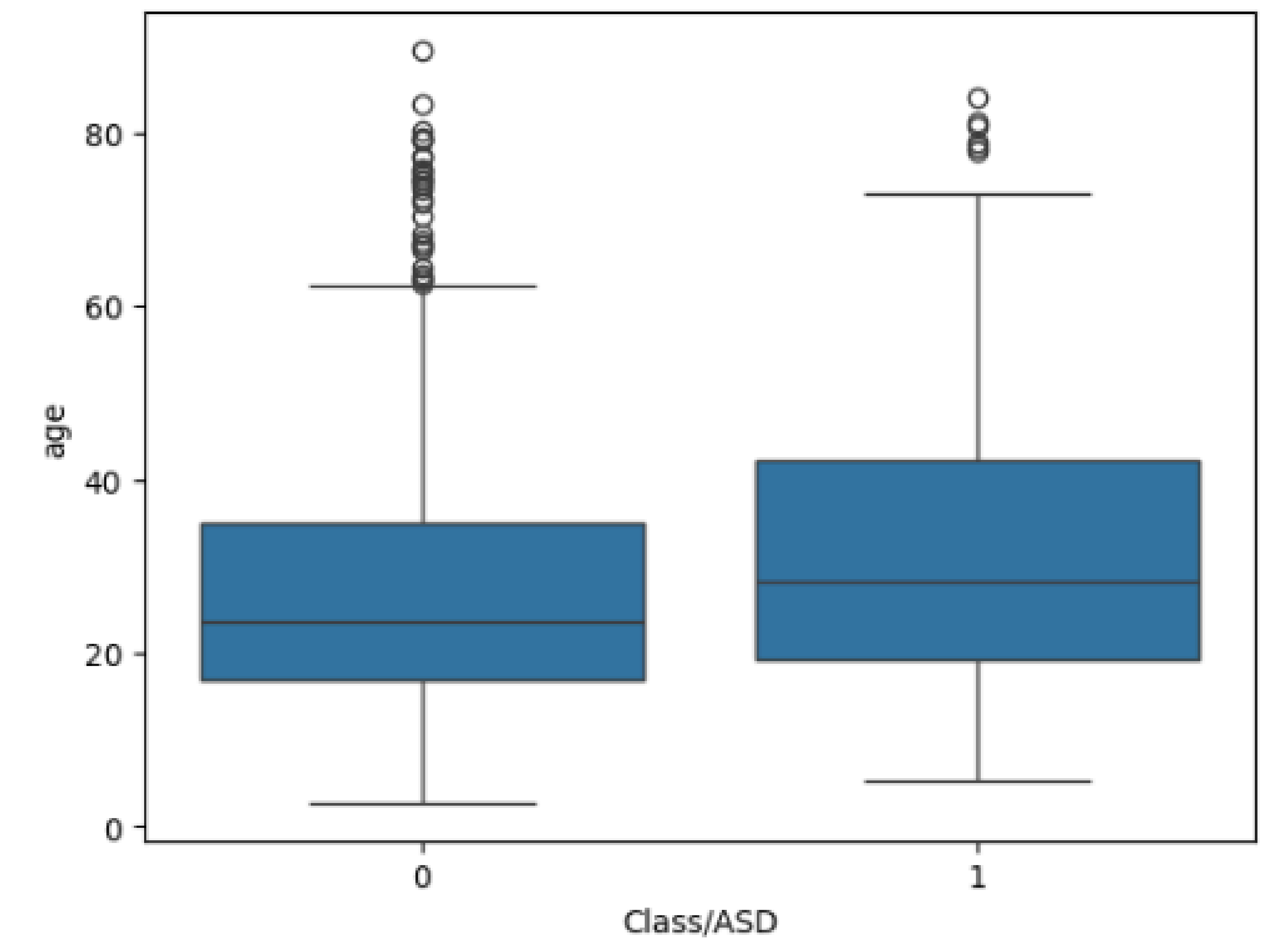
```
-----------------------------------------
relation
Self                          709
?                              40
Parent                         29
Relative                       18
Others                          2
Health care professional        2
Name: count, dtype: int64
```

Relation, Ethinicity column have special character '?', replacing"?" with 'others'

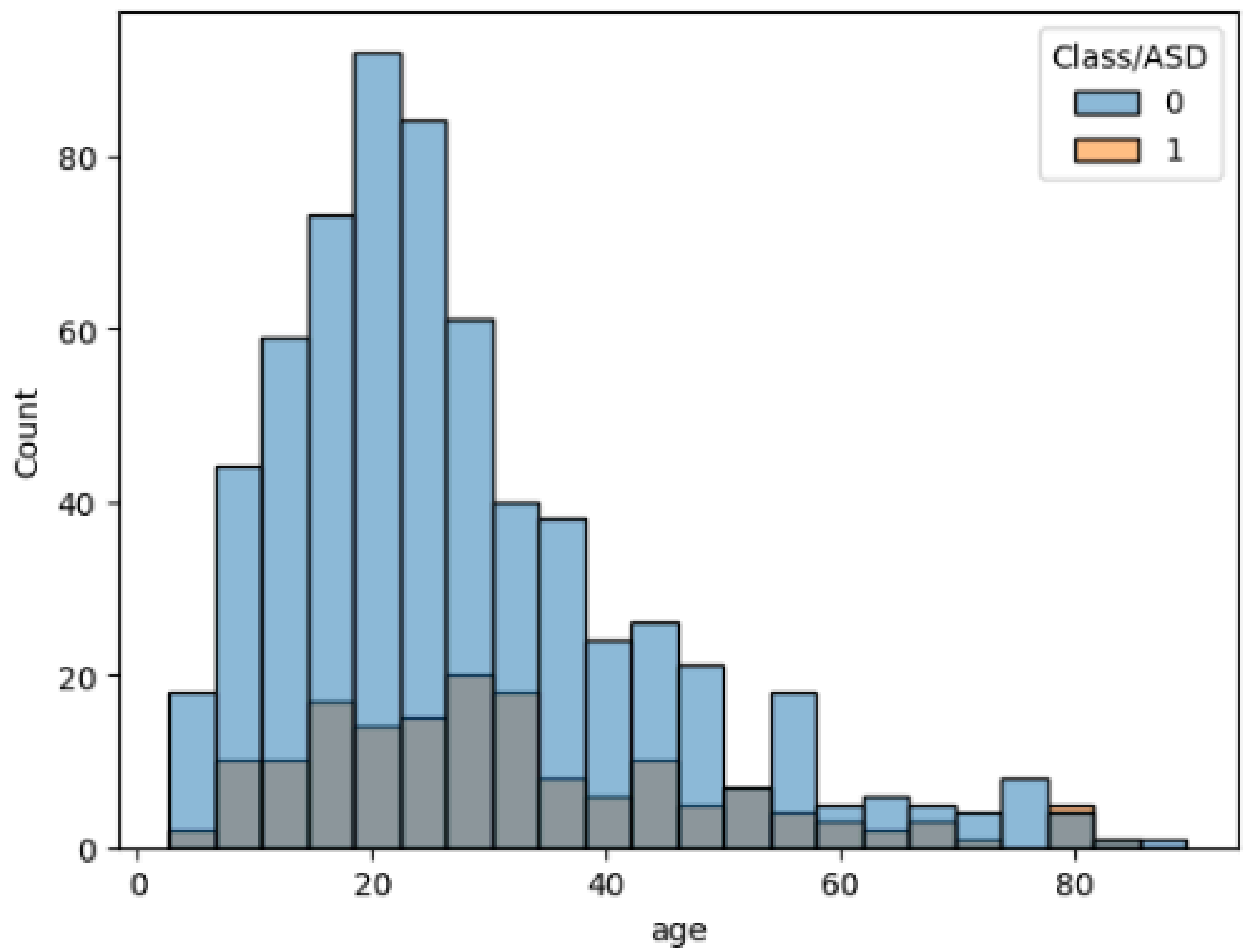
```
[ ]  train_df['relation'] = train_df['relation'].replace('?','Others')
     train_df['ethnicity'] = train_df['ethnicity'].replace('?','others')
```

ANALYISING NUMERIC FEATURE - AGE

<Axes: xlabel='Class/ASD', ylabel='age'>

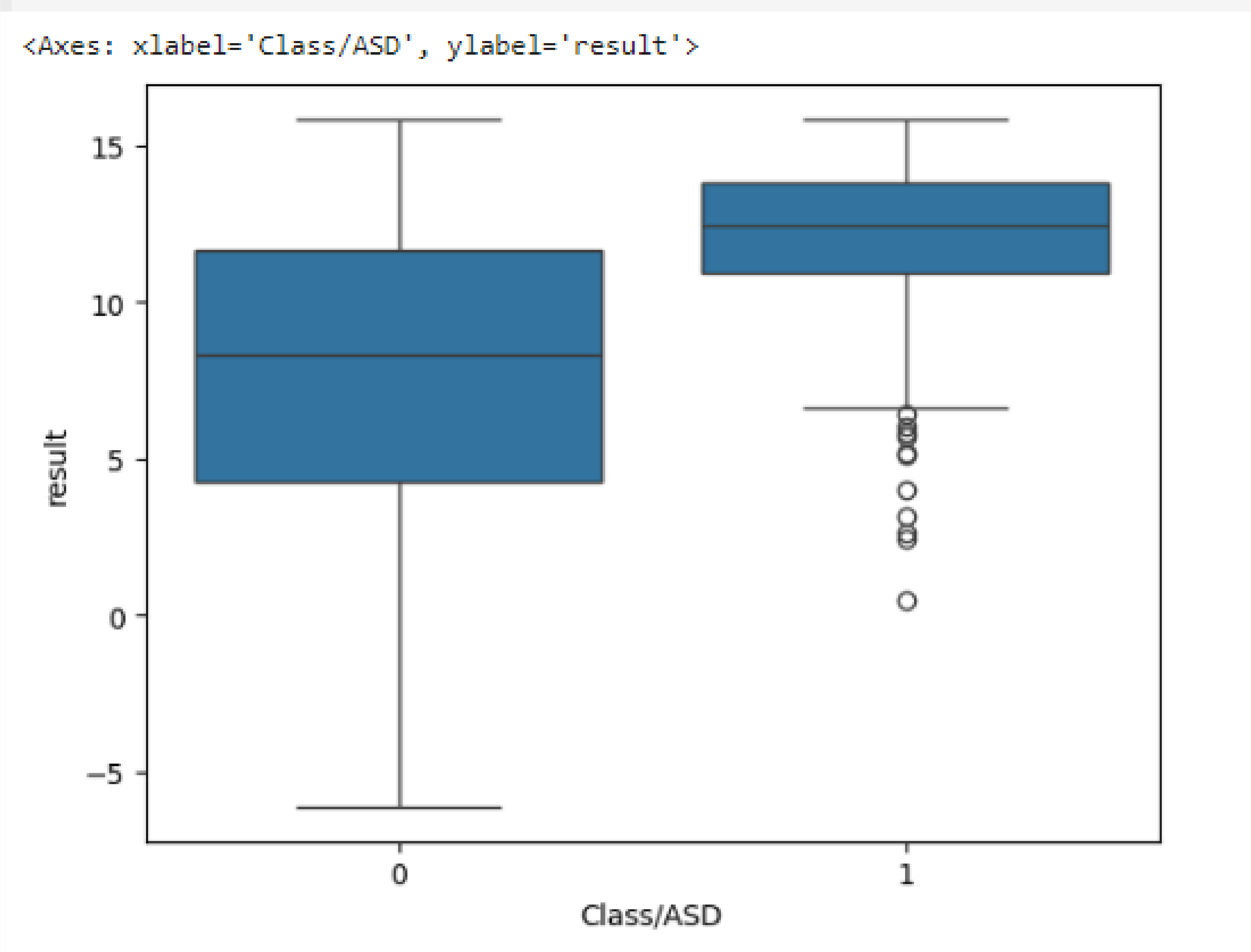


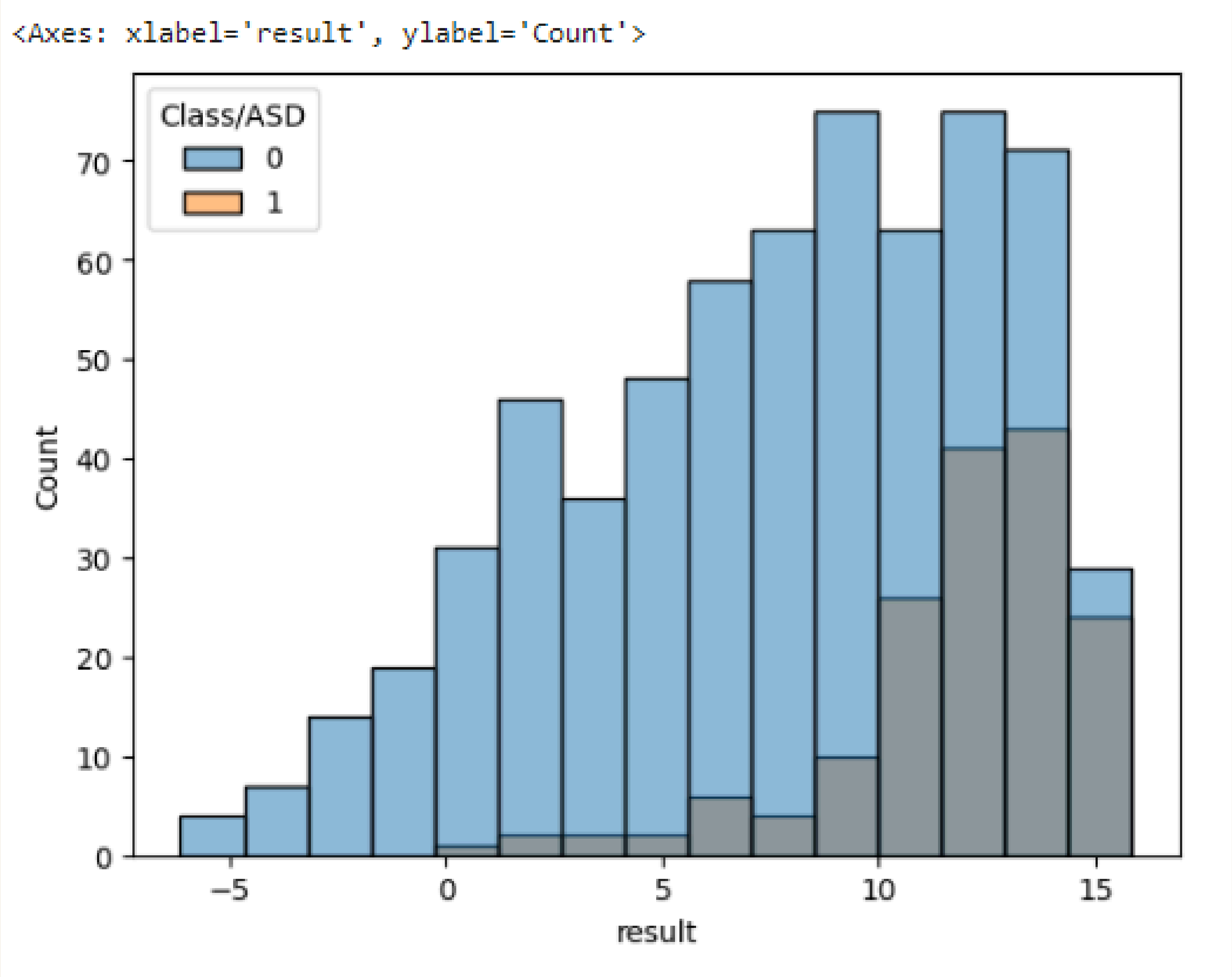
<Axes: xlabel='age', ylabel='Count'>



Range of age for both values of target class is same, thus age cannot be good classifier of target class. Thus dropping age column

ANALYISING NUMERIC FEATURE - RESULT





Range of result data for person having autism is shorter and higher when compared to person not having autism, thus it could be potential classifier of target class

FEATURE SELECTION USING CHI-SQUARE TEST

Chi-square test

The chi-square test is a statistical test used to determine whether there is a significant association between two categorical variables.

1.Define the Hypothesis:
Formulate the null hypothesis (H0) and the alternative hypothesis (H1) to determine whether there's an association between the categorical feature and the target variable.

2.Choose a Significance Level:
Select a significance level (α) to determine the threshold for accepting or rejecting the null hypothesis.

3. Create Contingency Table:
Construct a contingency table (cross-tabulation) to summarize the frequency distribution of the categorical feature and the target variable.

4. Expected Frequency:
Compute the expected frequency for each cell in the contingency table based on row and column totals.

5. Calculate Chi-Square Statistic:
Calculate the Chi-Square statistic using the formula, which measures the discrepancy between observed and expected frequencies normalized by the expected frequencies.

6.Calculate Degrees of Freedom:
Determine the degrees of freedom for the Chi-Square distribution based on the dimensions of the contingency table.
df = (total_rows - 1) * (total_cols - 1)

7.Find p-value:
Look up the p-value corresponding to the Chi-Square statistic and degrees of freedom in a Chi-Square distribution table or using statistical software.

8. Decide on Null Hypothesis:
Compare the obtained p-value with the chosen significance level (α). If the p-value is less than α, reject the null hypothesis, indicating a significant association between the categorical feature and the target variable.

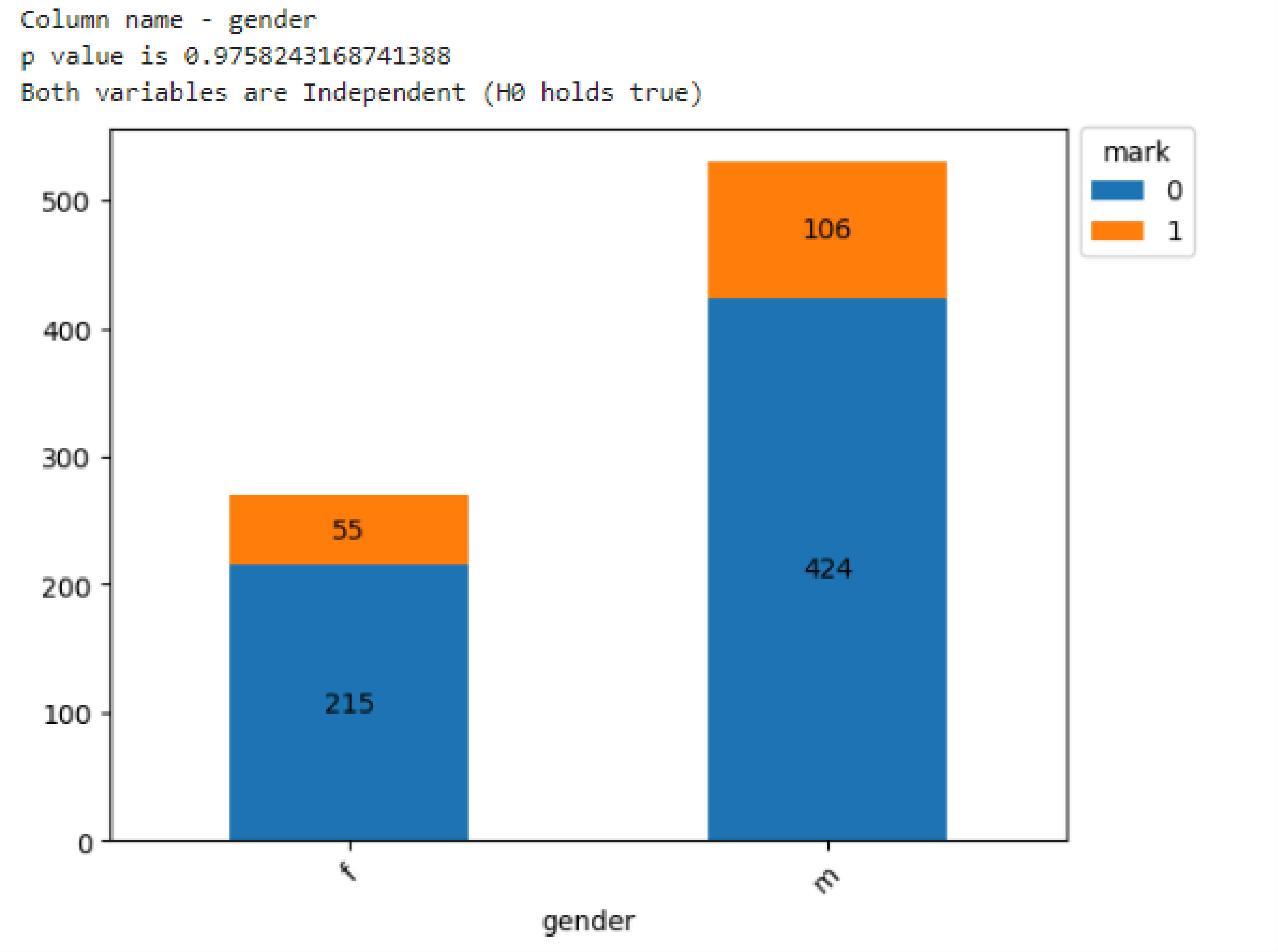
$$X^2 = \frac{\sum_{i=1}^{n} (Observed\ frequency - Expected\ frequency)^2}{Expected\ frequency}$$

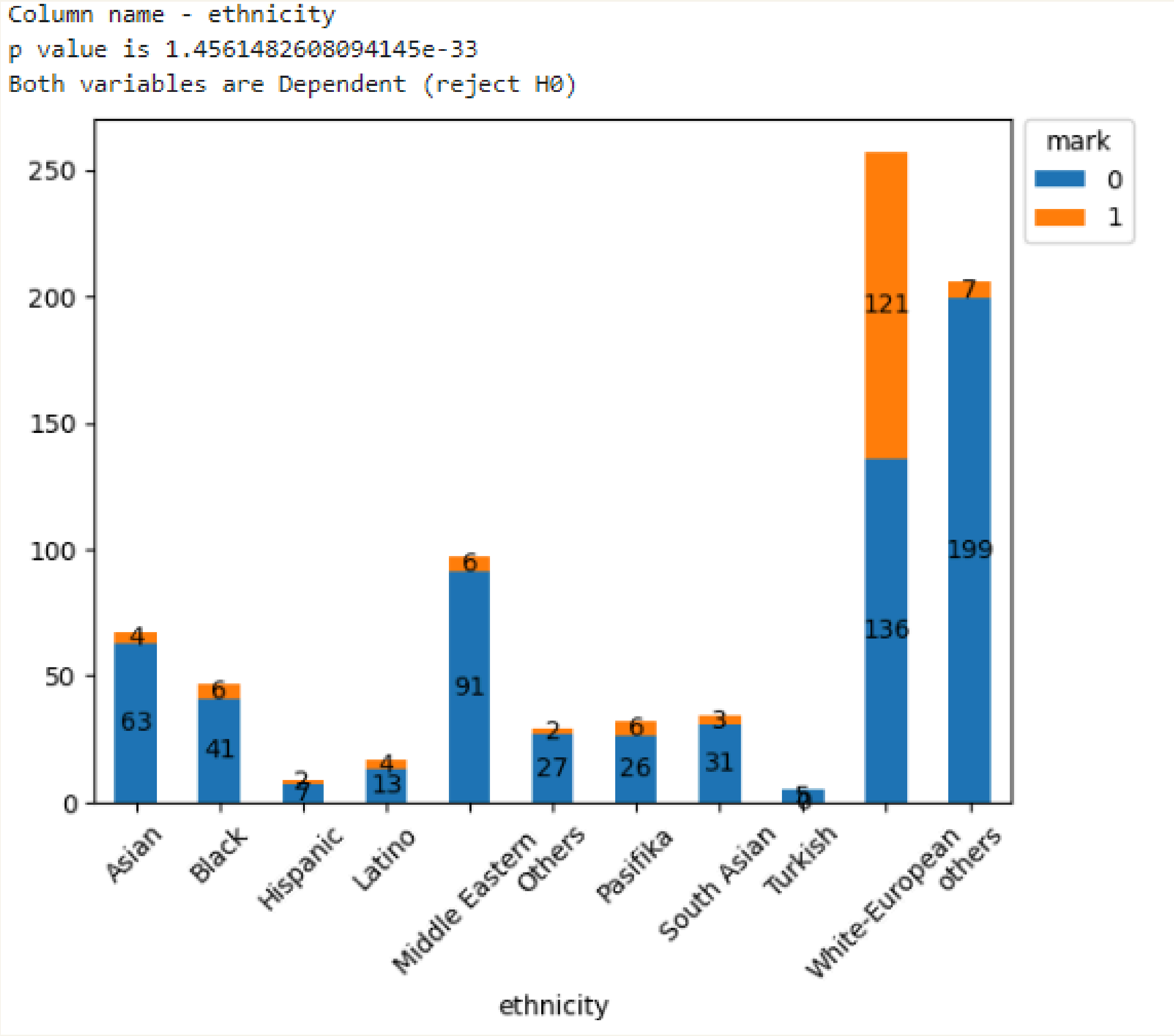
E = (row total x column total) / grand total

Why Chi-square test?

Chi-Square test is commonly used for feature selection when available features and target variable are categorical.

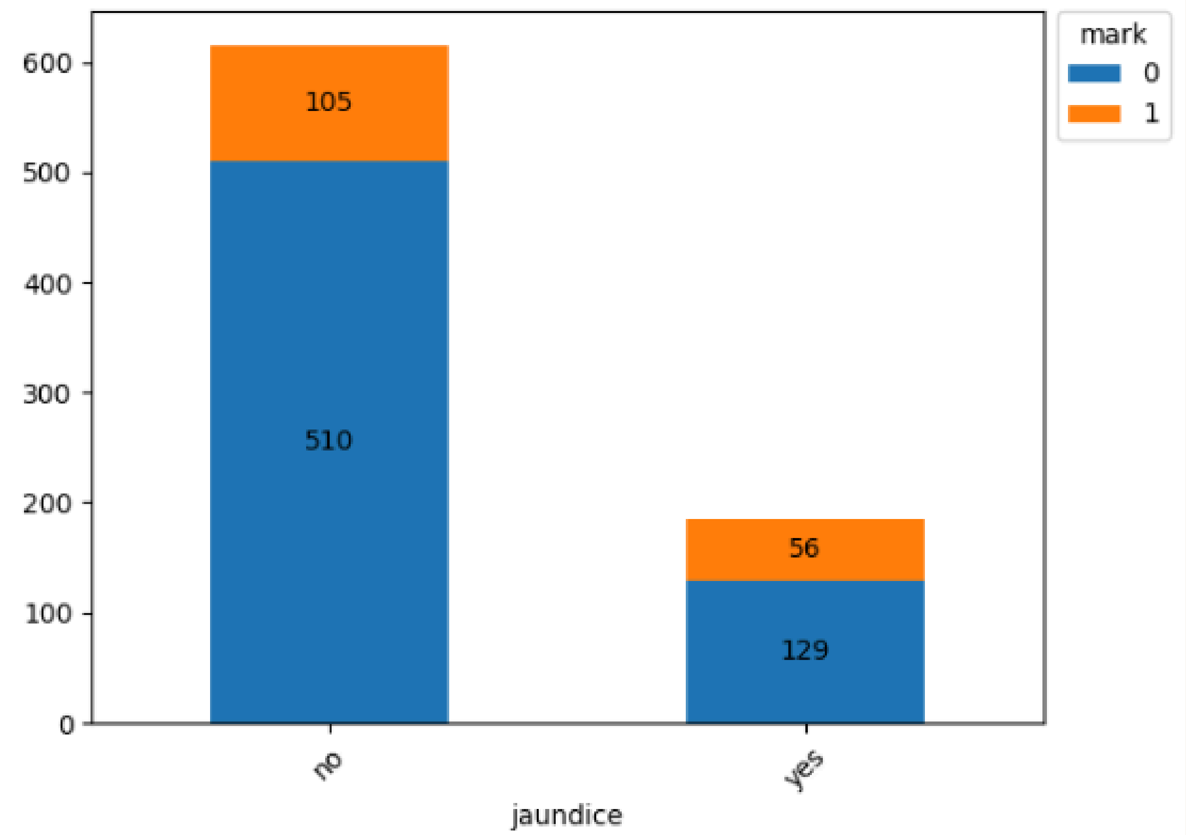
Using Chi Square Test to check dependence of categorical variable on target class



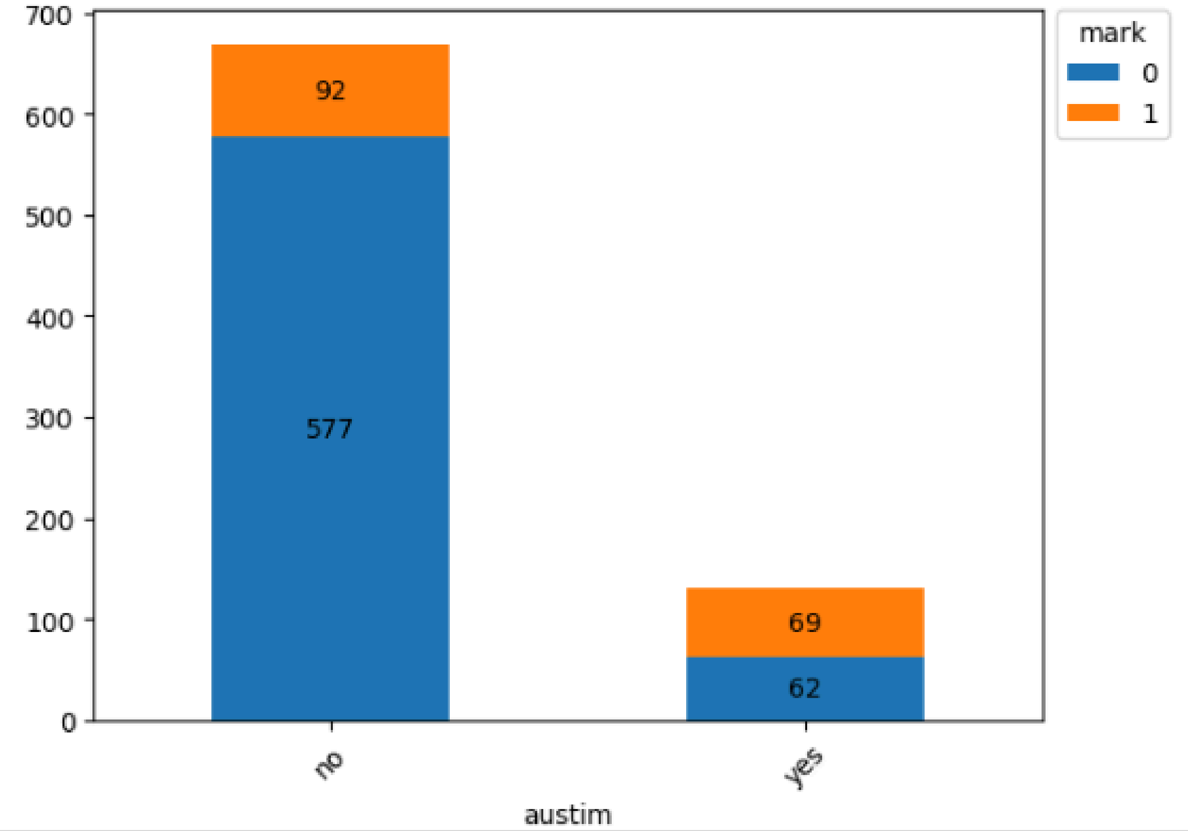


gender is independent from target variable, in chisquare test. Hence, dropping this feature.

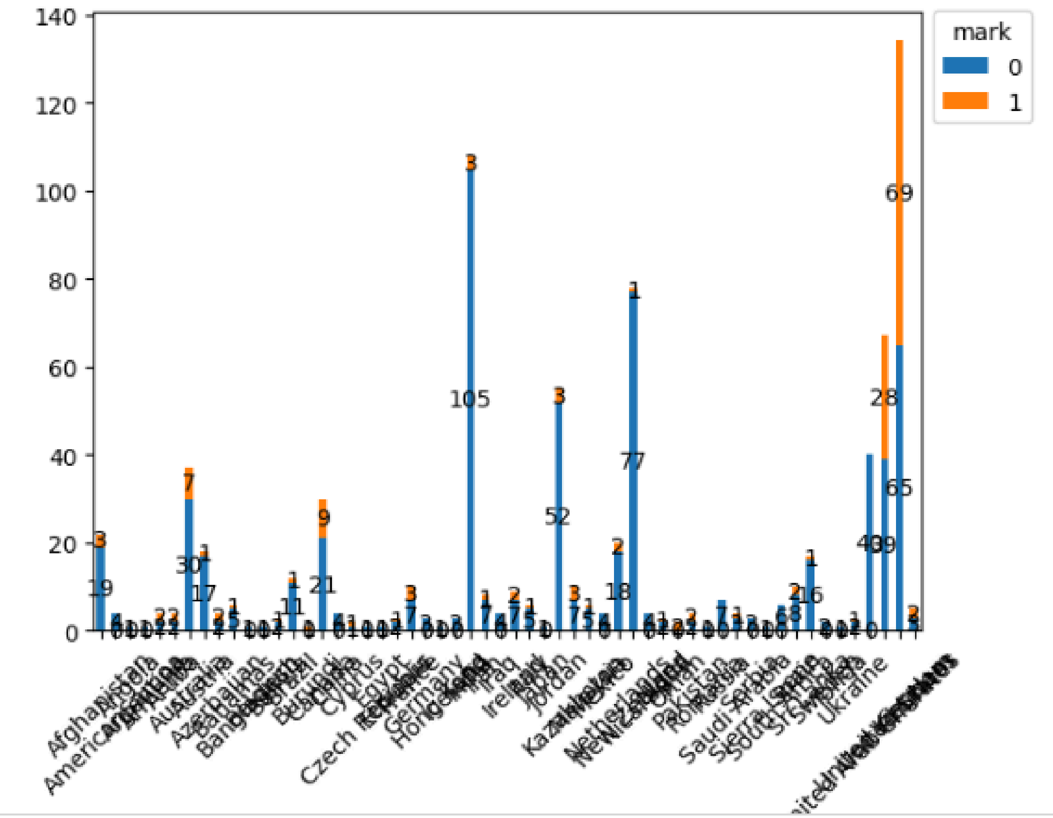
Column name - jaundice
p value is 0.00013300658957470307
Both variables are Dependent (reject H0)



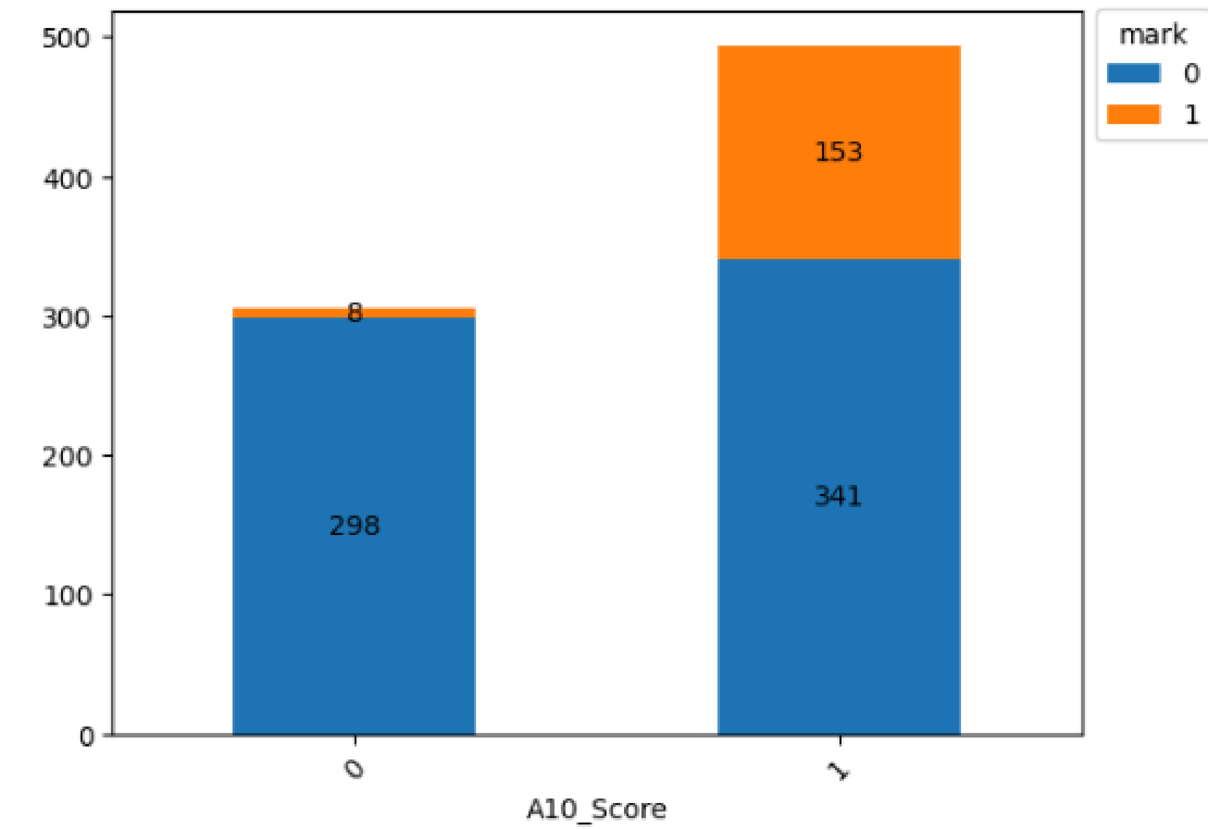
Column name - austim
p value is 1.0060560058593027e-23
Both variables are Dependent (reject H0)



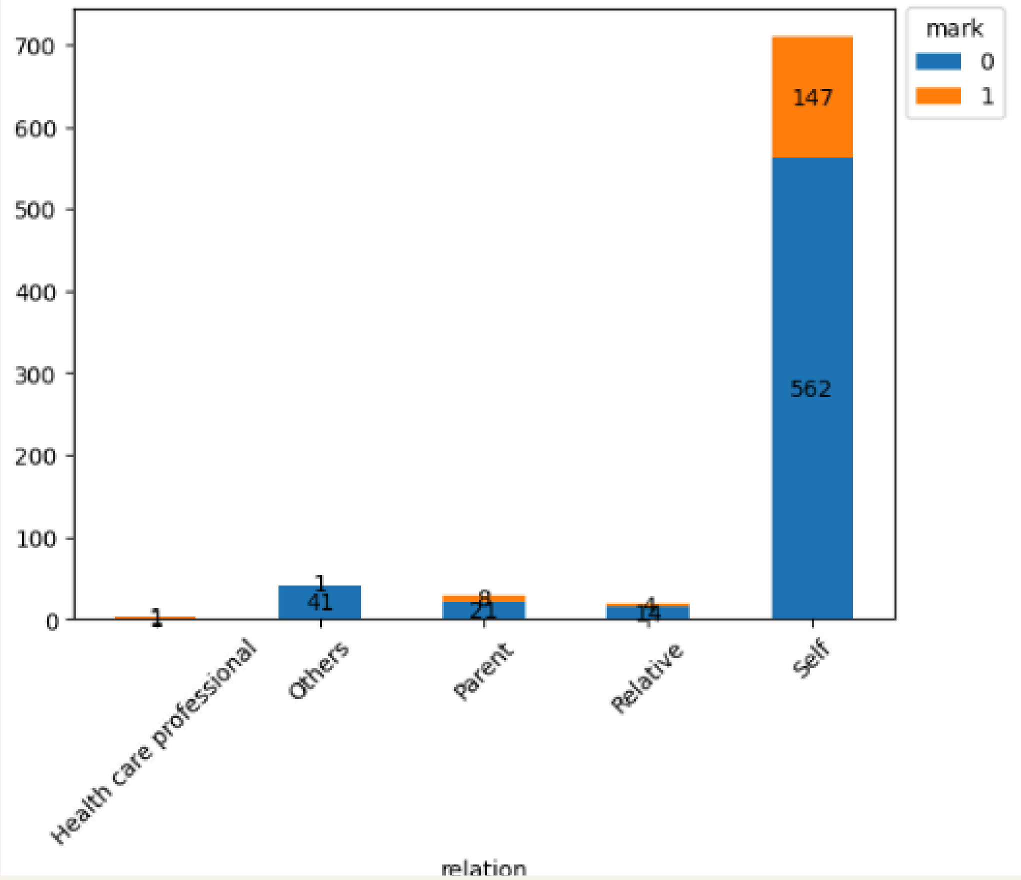
Column name - contry_of_res
p value is 2.8611111937550227e-19
Both variables are Dependent (reject H0)



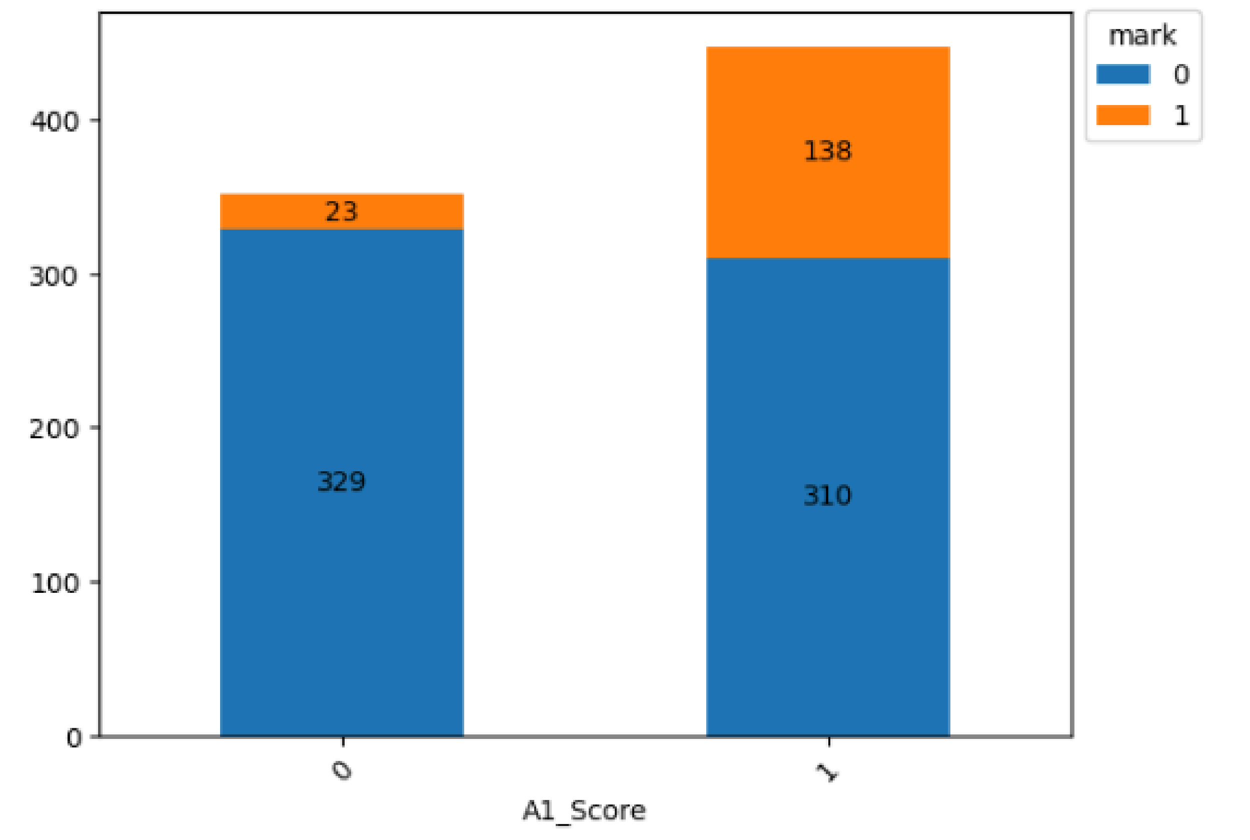
Column name - A10_Score
p value is 5.880205552224773e-22
Both variables are Dependent (reject H0)



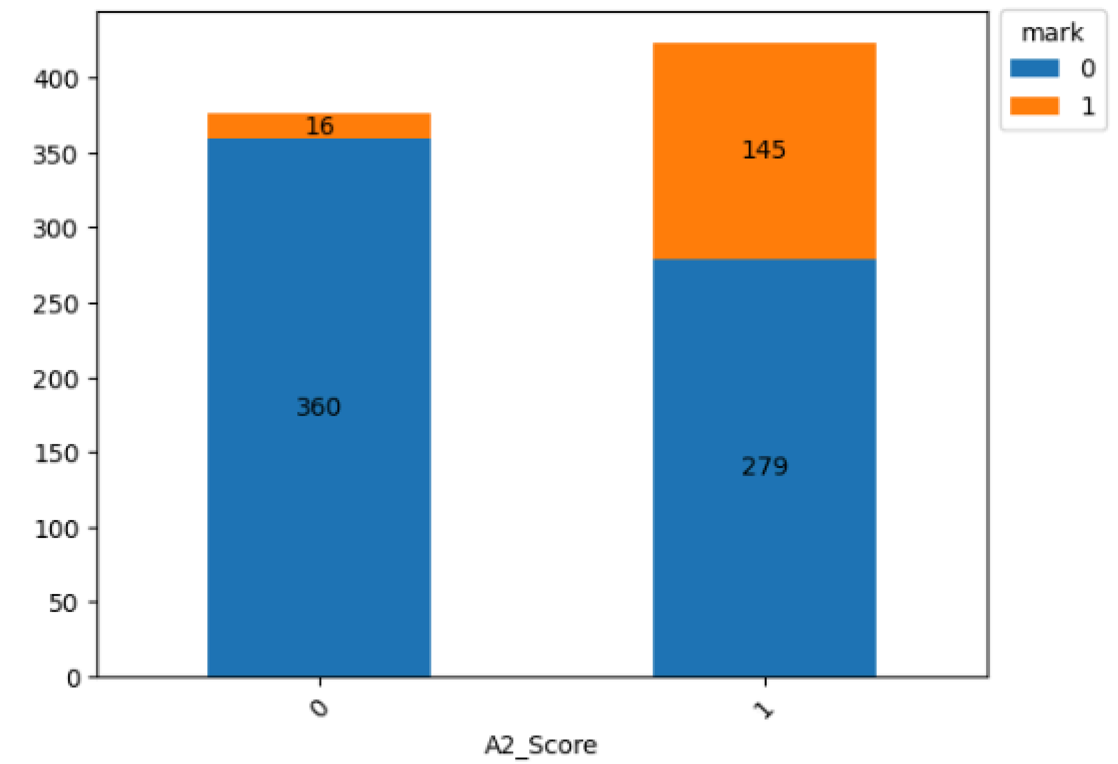
Column name - relation
p value is 0.03206590795264572
Both variables are Dependent (reject H0)



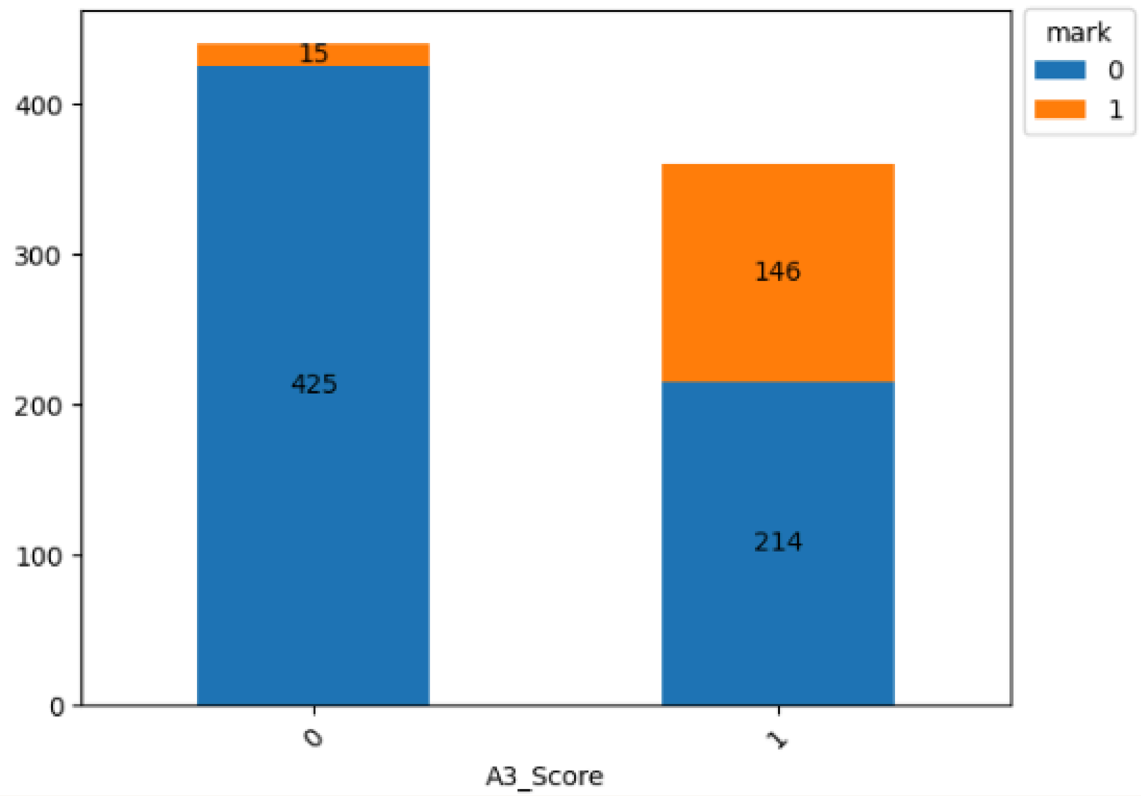
Column name - A1_Score
p value is 4.104487536920418e-17
Both variables are Dependent (reject H0)



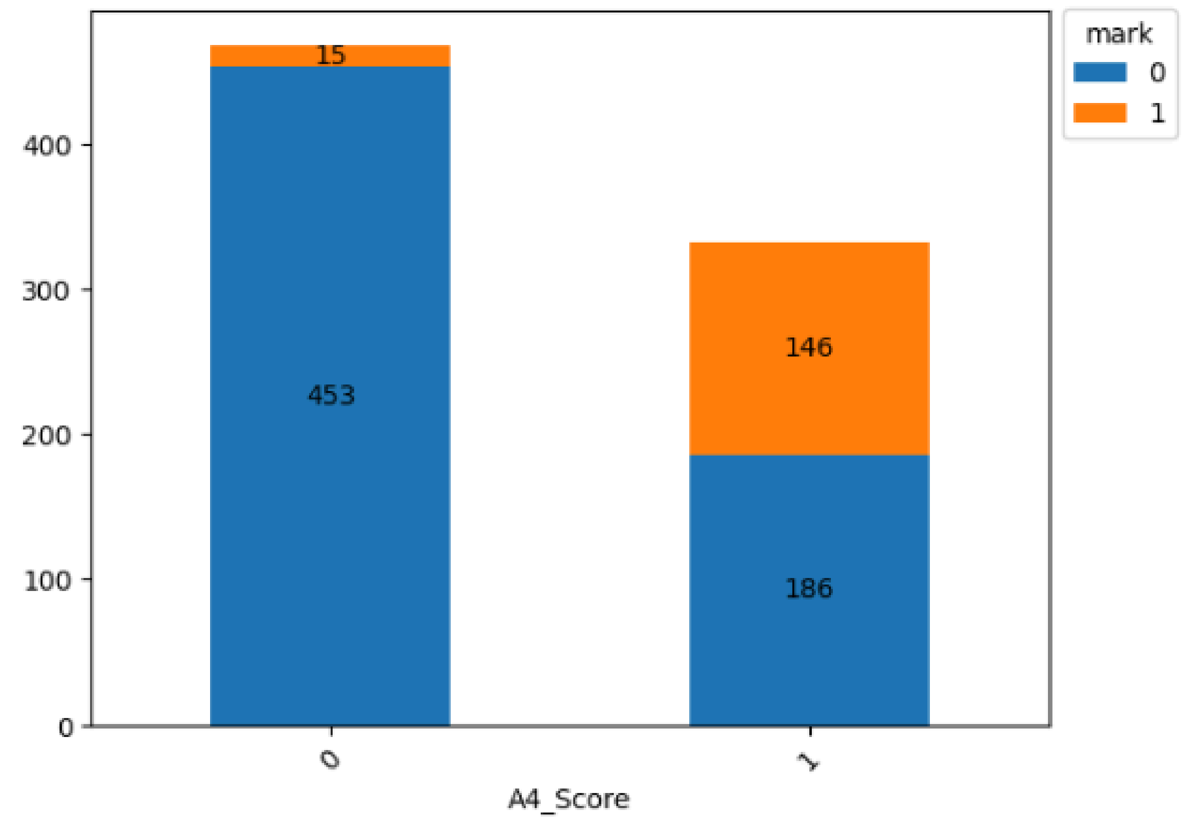
Column name - A2_Score
p value is 1.3998012922364413e-25
Both variables are Dependent (reject H0)



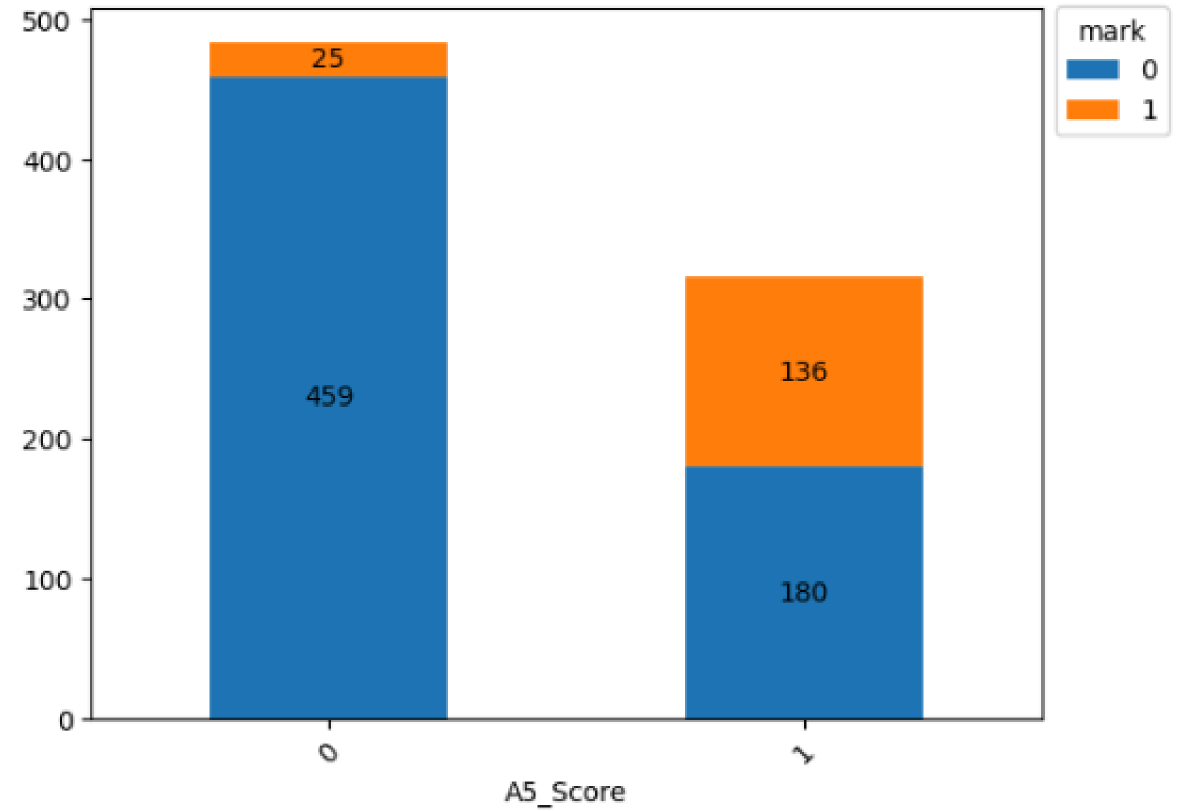
Column name - A3_Score
p value is 2.4007562062687566e-38
Both variables are Dependent (reject H0)



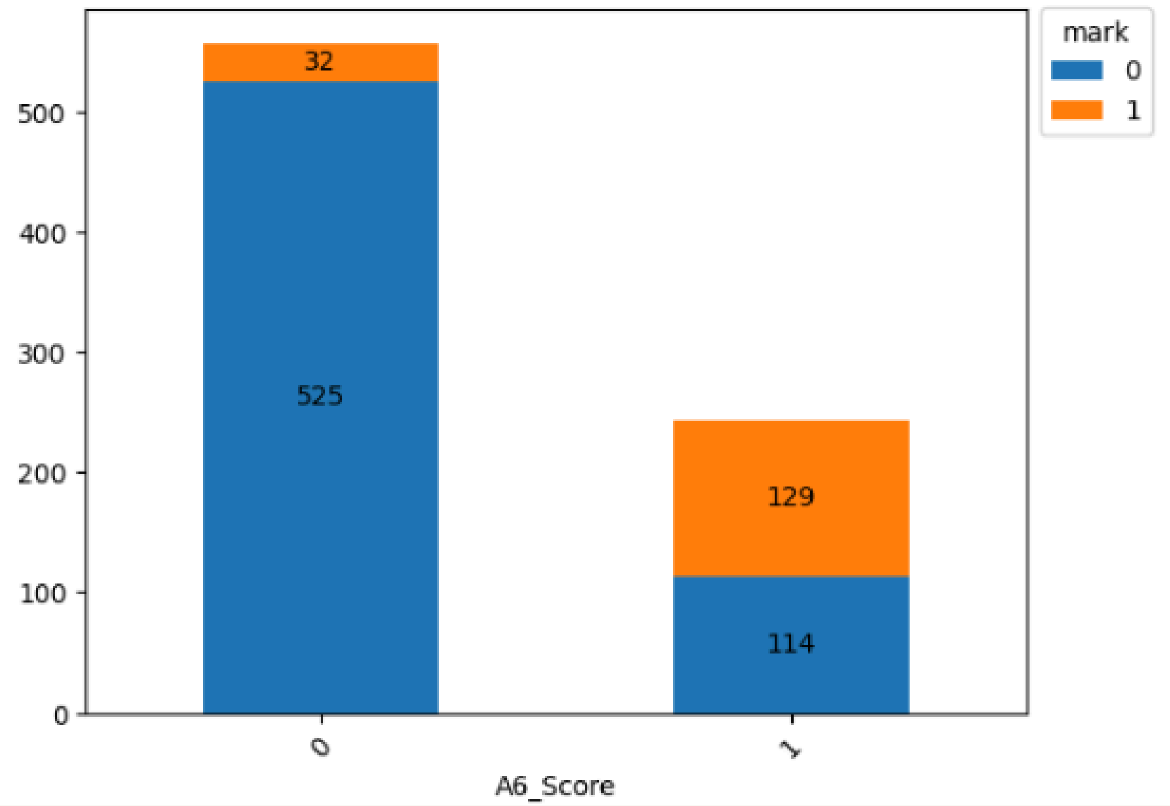
Column name - A4_Score
p value is 4.8840206399536454e-45
Both variables are Dependent (reject H0)



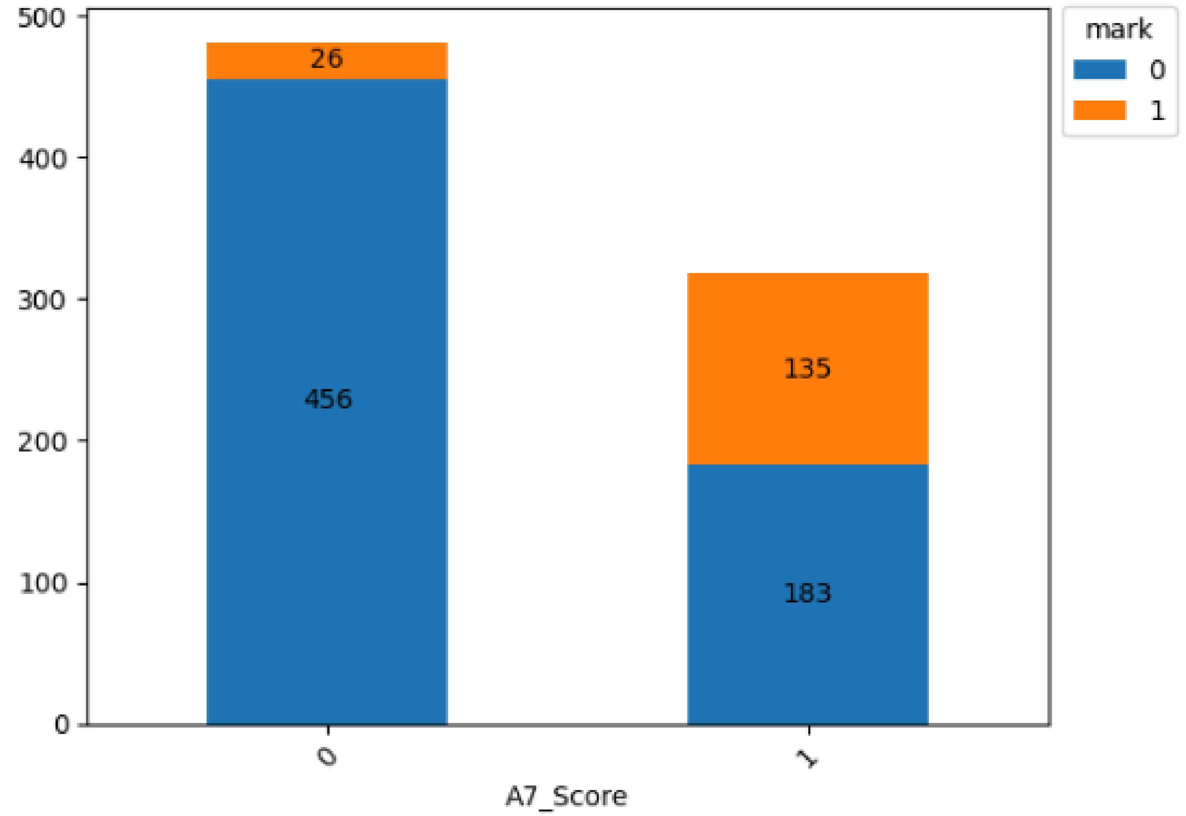
Column name - A5_Score
p value is 1.7931000962761736e-38
Both variables are Dependent (reject H0)

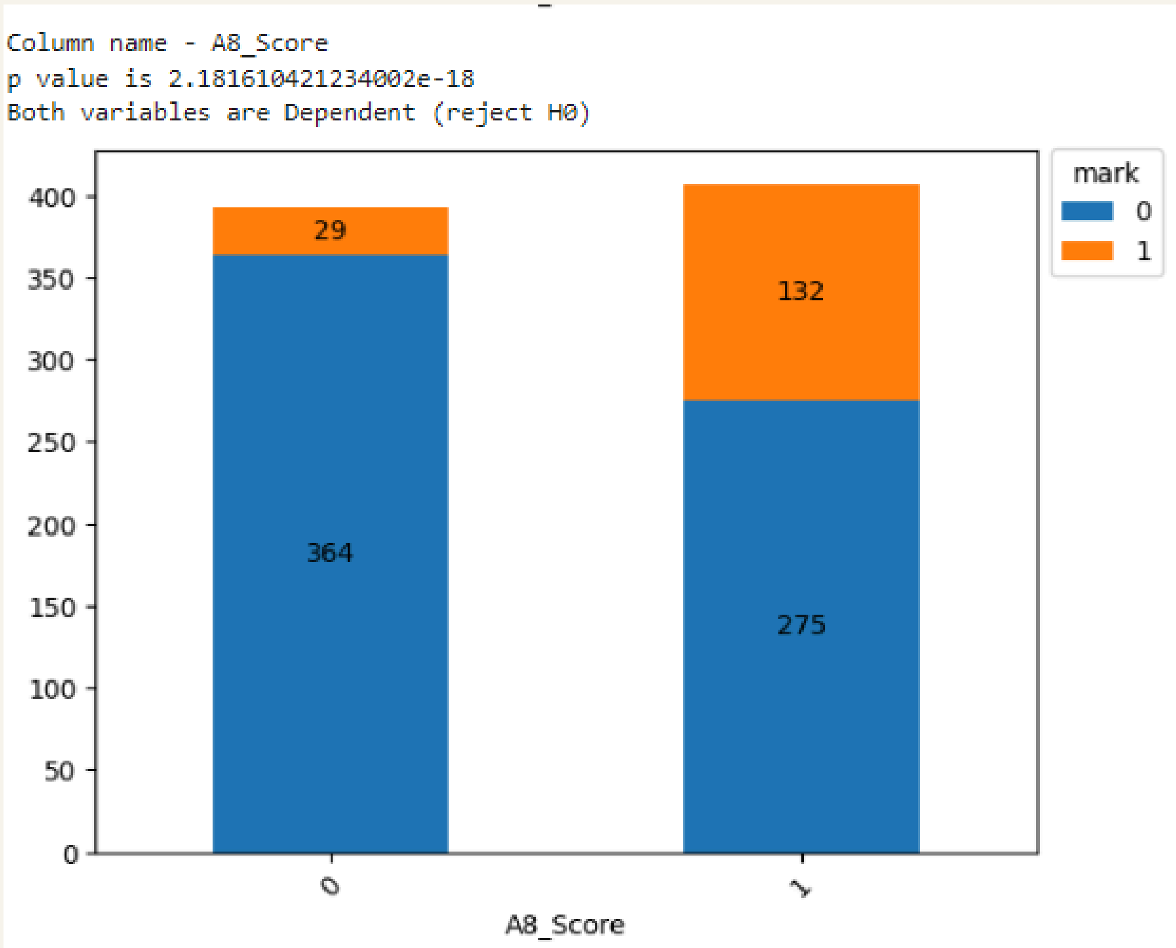


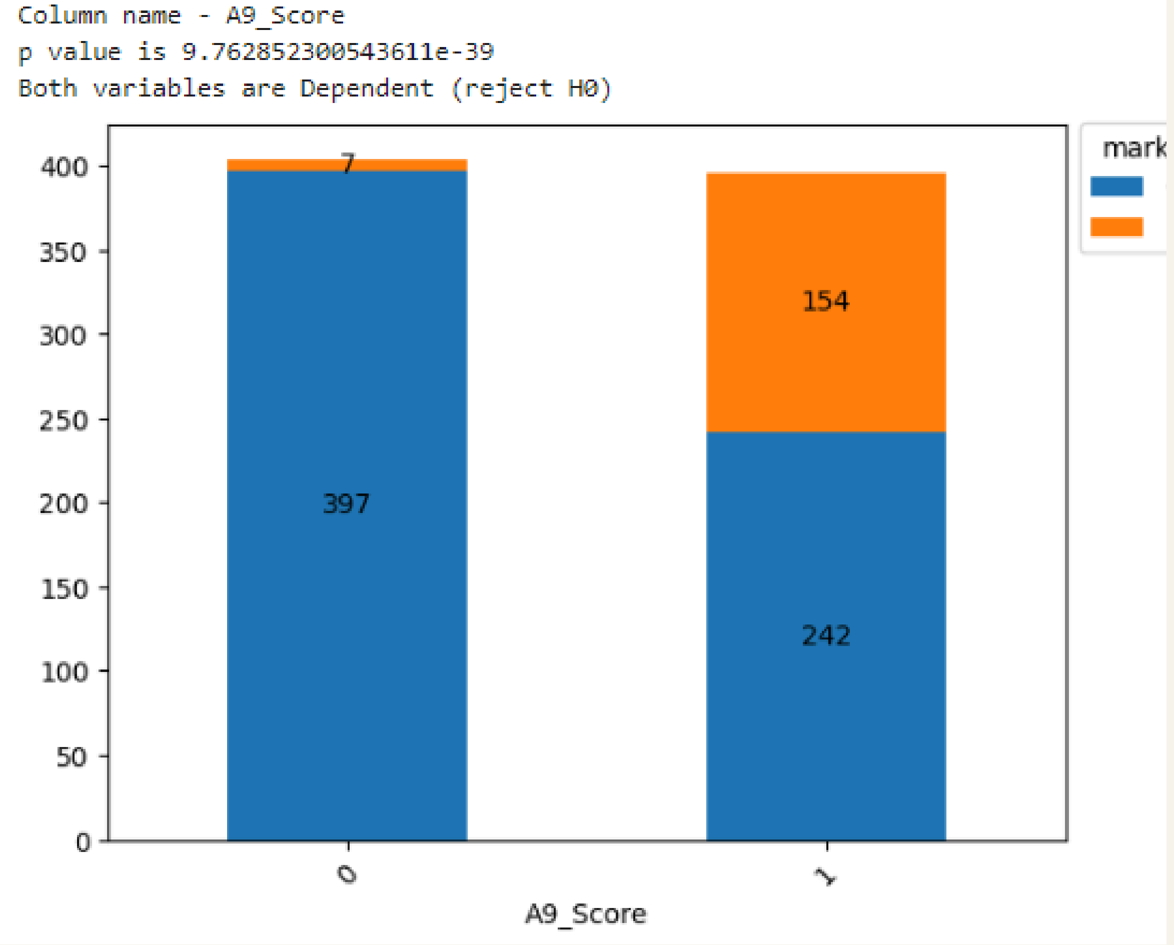
Column name - A6_Score
p value is 1.3536803601031668e-52
Both variables are Dependent (reject H0)

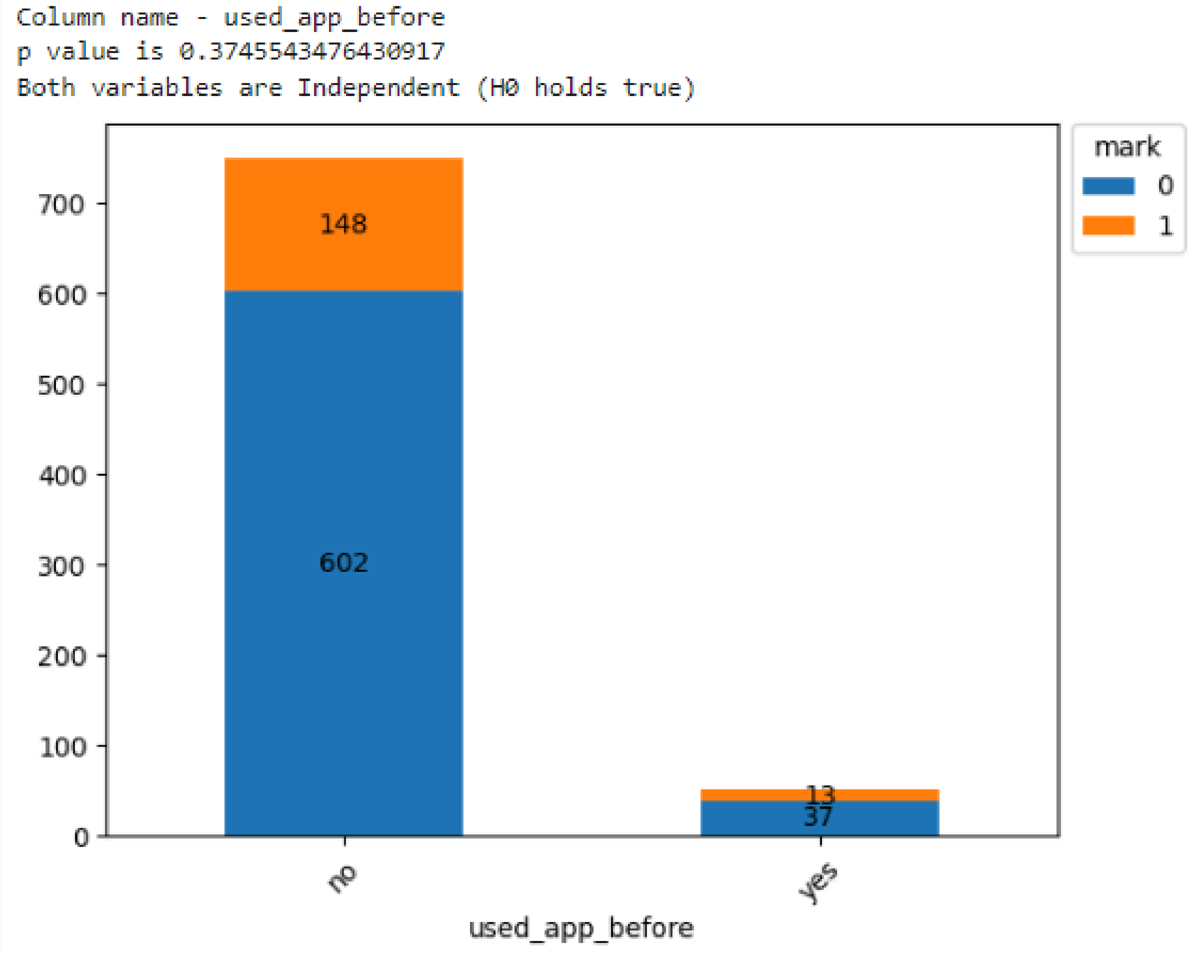


Column name - A7_Score
p value is 5.621312869749746e-37
Both variables are Dependent (reject H0)









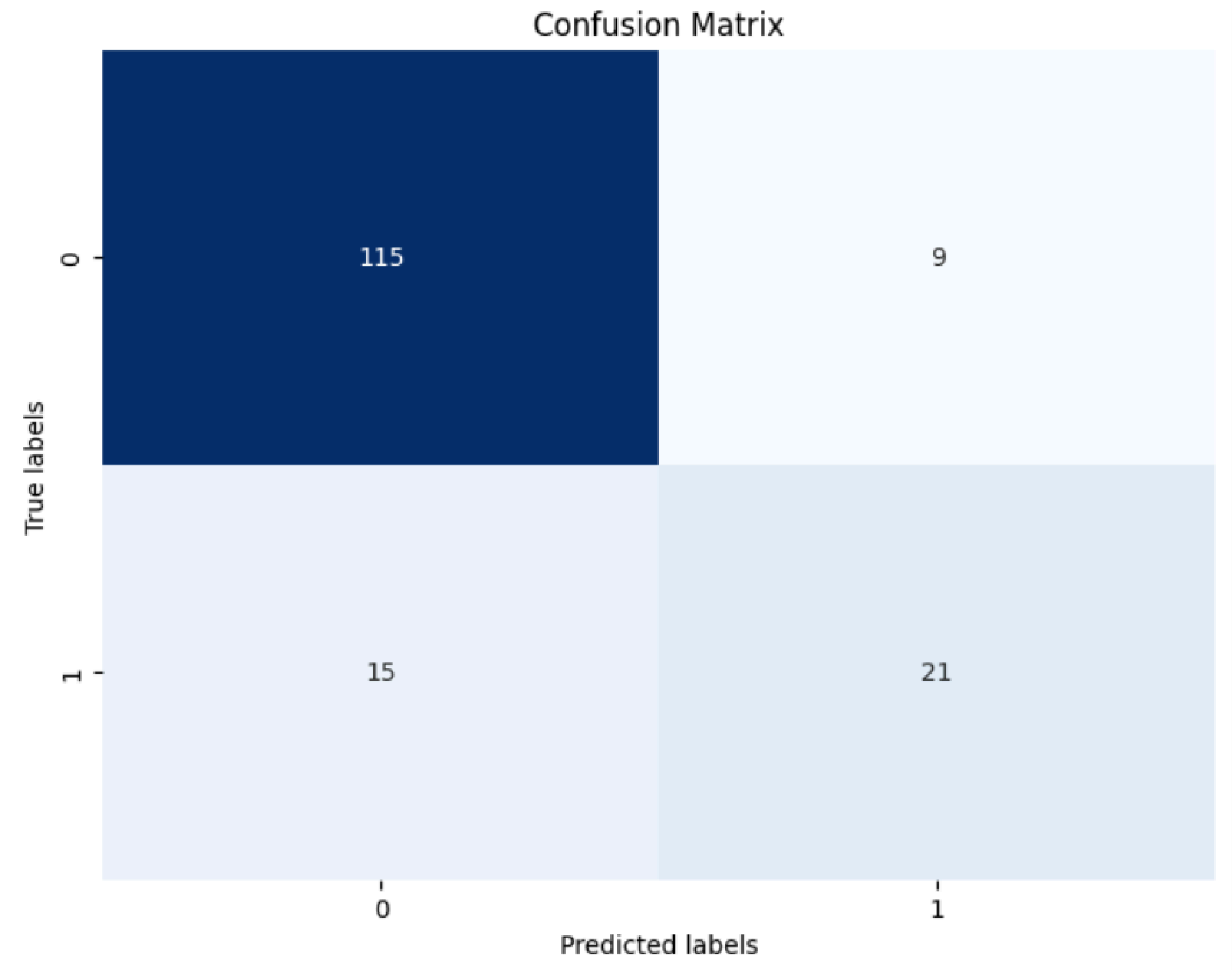
used_app_before is independent from target variable, in Chi Square Test. Hence, dropping this feature.

Hyperparameter Tunning

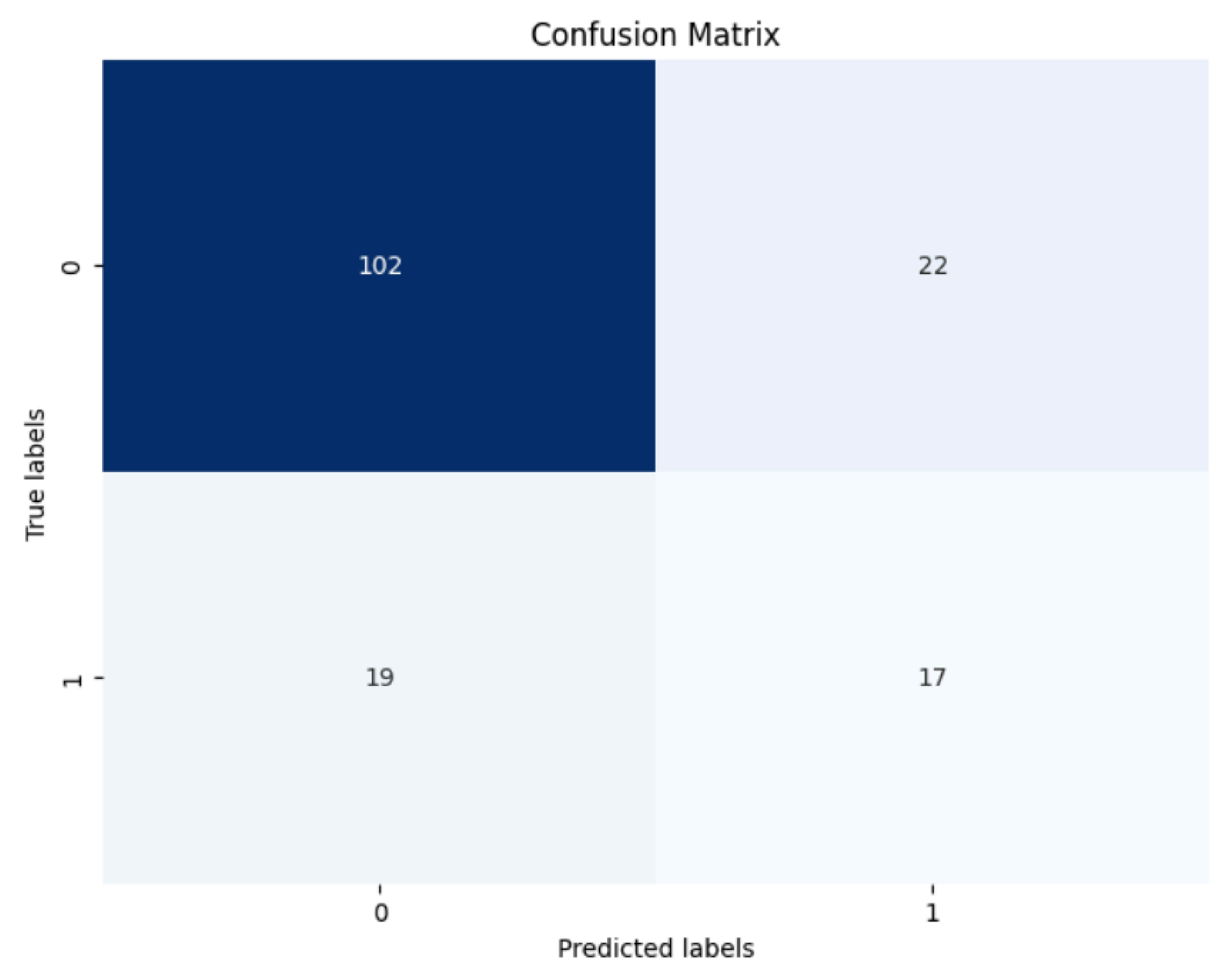
Model Name	Prameters Used	Best Paramter Value return by Grid SearchCV
Logistic Regression	'C': [0.001, 0.01, 0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1], 'penalty': ['l1', 'l2'], # Penalty norm 'solver': ['liblinear']	{'C': 0.3, 'penalty': 'l1', 'solver': 'liblinear'}
Decision Tree Classifier	'max_features': ['auto','sqrt', 'log2'], 'ccp_alpha': [0.1, .01, .001,0.2,0.3,0.4,0.5], 'max_depth' : [1,2,3,5, 6,7,8], 'criterion' :['gini', 'entropy']	'ccp_alpha': 0.01, 'criterion': 'gini', 'max_depth': 5, 'max_features': 'log2'
SVM	'C': [0.1, 0.01, 0.02,0.3,0.4,0.5,0.8], 'gamma': [1, 0.1, 0.01, 0.001, 0.0001], 'kernel': ['poly', 'rbf', 'sigmoid'],	'C': 0.5, 'gamma': 0.01, 'kernel': 'poly'

Applying Models and Plotting Confusion Matrix

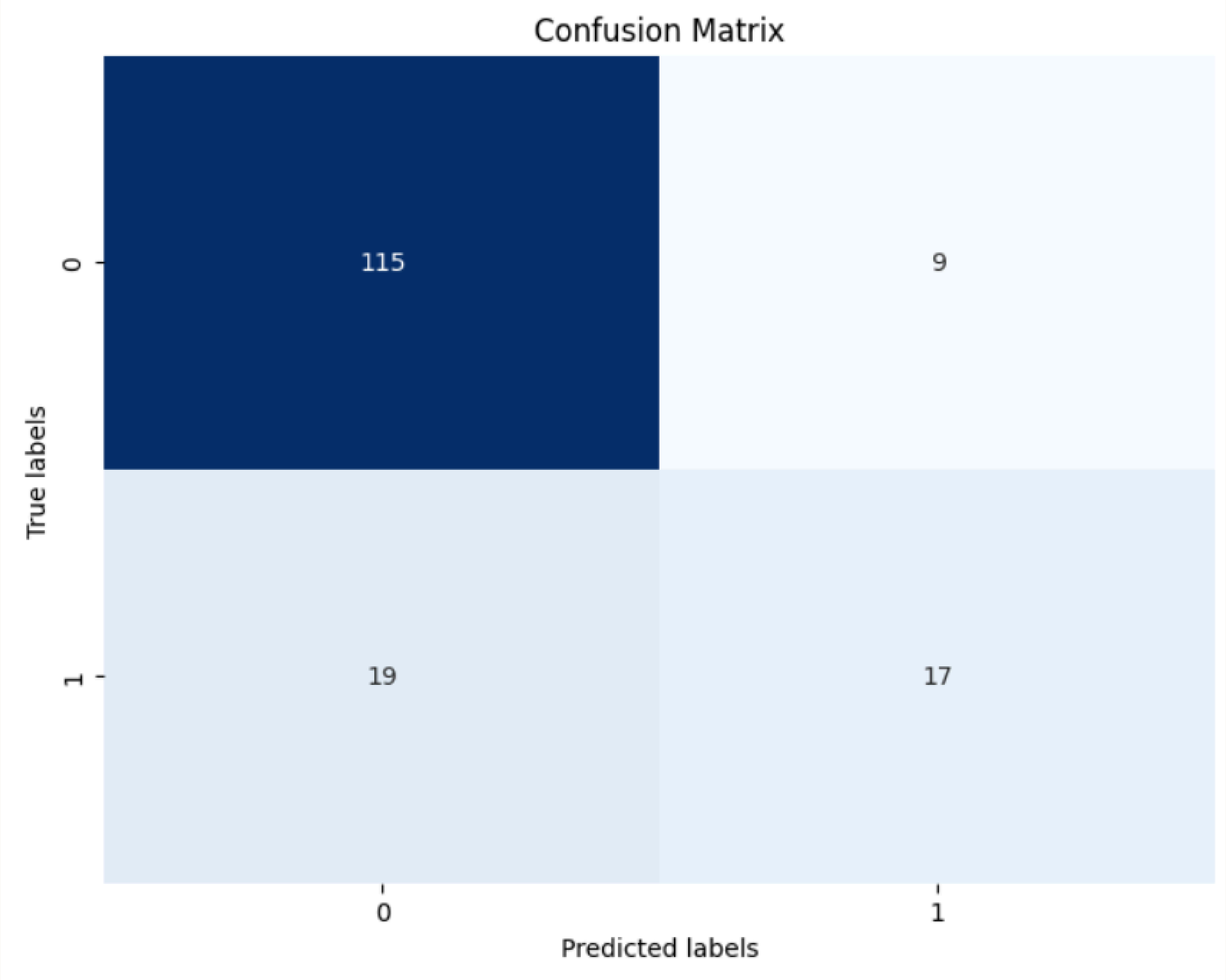
LOGISTIC REGRESSION



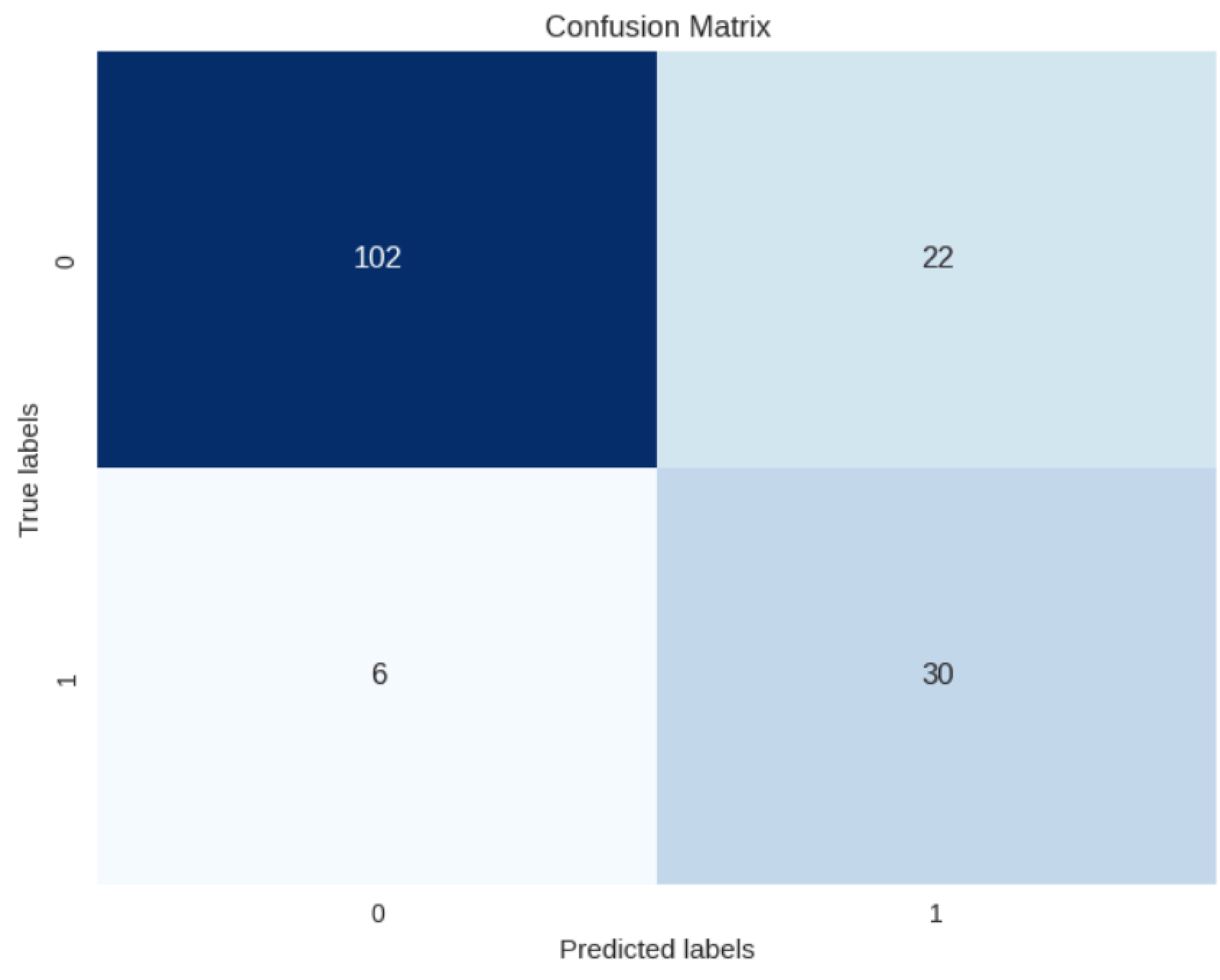
DECISION TREE CLASSIFIER



SVM



Naive Bayes

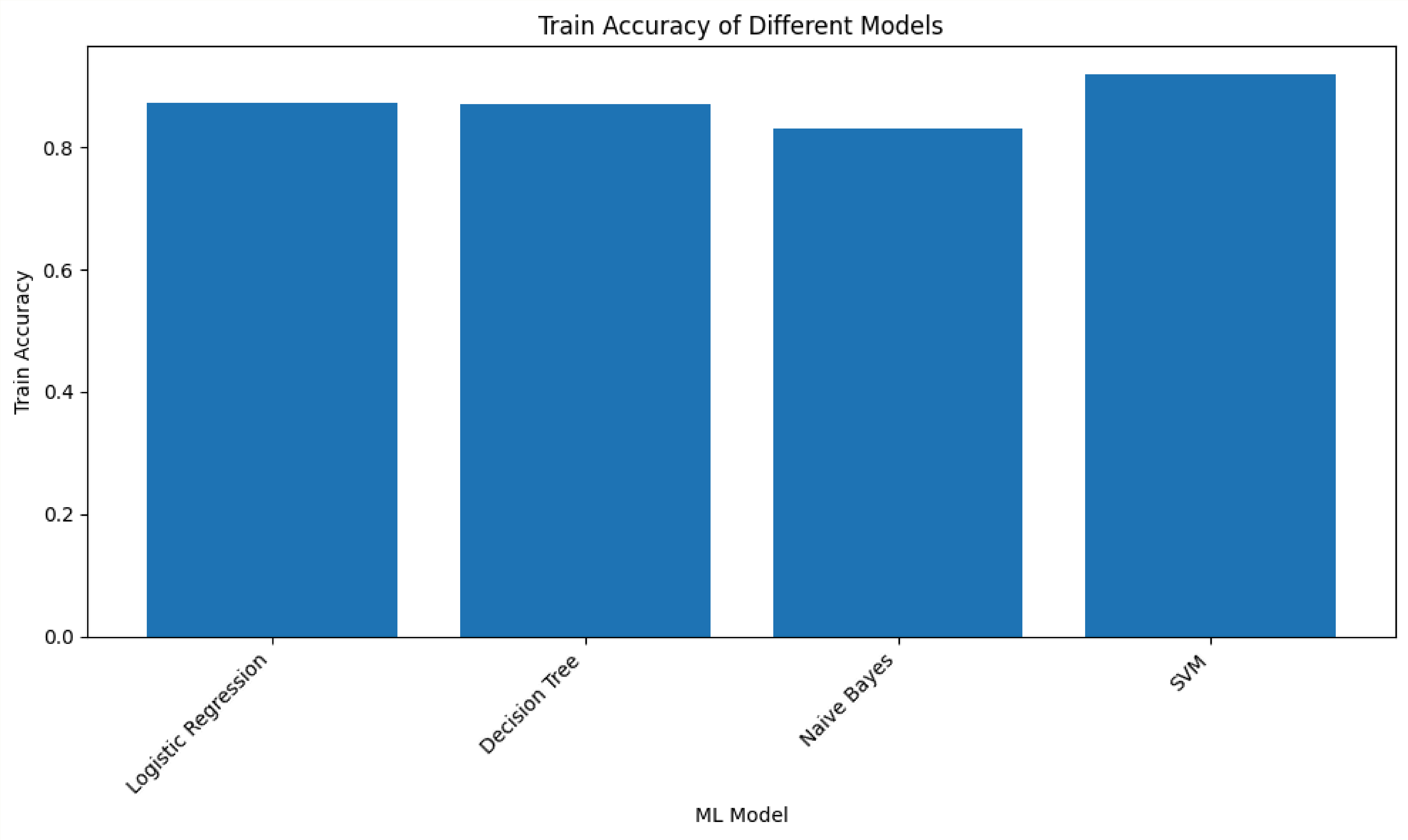


MODEL EVALUATION

Performance Analysis Of The Applied Models

	ML Model	Train Accuracy	Test Accuracy	Precision	Recall	f1 score	ROC_AUC
0	Logistic Regression	0.8734	0.8500	0.7075	0.600	0.6494	0.7699
1	Decision Tree	0.8703	0.8688	0.6721	0.656	0.6640	0.7892
2	Naive Bayes	0.8312	0.8250	0.5436	0.848	0.6625	0.8376
3	SVM	0.9188	0.8250	0.8349	0.728	0.7778	0.8465

Training Accuracy of Different Models



Cross Validation Results

Logistic Regression

Mean: 0.85
Standard Deviation: 0.021
Confidence Interval (95.0%): (0.834, 0.874)

Decision Tree Classifier

Mean: 0.847
Standard Deviation: 0.016
Confidence Interval (95.0%): (0.833, 0.861)

SVM

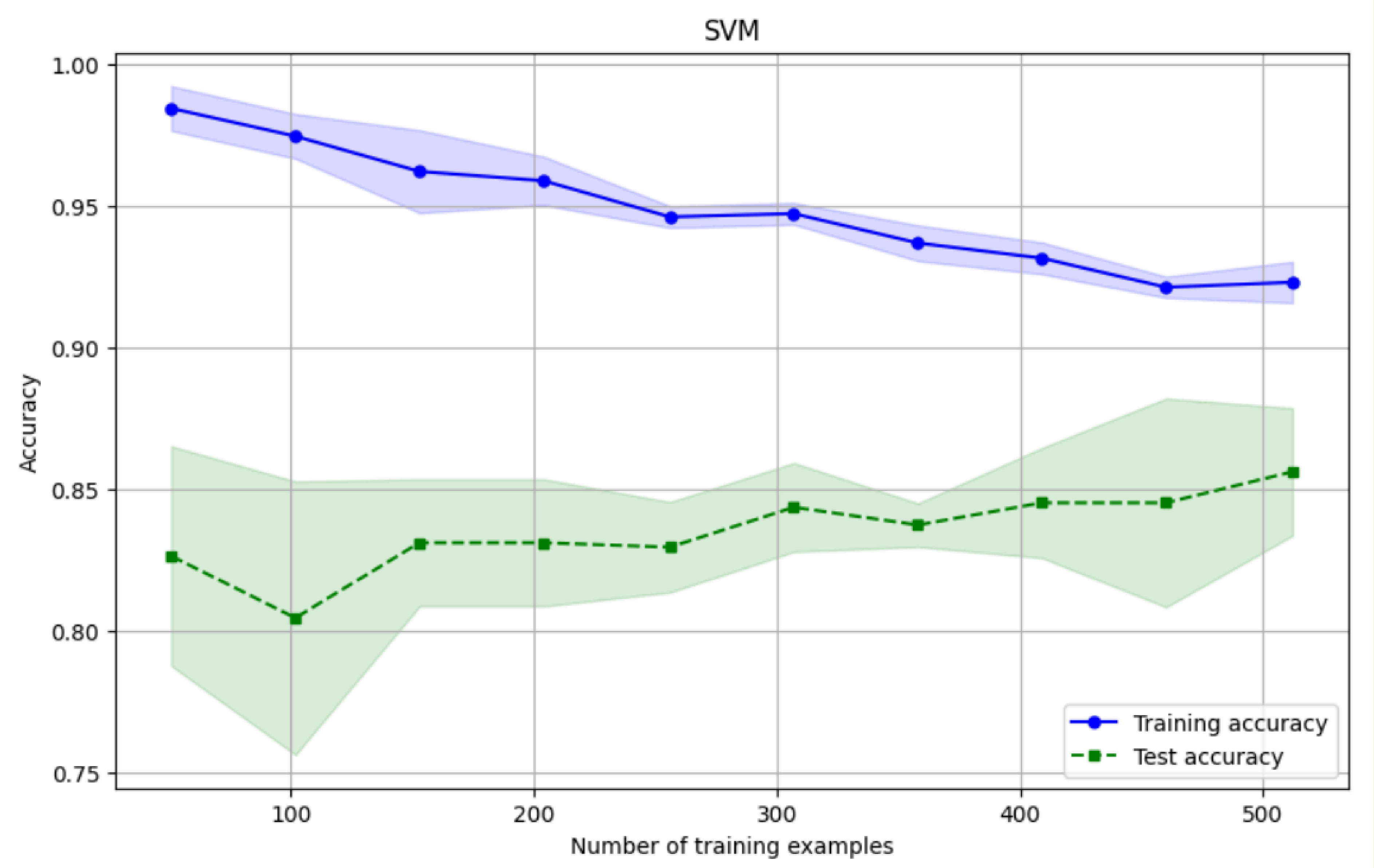
Mean: 0.856
Standard Deviation: 0.022
Confidence Interval (95.0%): (0.837, 0.876)

Naive Bayes

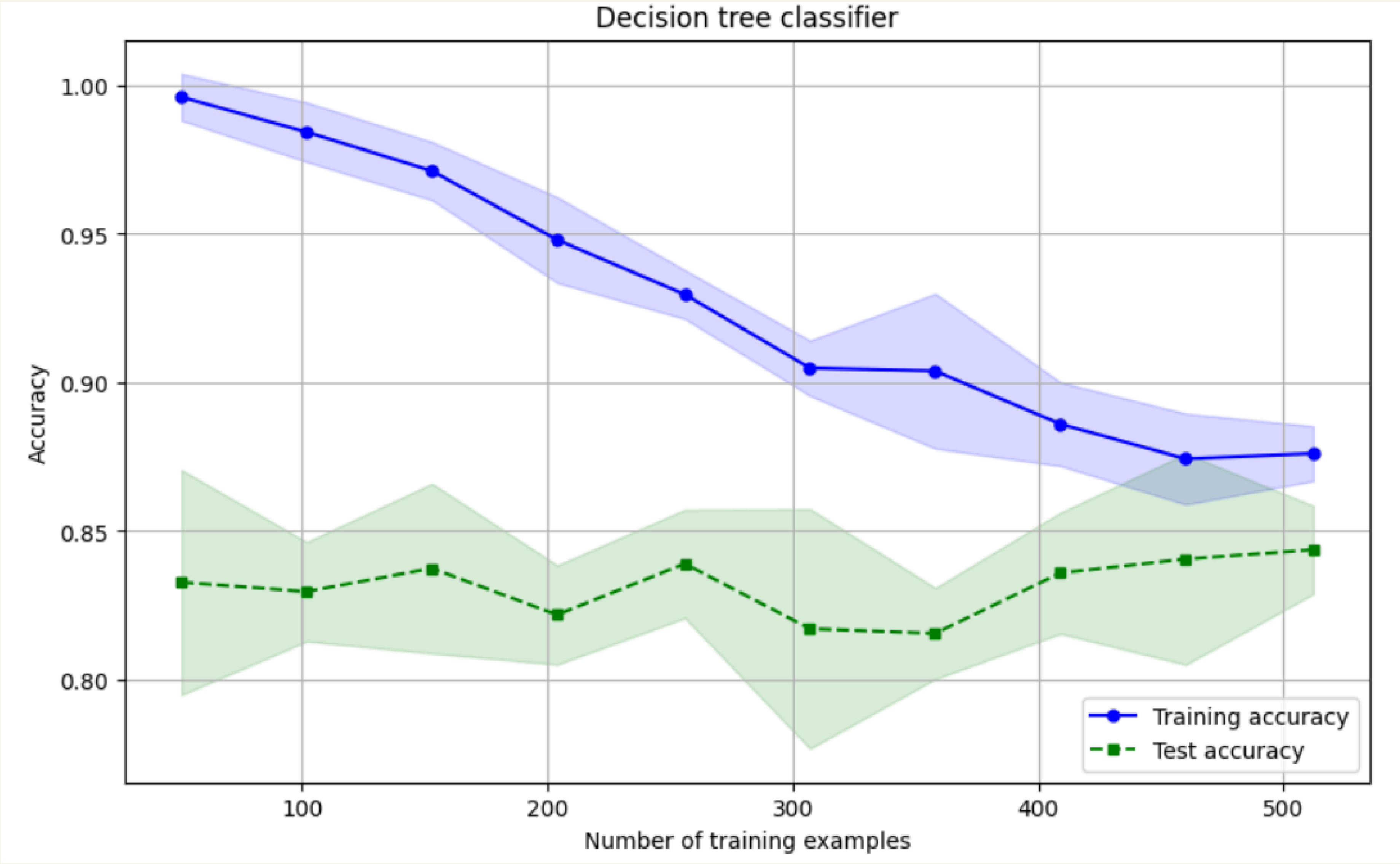
Mean: 0.830
Standard Deviation: 0.022
Confidence Interval (95.0%): (0.810, 0.849)

LEARNING CURVES

SVM

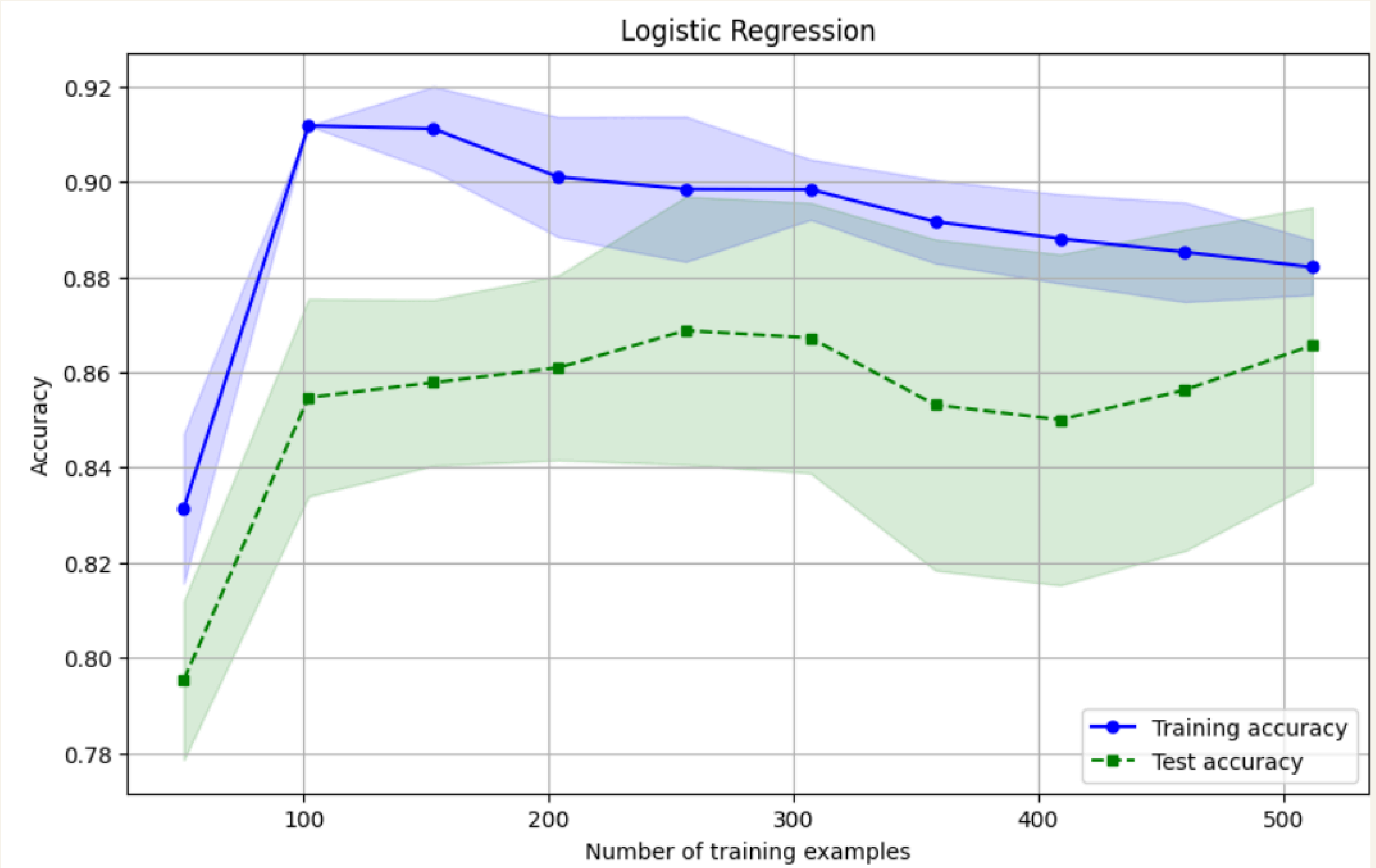


Decision Tree Classifier

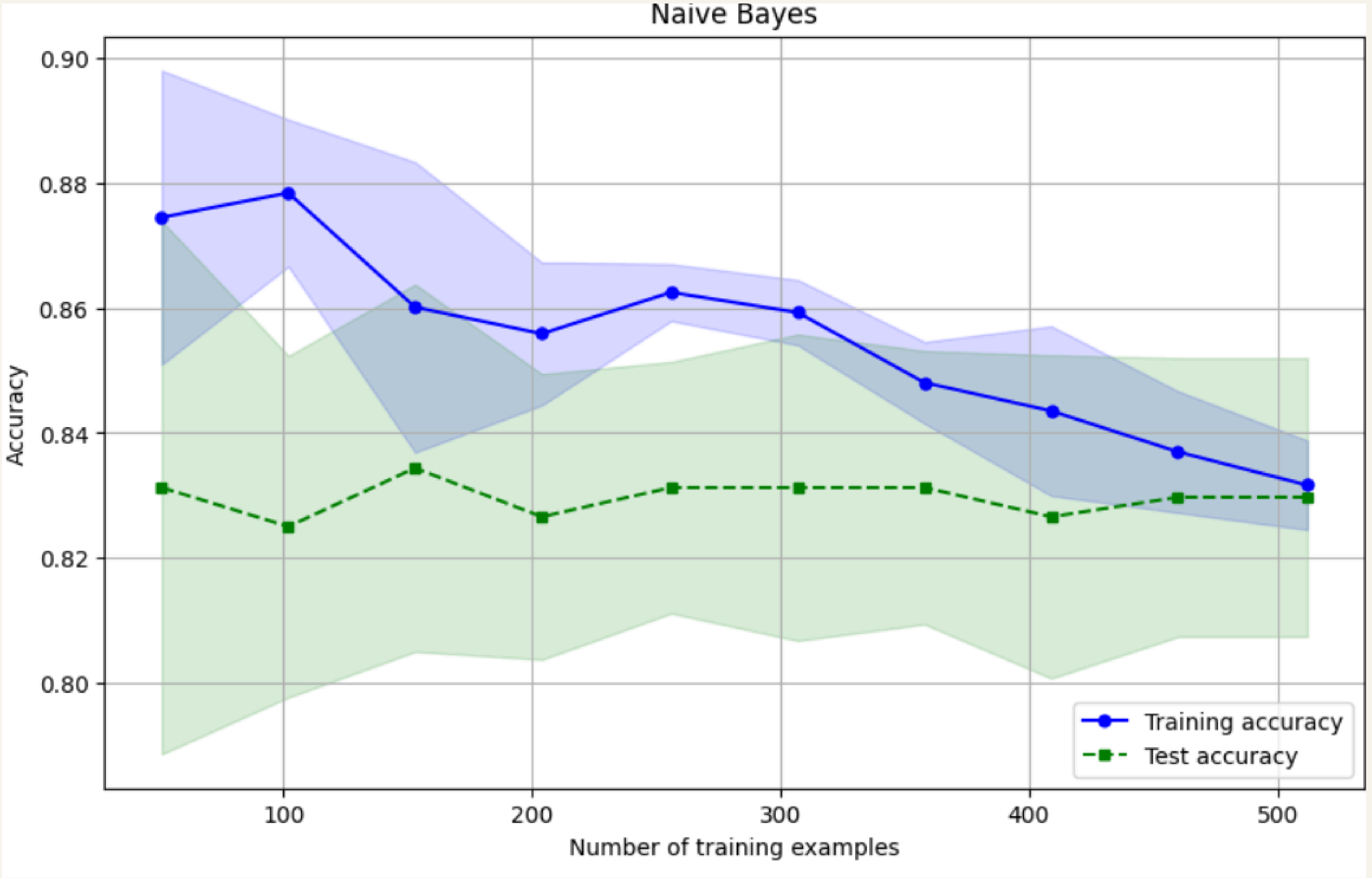


LEARNING CURVES

LOGISTIC REGRESSION



Naive Bayes



Pycaret Results

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.8797	0.9312	0.6500	0.7281	0.6818	0.6082	0.6129	0.9380
ridge	Ridge Classifier	0.8797	0.9244	0.6486	0.7309	0.6791	0.6058	0.6124	0.0620
qda	Quadratic Discriminant Analysis	0.8796	0.8940	0.7292	0.6991	0.7041	0.6296	0.6368	0.0310
lda	Linear Discriminant Analysis	0.8709	0.9241	0.7306	0.6683	0.6938	0.6127	0.6168	0.0320
rf	Random Forest Classifier	0.8618	0.9051	0.5681	0.7017	0.6201	0.5370	0.5461	0.2820
gbc	Gradient Boosting Classifier	0.8596	0.9074	0.5903	0.7061	0.6232	0.5397	0.5550	0.1850
et	Extra Trees Classifier	0.8550	0.9081	0.5681	0.6852	0.6059	0.5192	0.5312	0.1790
lightgbm	Light Gradient Boosting Machine	0.8530	0.9012	0.5681	0.6924	0.6085	0.5200	0.5335	0.5260
xgboost	Extreme Gradient Boosting	0.8508	0.8889	0.5792	0.6914	0.6085	0.5189	0.5347	0.0890
ada	Ada Boost Classifier	0.8482	0.8737	0.5736	0.6449	0.5815	0.4929	0.5089	0.1410
svm	SVM - Linear Kernel	0.8441	0.8739	0.5403	0.5625	0.5126	0.4385	0.4540	0.0560
nb	Naive Bayes	0.8396	0.9255	0.8542	0.5796	0.6832	0.5832	0.6086	0.0340
knn	K Neighbors Classifier	0.8237	0.7959	0.4625	0.5602	0.4937	0.3918	0.4008	0.0400
dt	Decision Tree Classifier	0.8126	0.6944	0.5000	0.5521	0.5038	0.3933	0.4047	0.0520
dummy	Dummy Classifier	0.8036	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0300

Cohen Kappa Statistic

Measure The Performance of Classification Models

Kappa Statistic

$$k = \frac{2 * (TP * TN - FN * FP)}{(TP + FP) * (FP + TN) + (TP + FN) * (FN + TN)}$$

Assess the level of agreement between an actual and predicted

Predicted (rater 2) \ Actual (rater 1)	YES	NO	
YES	45 (**TP**)	15 (**FN**)	60
NO	25 (**FP**)	15 (**TN**)	40
	70	30	

Kappa Score Interpretation

Kappa	**Agreement**
<0	Less than chance agreement
0.01-0.20	Slight agreement
0.21-0.40	Fair agreement
0.41-0.60	Moderate agreement
0.61-0.80	Substantial agreement
0.81-0.99	Almost perfect agreement

$$k = \frac{2 * (45 * 15 - 15 * 25)}{(45 + 25) * (25 + 15) + (45 + 15) * (15 + 15)} = 0.13 \ (13\%)$$

Source : https://bootcamp.uxdesign.cc/cohens-kappa-score-33a0710b2fe0

CONCLUSION

Based on the results of the experiments performed we can conclude that **SVM** gave the best results

With Holdout - Train Accuracy **0.91** and Test Accuracy **0.82**.

With Cross Validation - Mean accuracy **0.856**

Confidence Interval (95.0%): (0.837, 0.876)

REFERENCES

- https://archive.ics.uci.edu/dataset/426/autism+screening+adult
- Analysis and Detection of Autism Spectrum DisorderUsing Machine Learning Technique
- Predictive Modeling for Early Diagnosis of Autism SpectrumDisorder (ASD) using Machine Learning Classifiers
- Machine Learning-Based Models for Early Stage Detection of Autism Spectrum Disorder
- A Review of Machine Learning Methods of Feature Selection and Classification for Autism Spectrum Disorder
- A Review on Predicting Autism Spectrum Disorder(ASD) meltdownusing Machine Learning Algorithm

THANK YOU