LEARN MATE

"Empowering Autism, Unlocking Potential"

Team

Tharaki D.H.D
IT21254970

Ruhunuge R.S.D.P
IT21227486

Rupasinghe Y.S
IT21160820

Pehesarani W.K.A
IT21259470

Mr. Samadhi Chathuranga
Supervisor

Ms. Thisara Shyamalee
Co-Supervisor

Prof. Hemamali Perera
External Supervisor

Background

ASD (Autism Spectrum Disorder) - A neuro-developmental condition of variable severity with lifelong effects that can be recognized from early childhood,

Affects communication, learning, and behavior of a person

Introduction

- Enhance the learning skills of autistic children using machine learning techniques.
- Provide customized recommendation plans on a weekly basis.
- Capture detailed progress of each child.
- Visualize the child's progress through intuitive graphs and charts using color theory.
- Enable parents and teachers to monitor and review the child's development.

Objectives

- Understand How Autistic Children Learn
- Create customized Learning Plans
- Test Plans in Small Studies
- Improve Teaching Tools and Support
- Make sure Plans Work Everywhere
- Protect Autistic Children's Rights

Research Problems

How to enhance autistic children's abilities based on domains of learning with the use of machine learning?

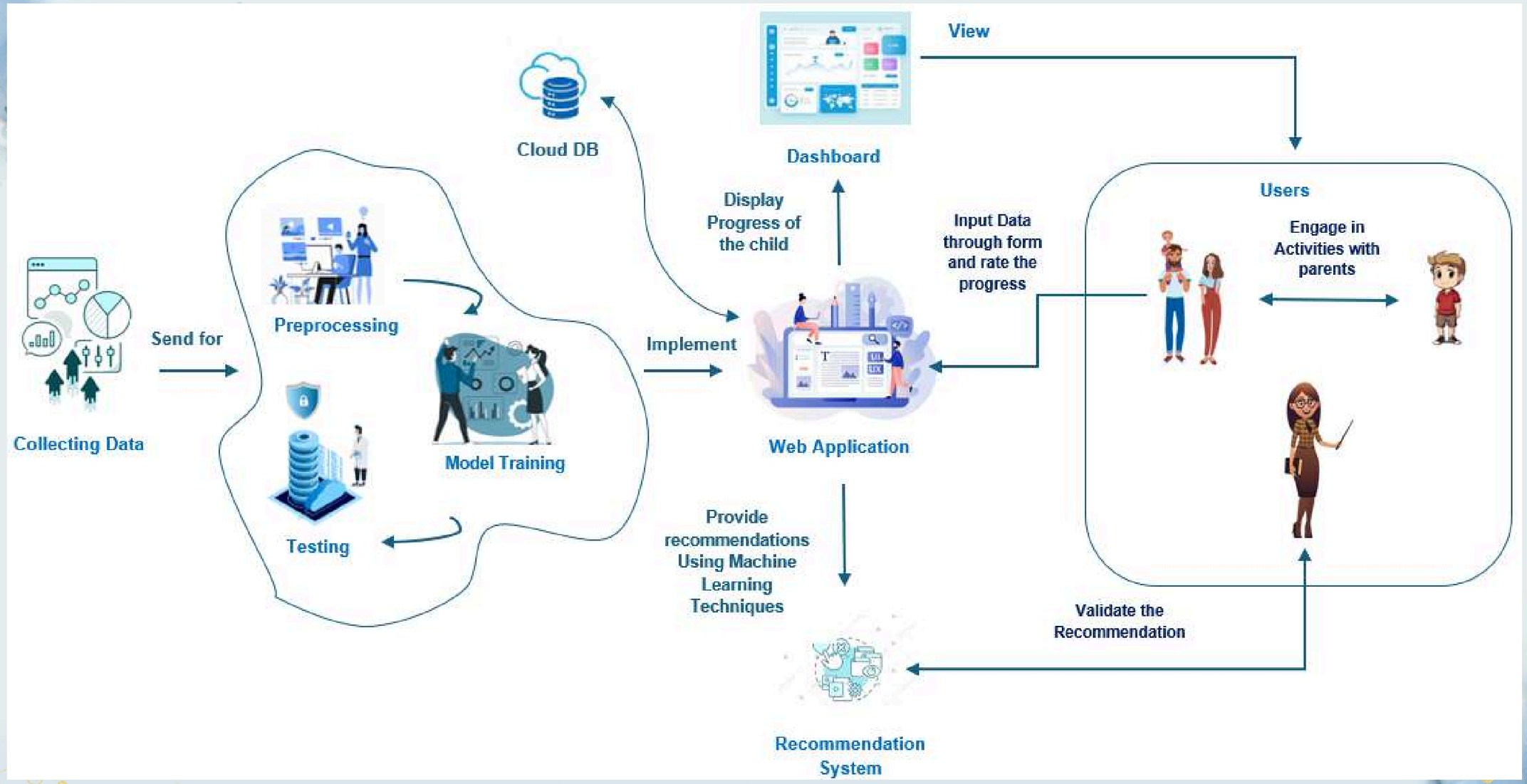
How to create peer-support strategies that foster social interactions and understanding?

Proposed Solutions

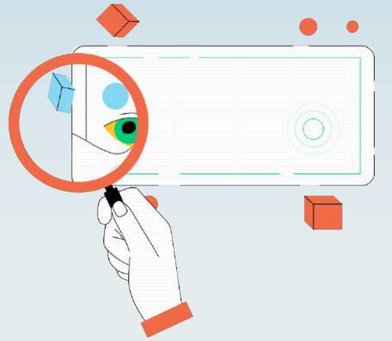
- Integrating Learning Recommendations with Parent-Friendly Dashboards

- Peer Support and Awareness Modules

Solution Architecture



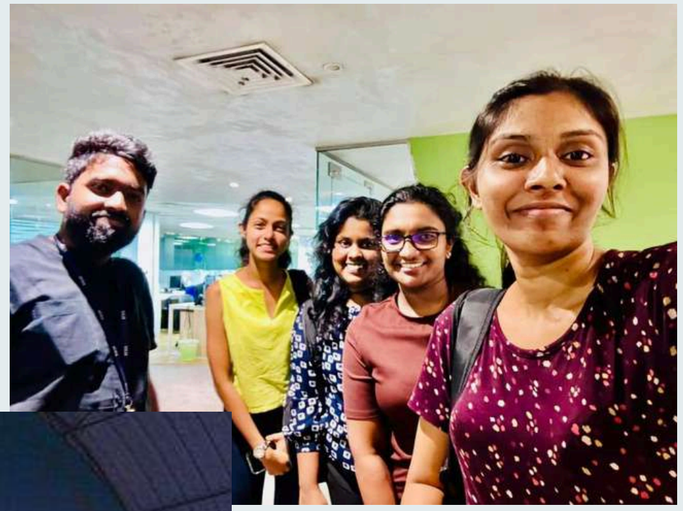
Data Collection

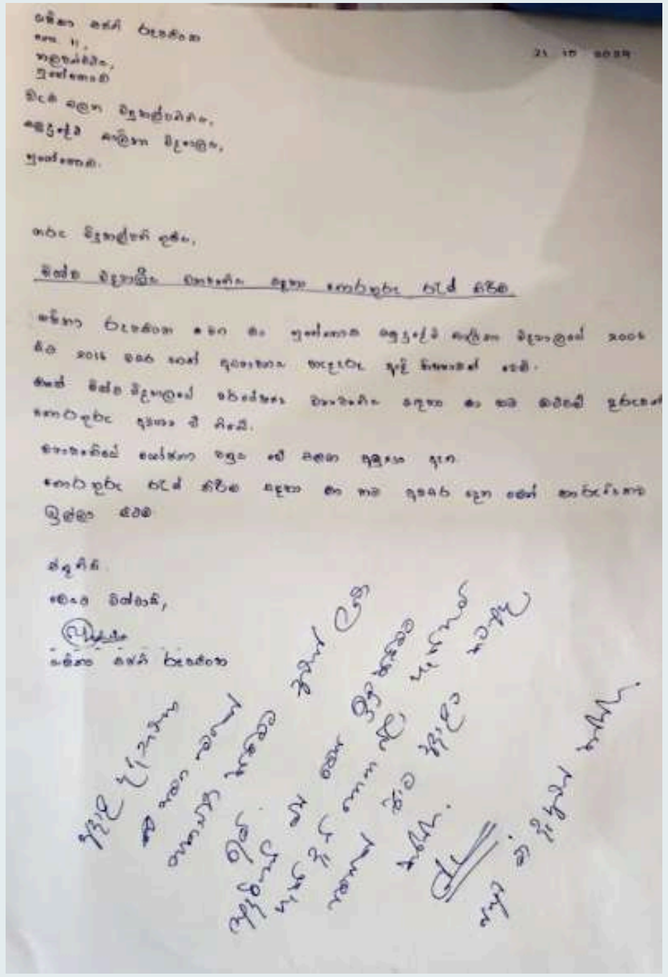


- **Google Form Development**
  - Created a Google Form to collect data about autistic children. Link
  - Questions focused on learning domains
- **Expert Guidance**
  - Form structure and questions were designed with incorporated knowledge and reference documents provided by Mr. Lakmal Ponnamperuma
- **Distribution**
  - Distributed the Google Form through schools and learning centers.
    - Ayati Centre - Ragama
    - Chithralane Centre - Narahenpita
    - Ash Alifaa Centre - Kotikawatta
    - MJF Centre - Moratuwa
    - Sunera Padanma- Homagama
    - Oruwala Central School - Oruwala
    - Samudradevi School - Nugegoda
    - Kottawa Darmapala Primary School - Digana

(No sensitive data was collected during this process)

Field Visits, Discussions & Information Gathering





intern.chitralane <intern.chitralane@gmail.com> Wed, Sep 18, 9:41AM

to me

Dear Dilmi,

Your group may visit our institute on either of the following days,
25th September
26th September
27th September

If a later day is required we can consider a possible date next month. I can arrange resource persons and observation and other requirements for you and get appointments with necessary parties once you confirm that you're attending by replying to this email with the most preferred date specified.

Please note that there is a payment to be made which is 5000/= per person for a project which will be charged from you upon arrival.

further details and instructions could be shared upon confirmation from your end.

Best Regards,

Inuri Wijesuriya
Project Officer
Chitralane Welfare Society

Ayati Research <ayatiresearch@gmail.com> Thu, Oct 10, 3:14 PM

to me

Dear Dilmi,

Please note that you need approval from the ERC Kelaniya (attached to the Medical Faculty) to continue your research at Ayati.
https://medicine.kln.ac.lk/prog/erc/

Thank you
Regards
Research Committee

Meta-Cognitive Domain

Tharaki D.H.D

Introduction

- Awareness and comprehension of one's own thought and learning processes.
- Critical for problem-solving and academic success.
- Increased Independence and Self-assurance and enhances daily activities.
- Supports social and personal development, aiding in overcoming obstacles.

Knowledge

Regulations

Background

- Target Problem: Insufficient awareness among peers and parents regarding effective strategies to impart knowledge and establish appropriate regulations for supporting the development of autistic children

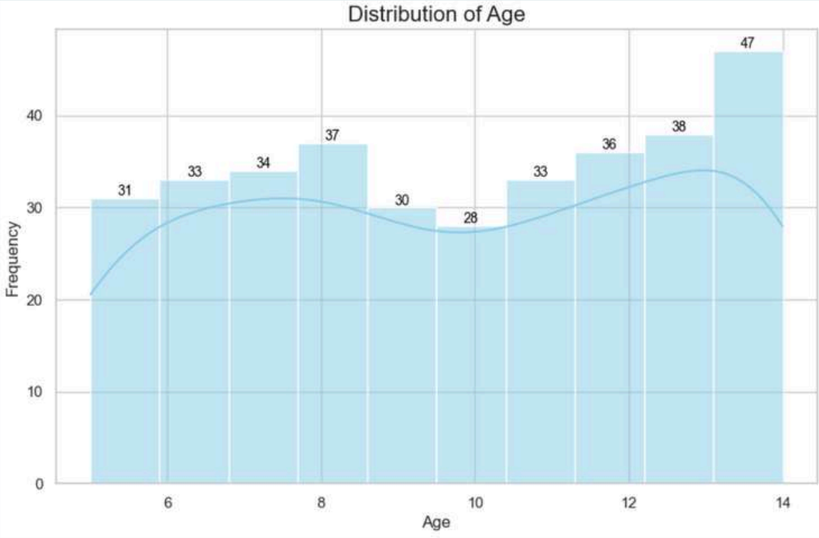
- Proposed Solution: Use classification methods to categorize autistic children according to the metacognitive level and provide customized activity lists, to be done with the help of parents and caregivers.

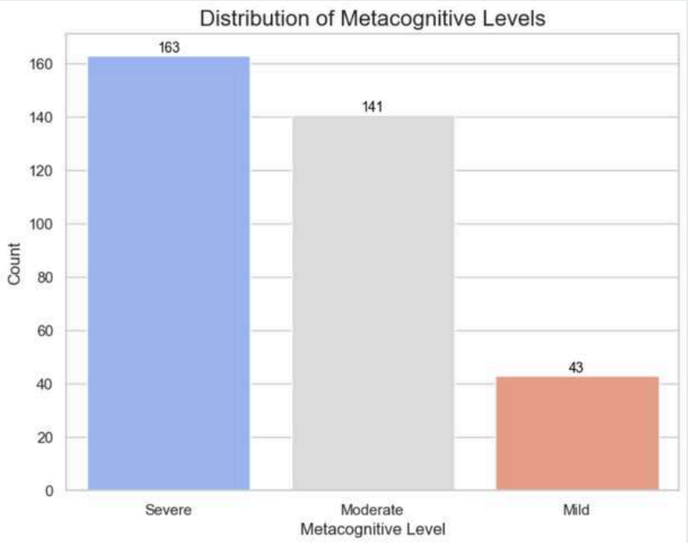
Dataset

- Features:
  - Demographic : Gender, Age, Family history (If a family member has ASD)
  - Other features:
    1. Concentration
    2. Independent Task Planning
    3. Remember Steps
    4. Finish Chores Propoerly
    5. Identify Goals
    6. Recognize Mistakes
    7. Prioritize Tasks
- Target variable: Cognitive Level; mild, moderate & severe
- Data volume:
  - Total records: 347
  - Number of features/columns: 11 features

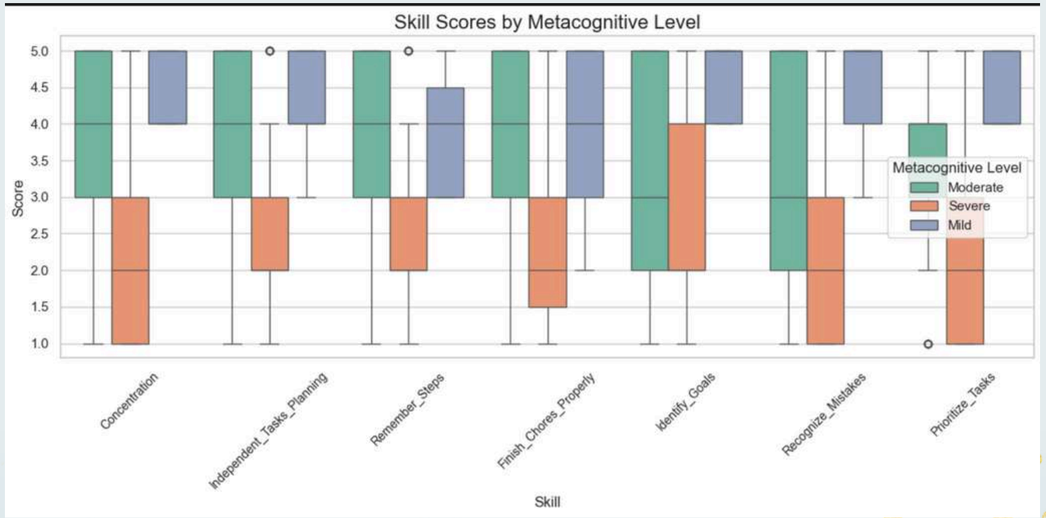
Gender	Age	Family_History	Concentration	Independent_Tasks_Planning	Remember_Steps	Finish_Chores_Properly	Identify_Goals	Recognize_Mistakes	Prioritize_Tasks	Metacognitive_Level
Female	13	Yes	4	5	2	5	3	5	1	Moderate
Female	9	No	1	4	2	5	5	3	4	Moderate
Female	10	No	3	3	3	5	2	3	4	Moderate
Male	5	Yes	2	5	1	5	5	5	4	Moderate
Male	12	Yes	3	3	5	3	3	3	4	Moderate
Male	6	Yes	1	3	3	1	4	1	2	Severe
Female	12	No	4	3	5	4	2	4	4	Moderate
Female	14	Yes	4	2	5	2	1	1	4	Severe
Male	6	No	1	2	3	3	1	5	4	Severe
Female	6	Yes	2	2	1	3	4	2	2	Severe
Male	8	Yes	4	3	3	4	5	4	4	Mild

Exploratory Data Analysis





Distribution of the dataset according to different attributes



Data Preprocessing

```python
# Define mappings for encoding
mappings = {
    'Gender': {'Male': 0, 'Female': 1},
    'Family_History': {'Yes': 1, 'No': 0},
}

def apply_mapping_encoding(df, mappings):
    for column, mapping in mappings.items():
        if column in df.columns:
            df[column] = df[column].map(mapping)
    return df

df1 = apply_mapping_encoding(df, mappings)
```

Encoding Categorical Variables

```python
# Function to handle missing values
def handle_missing_values(df):
    for column in df:
        if df[column].dtype == 'object':
            df[column].fillna(df[column].mode()[0], inplace=True)
        else:
            df[column].fillna(df[column].median(), inplace=True)
    return df

df = handle_missing_values(df)
```

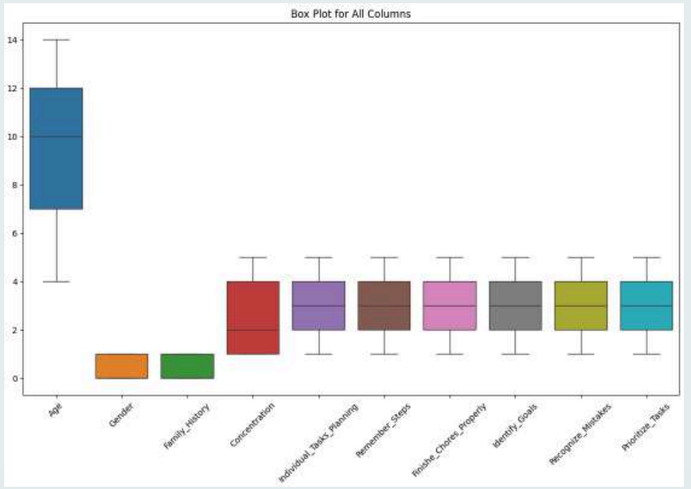
Handle Missing Values

```python
#Balancing the dataset
smote = SMOTE(random_state=42)
X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)
print(f"Train size: {len(X_train_smote)}")
print(f"Test size: {len(X_test)}")
```

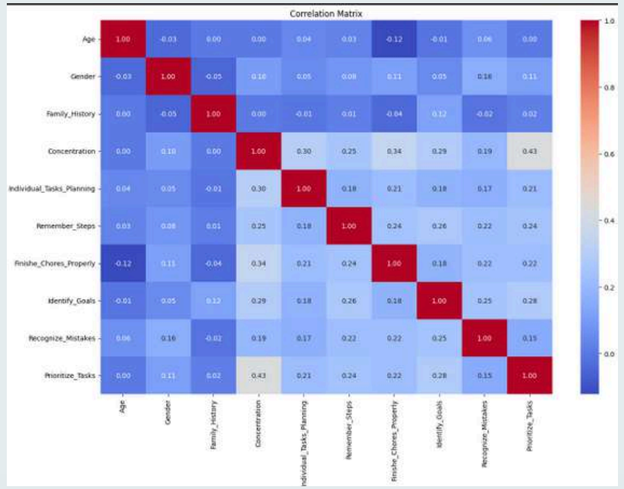
Balancing the Classes

```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
```

Dataset Split



Outlier Boxplot



Correlation Matrix

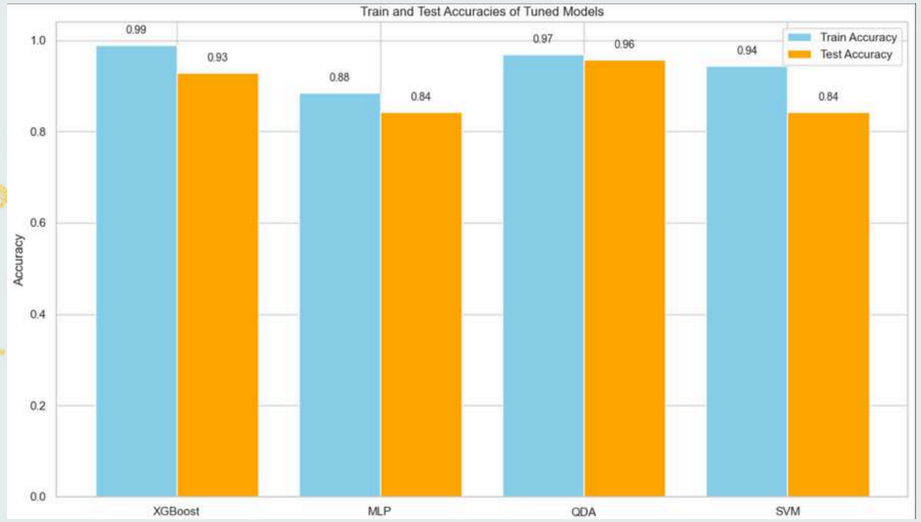
Model Building

- Classification models developed:
  - Extreme Gradient Boosting (XGBoost)
    - Efficient for small datasets, handles missing values, and identifies important features, making it ideal for structured tabular data.
  - Multi-Layer Perceptron (MLP)
    - Captures complex non-linear relationships between features and metacognitive levels, useful if data patterns are not strictly linear
  - Quadratic Discriminant Analysis (QDA)
    - Assumes Gaussian-distributed features for each class, making it effective for small datasets with well-separated groups.
  - Support Vector Machine (SVM)
    - Works well with small, high-dimensional datasets by finding optimal decision boundaries

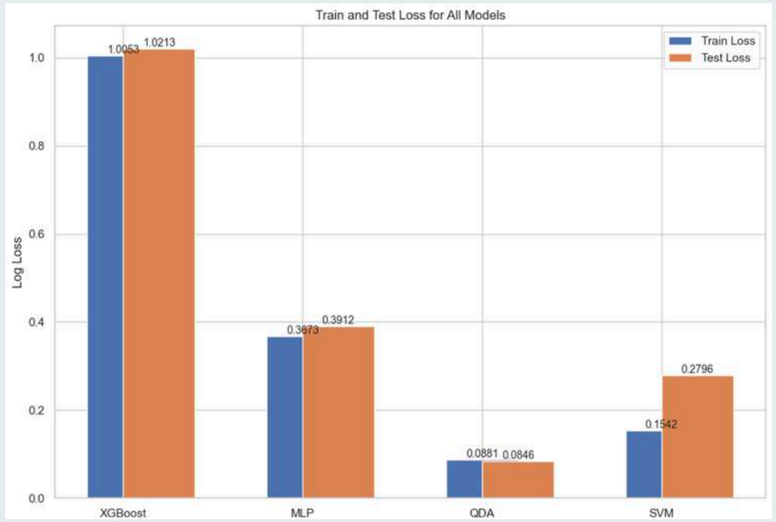
- Compare the accuracy and performances of each method and select the best model for the dataset. (using Train Accuracy, Test Accuracy, Precision, Recall, F1 Score)

	Model	Train Accuracy	Test Accuracy	Precision	Recall	F1 Score
0	XGBoost	0.989744	0.928571	0.930556	0.928571	0.928681
1	MLP	0.884615	0.842857	0.852586	0.842857	0.843610
2	QDA	0.969231	0.957143	0.962758	0.957143	0.957465
3	SVM	0.943590	0.842857	0.873094	0.842857	0.849110

Performce Metrices



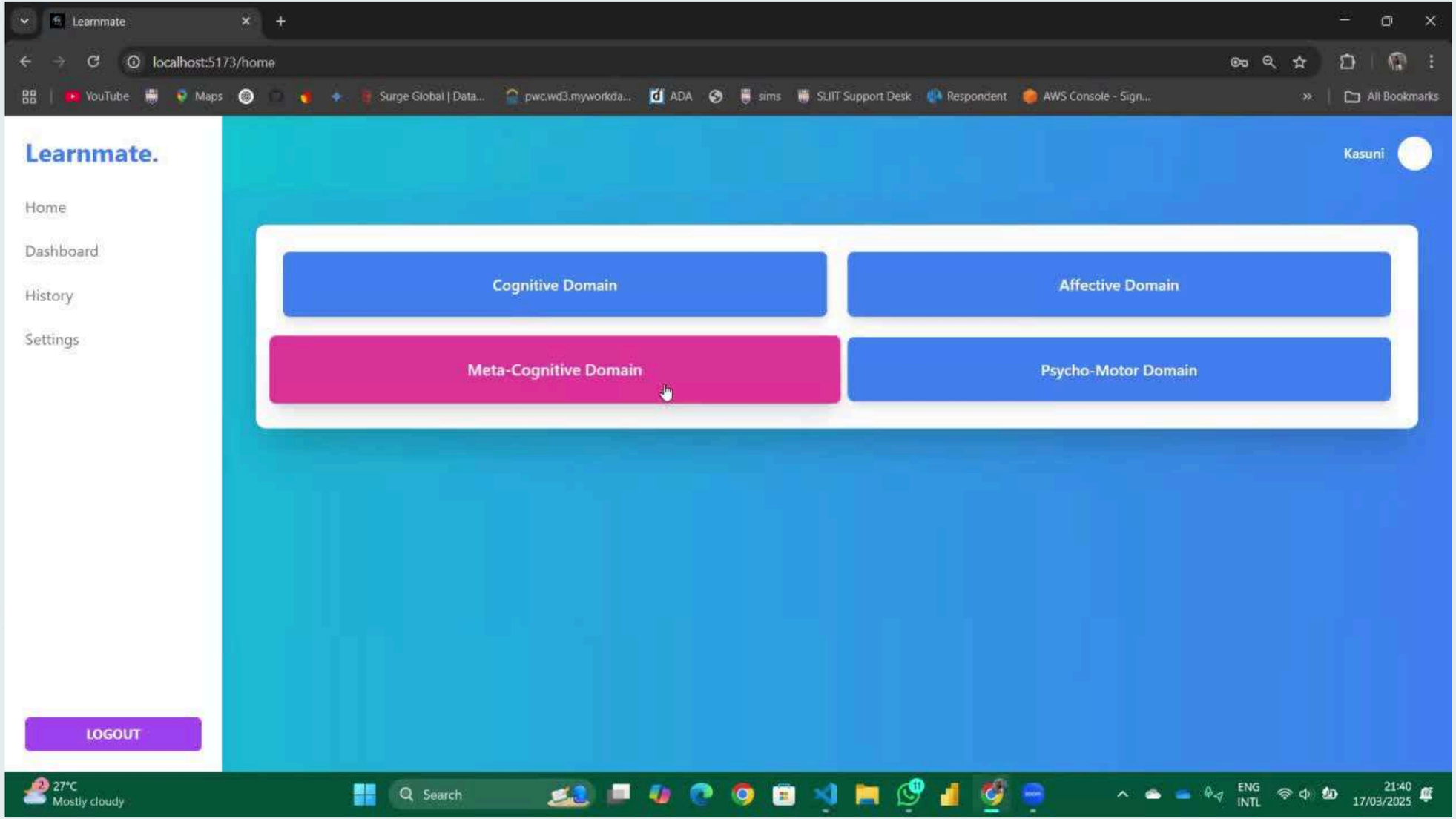
Train-Test Accuracies



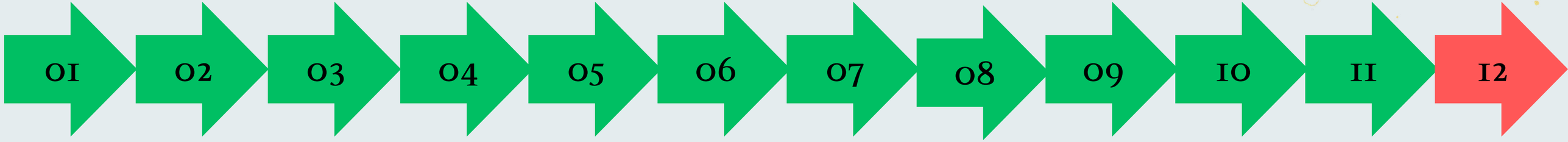
Train-Test Log Loss

Best Model is QDA

Demo - Metacognitive Domain



Project Completion



01. Background Search
02. Data Gathering
03. Data Analysis & Pre-processing
04. Model Building
05. Best Model Selection
06. Define recommendations according to level and age.
07. Metacognitive Domain Cluster Prediction
08. Customized Recommendations
09. Progress Tracking
10. Building a web app & integrating the models
11. Testing
12. Deployment and releasing to the required institutes

Completed 90%

To do 10%

Cognitive Domain

Rupasinghe Y. S.

Introduction

- Development of intellectual skills such as critical thinking, problem solving and creating a knowledge base.

- 6 levels in cognitive domain:
  - Knowledge
  - Comprehension
  - Application
  - Analysis
  - Synthesis
  - Evaluation

- Cognitive Domain Skills:
  - Problem Solving Skills
  - Error Monitoring Skills
  - Knowledge Base

Background

- Target Problem: Peers and parents face difficulties when effectively interacting with children with ASD due to insufficient knowledge regarding effective strategies and establish appropriate regulations for supporting the development of autistic children

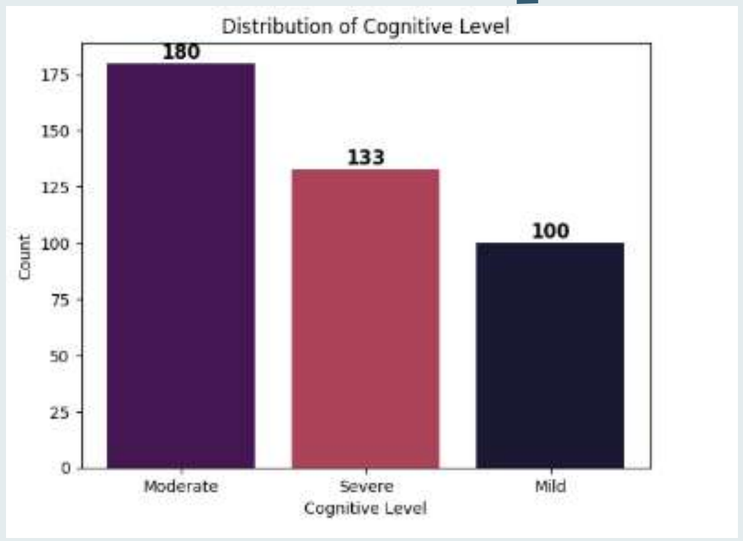
- Proposed Solution: Use classification models to categorize autistic children and provide customized activity lists, to be done with the help of parents and caregivers.

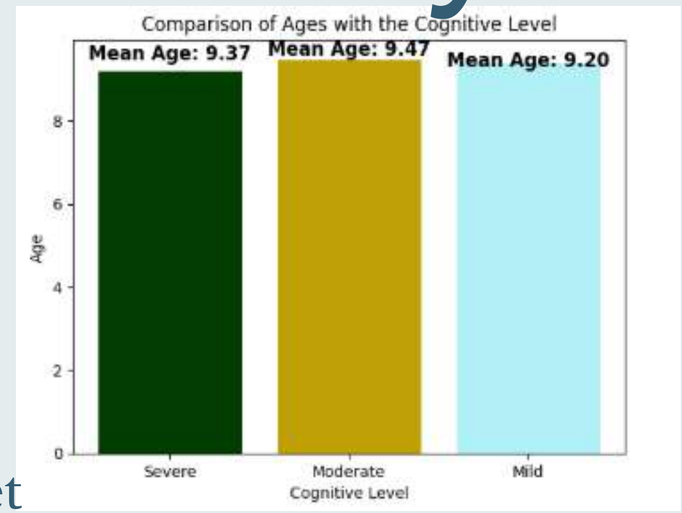
Dataset

- Features:
  - Demographic : Gender, Age, Family history (If a family member has ASD)
  - Other features:
    1. Problem Solving
    2. Visual Learning Preference
    3. Response to Guidance
    4. Task Independence
    5. Object Identification
    6. Error Correction
- Target variable: Cognitive Level; mild, moderate & severe
- Data volume:
  - Total records: 407
  - Number of features/columns: 08 features

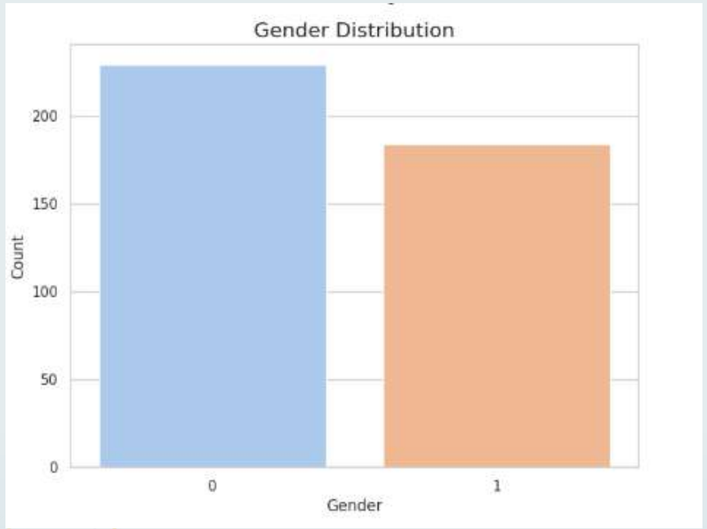
	A	B	C	D	E	F	G	H	I	J
1	Gender	Age (5-14)	Family History	Problem Solving	Visual Learning Preference	Response to Guidance	Task Independence	Object Identification	Error Correction	Cognitive Leve
2	Female	8	No	4	4	3	3	2	1	Moderate
3	Female	14	Yes	2	3	1	3	2	2	Severe
4	Female	10	No	1	1	1	3	2	1	Severe
5	Male	12	No	1	4	2	3	1	1	Severe
6	Male	13	No	4	4	3	5	5	3	Mild
7	Male	12	Yes	2	3	3	3	3	3	Moderate
8	Male	7	Yes	1	3	1	2	2	4	Severe
9	Female	12	No	2	4	5	3	5	5	Mild
10	Male	6	Yes	4	2	5	4	4	4	Mild
11	Male	8	Yes	4	4	2	4	5	4	Mild
12	Female	13	Yes	5	2	1	3	2	2	Severe
13	Male	12	No	3	2	1	3	2	2	Severe
14	Male	13	Yes	1	1	3	1	4	3	Severe
15	Male	8	No	3	3	4	2	2	3	Moderate

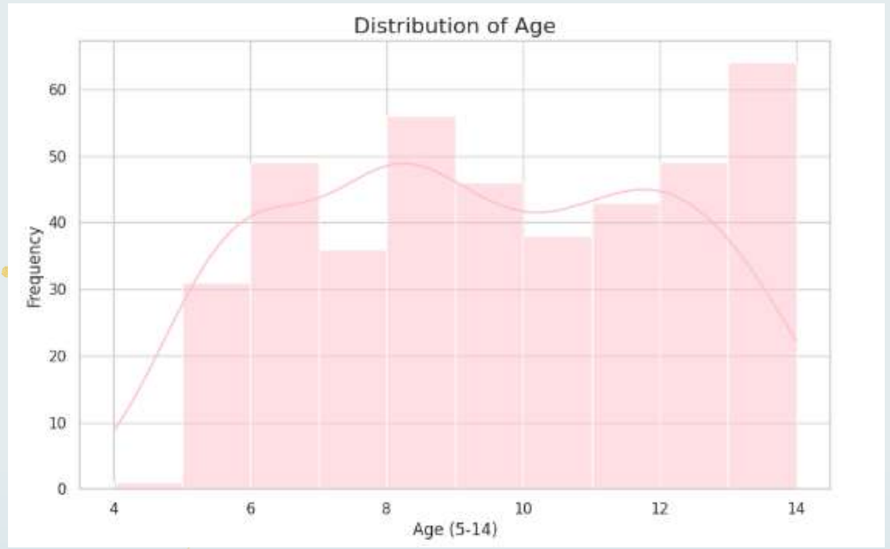
Exploratory Data Analysis





Distribution of the dataset based on different variables





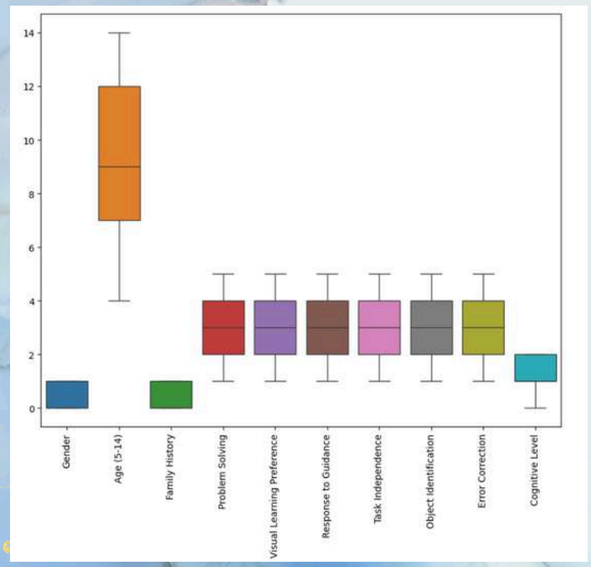
Data Preprocessing

```
#check for null valuess
df.isnull().sum().sort_values(ascending=False)
```

	0
Gender	0
Age (5-14)	0
Family History	0
Problem Solving	0
Visual Learning Preference	0
Response to Guidance	0
Task Independence	0
Object Identification	0
Error Correction	0
Cognitive Level	0

dtype: int64

Handling Null Values



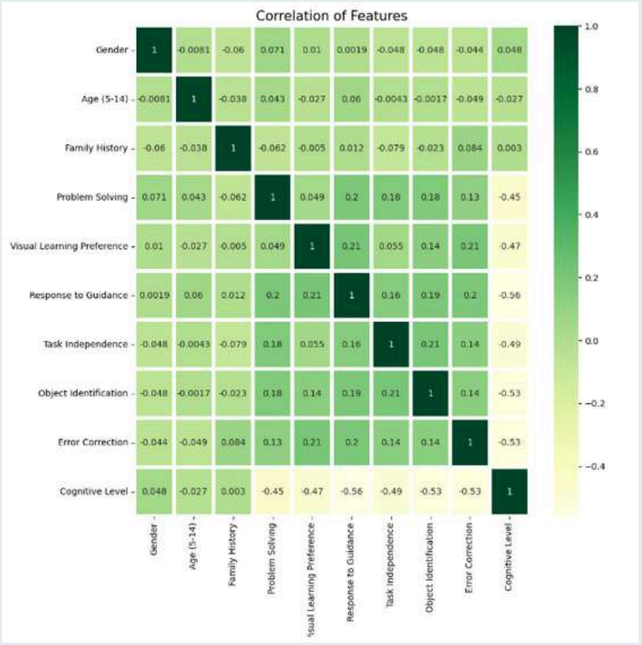
Outlier Boxplot

```
#encode categorical variables

mappings = {
    "Gender": {"Male": 0, "Female": 1},
    "Family History": {"Yes": 1, "No": 0},
    "Cognitive Level": {"Mild": 0, "Moderate": 1, "Severe": 2}
}

# Apply mappings to the DataFrame
for column, mapping in mappings.items():
    if column in df.columns:  # Check if the column exists in your DataFrame
        df[column] = df[column].map(mapping)
```

Encoding Categorical Variables

```
#Sorting correlation values in descending order by FDcreated attribute
df.corr()["Cognitive Level"].sort_values(ascending=False)
```

	Cognitive Level
Cognitive Level	1.000000
Gender	0.047603
Family History	0.002988
Age (5-14)	-0.027358
Problem Solving	-0.453286
Visual Learning Preference	-0.466328
Task Independence	-0.492373
Error Correction	-0.529899
Object Identification	-0.530224
Response to Guidance	-0.559158



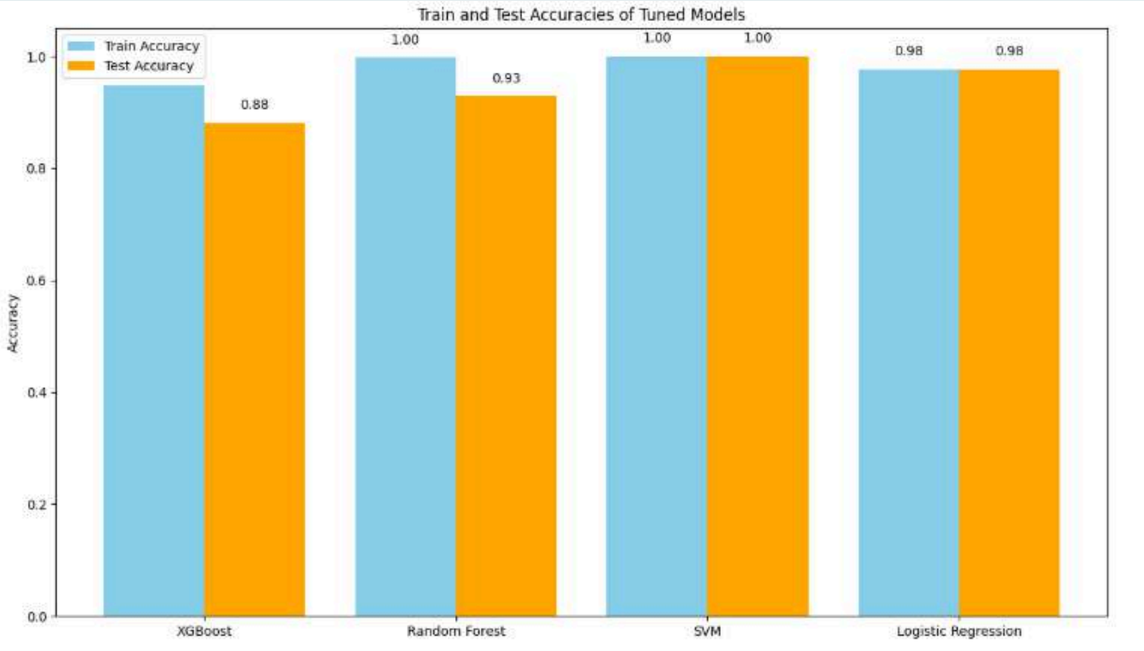
Correlation Matrix

Model Building

- Classification models used:
  - **Random Forest Classifier** (Robust to noisy data and missing values and reduces overfitting due to averaging multiple trees)
  - **Logistic Regression Model** (Works well when the relationship between features and output is linear and when interpretability is crucial and the dataset is not highly complex)
  - **Support Vector Machine** (Works well for both linear and non-linear classification and robust against overfitting in small to medium-sized datasets)
  - **XGBoost Model** (Efficient for small datasets, and it identifies important features)
- Compare the accuracy, recall, precision and F1-score of each method and select the best model
- Creating a function to analyze to predict the "Cognitive Level" of the child with ASD and provide recommendations according to each class.

Model Building

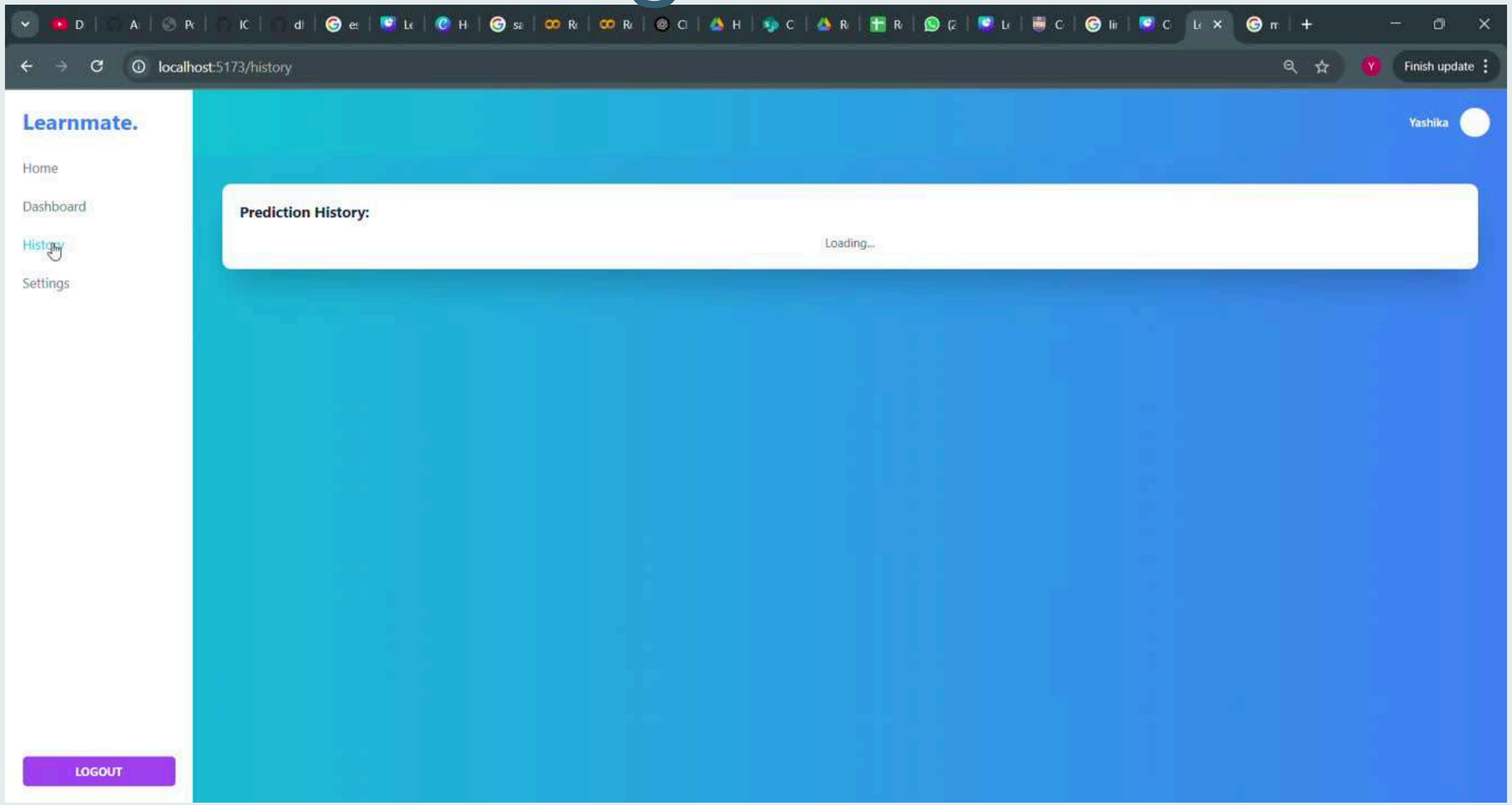
- Logistic Regression has the highest testing accuracy (97.62%) and lowest testing error (0.0238), outperforming both XGBoost and Random Forest. Logistic Regression provides consistent F1-scores across all classes, making it suitable for balanced datasets. Logistic Regression maintains a small gap between training, validation, and test accuracies, indicating strong generalization.



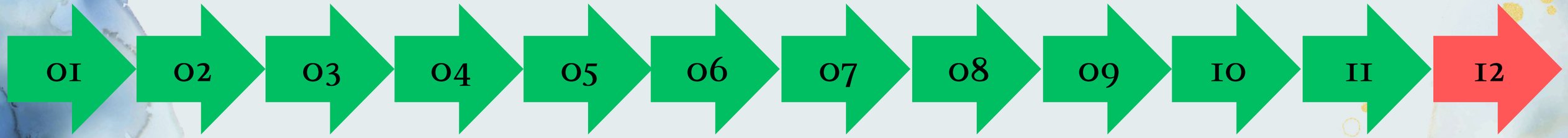
	Model	Train Accuracy	Test Accuracy	Precision	Recall
0	XGBoost	0.948357	0.880952	0.893315	0.880952
1	Random Forest	0.997653	0.928571	0.928685	0.928571
2	SVM	1.000000	1.000000	1.000000	1.000000
3	Logistic Regression	0.976526	0.976190	0.977591	0.976190

Best Model is Logistic Regression

Demo - Cognitive Domain



Project completion plan



01. Background Search
02. Data Gathering
03. Data Analysis & Pre-processing
04. Model Building and Validation
05. Best Model Selection
06. Define recommendations according to level and age.
07. Cognitive Level Prediction
08. Customized Recommendations
09. Progress Tracking
10. Building a web app & integrating the models
11. Testing
12. Releasing to the required institutes

Completed (90%)

To do (10%)

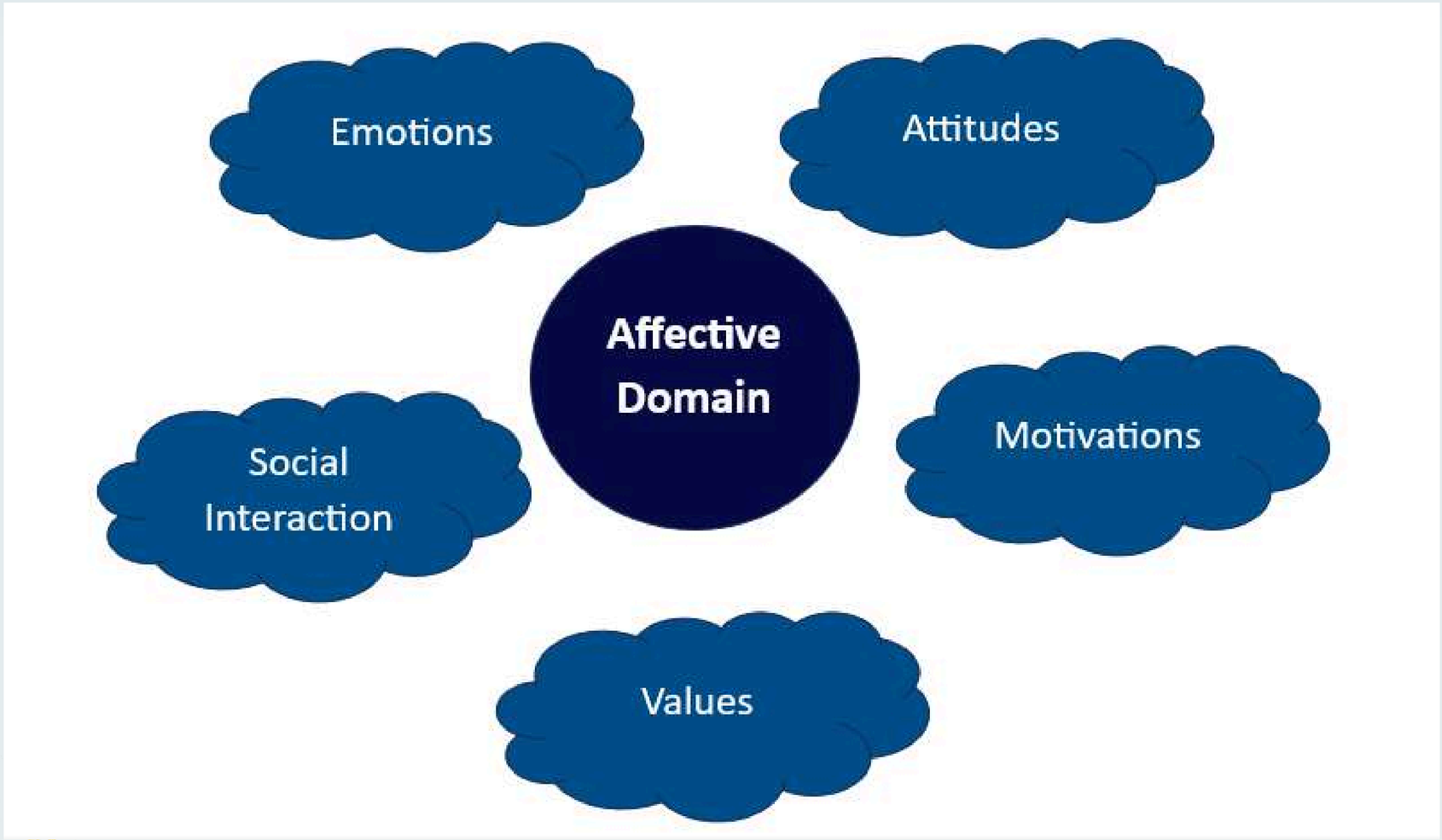
Affective Domain

Ruhunage R.S.D.P

Introduction

- Understanding and managing emotions helps in the construction and maintenance of engaging relations.
- Important for emotional well-being, social life, and adaptive behavior in everyday situations.
- Encourages empathy, regulation of feelings, and belonging to a group, boosting self-confidence.
- Supports the development of positive attitudes and emotional resilience to surmount social challenges.

Affective Domain Skills



Background

- Target Problem: Peers and parents have very limited information concerning the emotional needs and social difficulties of children with autism. This leads to insufficient support in providing them with emotional and social skills.

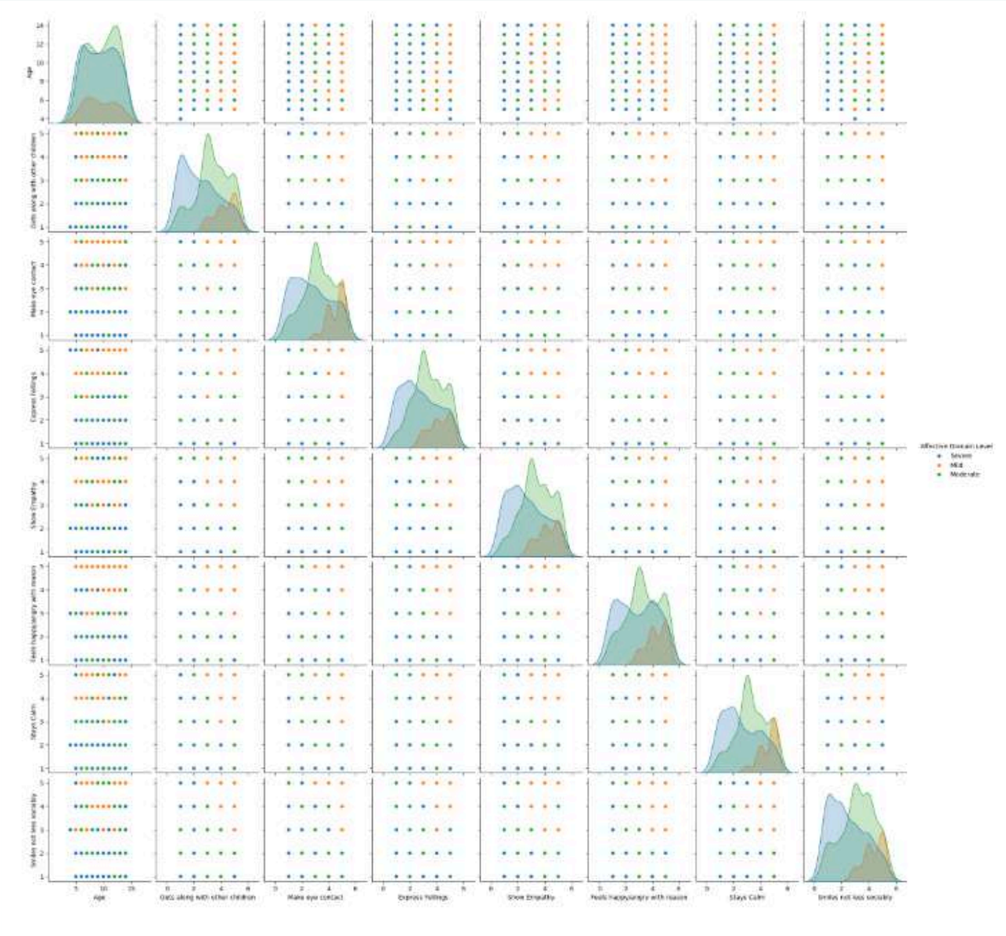
- Proposed Solution: Employ classification to divided autistic children based on their patterns of emotional and social behavior, and provide customized and emotion-based and social-based activities developed in collaboration with parents, teachers, peers and caregivers, to enhance their emotional and social skills.

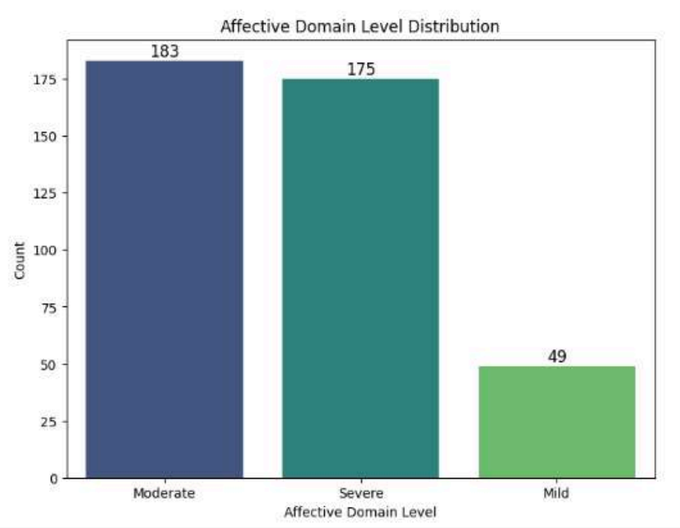
Dataset

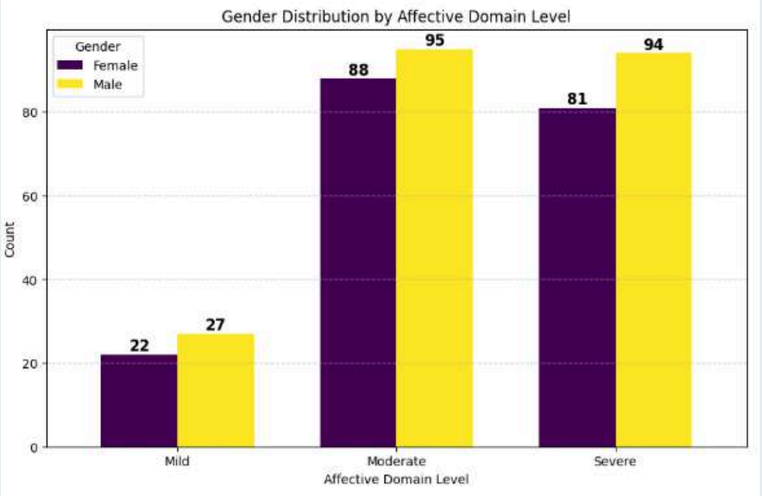
- Features:
  - Demographic: Gender, Age, Family history (If a family member has ASD)
  - Other features:
    1. Gets along with other children
    2. Make eye contact
    3. Express Feelings appropriately
    4. Show Empathy
    5. Feels happy/angry with reason
    6. Stays Calm
    7. Smiles not less socially
- Target variable: Affective Domain Level-› mild, moderate & severe
- Data volume:
  - Total records: 407
  - Number of features(columns): 11 features

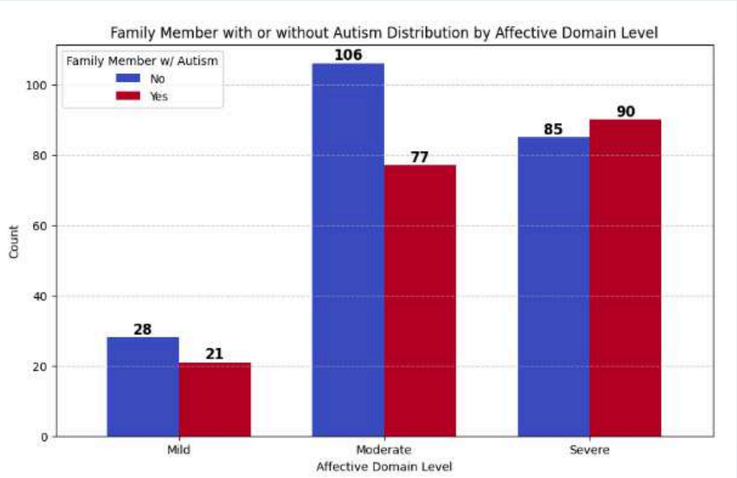
	A	B	C	D	E	F	G	H	I	J	K
1	Gender	Age	Family Member w/ Autism	Gets along with other children	Make eye contact	Express Fellings	Show Empathy	Feels happy/angry with reason	Stays Calm	Smiles not less sociably	Affective Domain Level
2	Female	14	Yes	3	3	4	2	3	2	2	Severe
3	Male	5	No	2	3	3	2	4	2	4	Severe
4	Female	12	No	5	4	5	5	4	3	5	Mild
5	Female	11	Yes	4	5	5	4	3		4	Moderate
6	Female	8	No	3	3	3	3	5	3	3	Moderate
7	Female	10	No	4	5	5	3	3	2	1	Moderate
8	Male	5	No	3	3	3	3	3	3	3	Moderate
9	Male	6	No	4	4	3	4	3	3	5	Moderate
10	Male	6	No	2	2	2	1	2	1	4	Severe

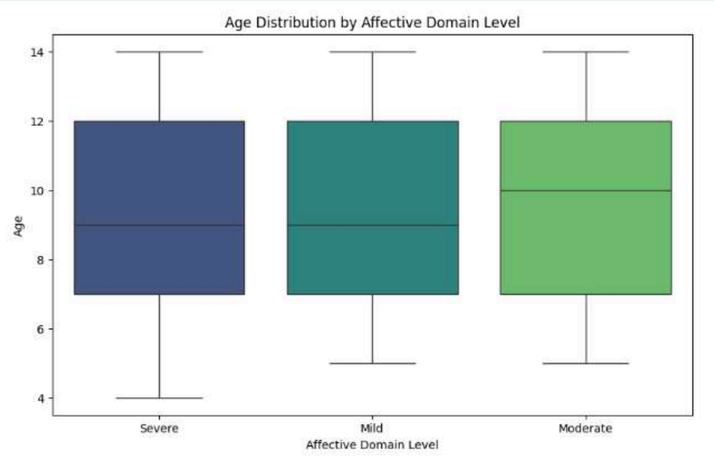
Exploratory Data Analysis











Data Preprocessing

```
categorical_cols = ['Gender', 'Family Member w/ Autism']

#one-hot encoded columns
encoded_features = pd.get_dummies(datasetAD[categorical_cols], columns=categorical_cols, prefix=['Gender', 'FamilyHistory'], drop_first=True)
datasetAD = datasetAD.drop(columns=categorical_cols)
datasetAD = pd.concat([datasetAD, encoded_features], axis=1)

#converted to integers
bool_columns = datasetAD.select_dtypes(include=['bool']).columns
datasetAD[bool_columns] = datasetAD[bool_columns].astype(int)
```

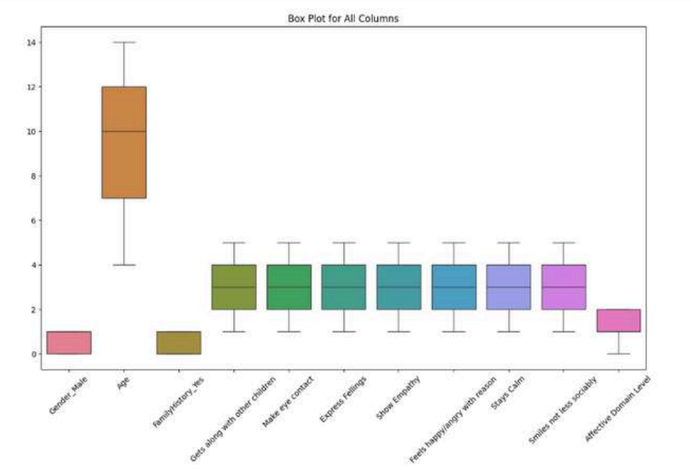
```
[ ]  #encode 'Affective Domain Level' attribute

     ADLevel_mapping = {'Severe': 2, 'Moderate': 1, 'Mild':0}

     datasetAD['Affective Domain Level'] = datasetAD['Affective Domain Level'].map(ADLevel_mapping)
     print(datasetAD["Affective Domain Level"].unique())

     [2 0 1]
```

Encoding Categorical Variables

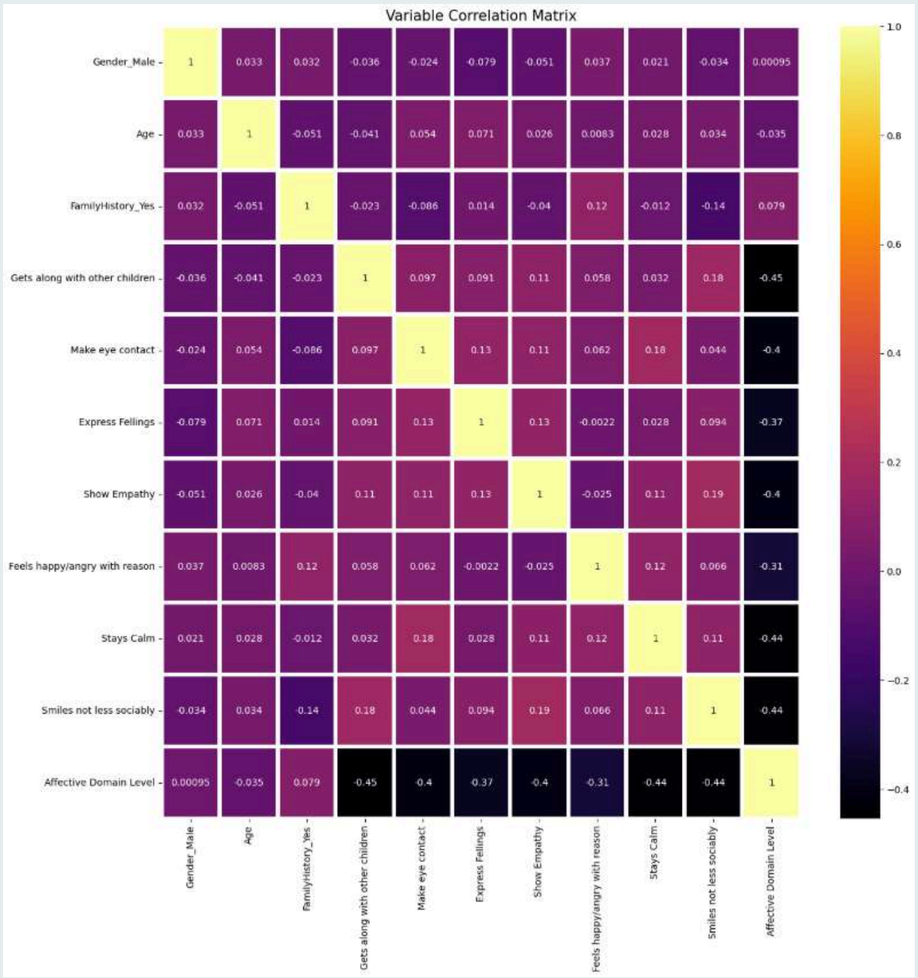
```
#check null values
datasetAD.isnull().sum().sort_values(ascending=False)
```

	0
Show Empathy	2
Stays Calm	2
Make eye contact	1
Gender	0
Age	0
Family Member w/ Autism	0
Gets along with other children	0
Express Fellings	0
Feels happy/angry with reason	0
Smiles not less sociably	0
Affective Domain Level	0

Handling Null Values



Outlier Boxplot



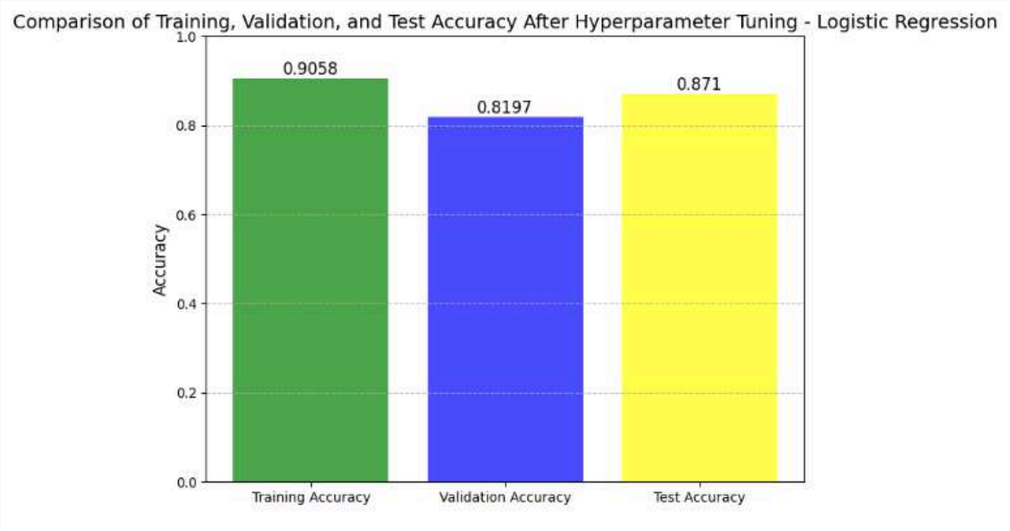
Correlation Matrix

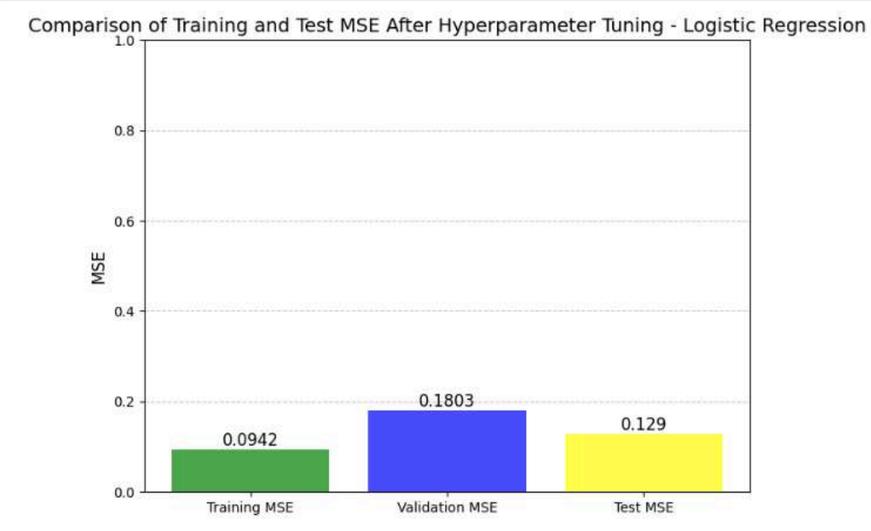
Model Building

- Training set assigned 70%, validation set assigned 15%, and remain data assigned testing set.
- Classification models developed:
  - Random Forest - Handles both numerical and categorical data, reduces overfitting, and identifies essential features.
  - K-Nearest Neighbor (KNN) - Simple, effective for small datasets, and adapts well to non-linear decision boundaries.
  - Naive Bayes - Efficient for small and independent datasets.
  - Logistic Regression - Works well for small datasets, and is effective for linear relationships.
  - Neural Network - Captures complex patterns, generalizes with regularization, and adapts to structured data.
- Based on the results from these methods, compare accuracy, precision, recall, and F1-score to decide the best model for the given dataset.
- Creating a function to analyze to predict the “Affective Level” of the child with ASD and provide recommendations according to each class.

Model Building

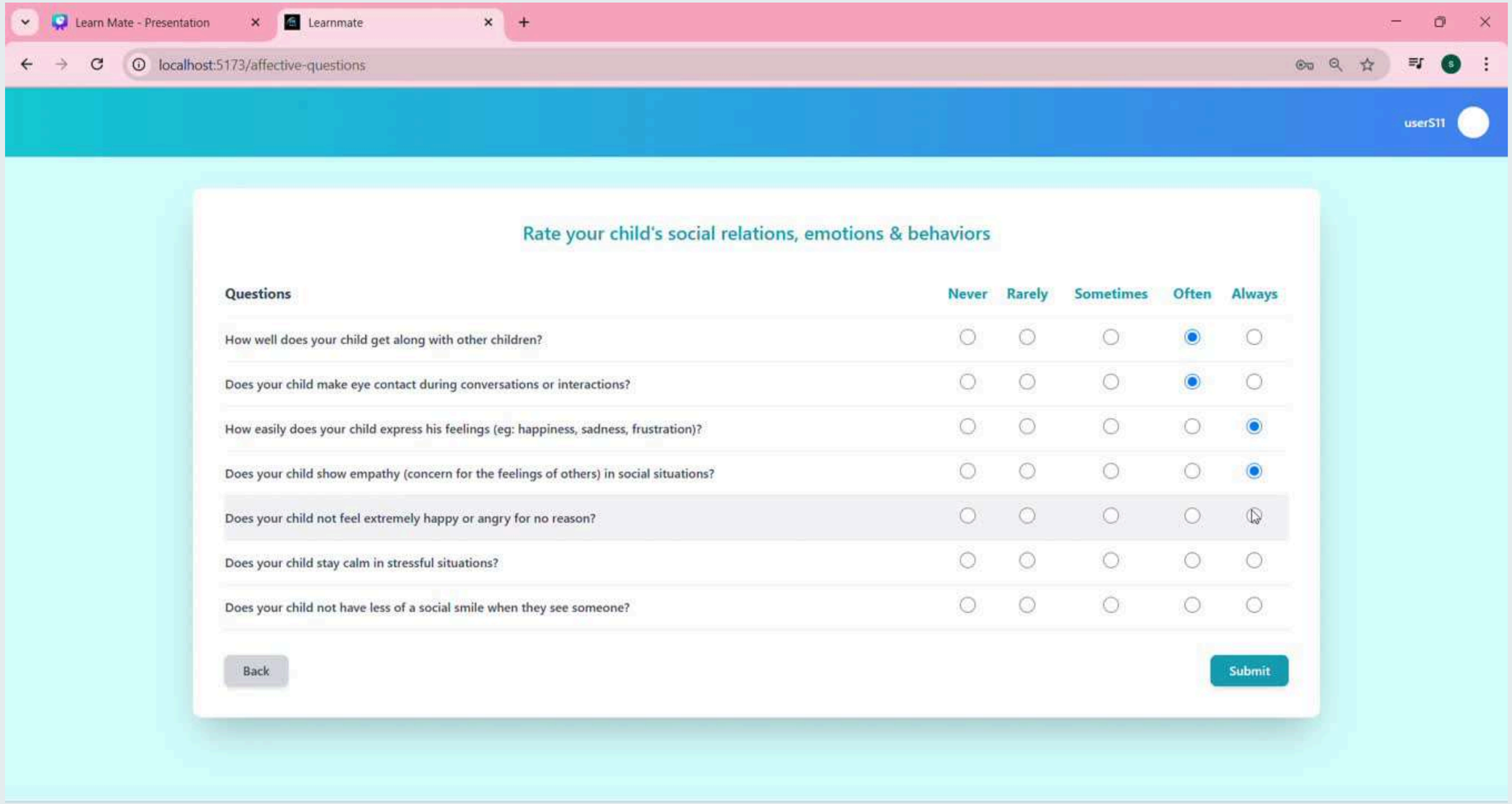
	A	B	C	D
1	**Model**	**Validation Accuracy**	**Test Accuracy**	**Reason**
2	Random Forest	85%	72%	Performance improved post-hypertuning, but recall is still lower on the test set.
3	KNN	63%	68%	Struggles with recall on the validation and test sets.
4	Naive Bayes	92%	79%	Consistently high performance on training and validation sets, but test set recall is lower.
5	Logistic Regression	85%	91%	**BEST MODEL [Consistently achieved the highest accuracy and balanced precision, recall, and f1-scores.]**
6	Neural Network	88%	81%	Struggles with recall, though performs well in training set accuracy.
7				



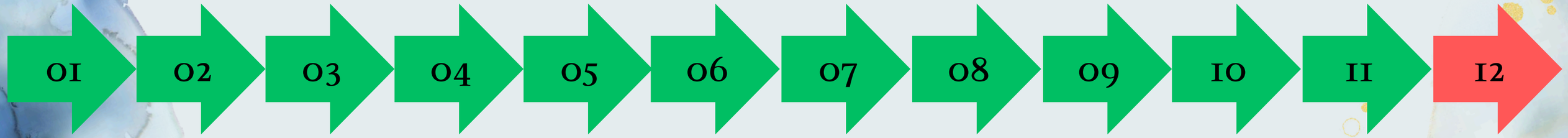


Best Model is Logistic Regression

Demo - Affective Domain



Project completion plan



01. Background Search
02. Data Gathering
03. Data Analysis & Pre-processing
04. Model Building and Validation
05. Best Model Selection
06. Define recommendations according to level and age.
07. Affective Domain Level Prediction
08. Customized Recommendations
09. Progress Tracking
10. Building a web app & integrating the models
11. Testing
12. Releasing to the required institutes

Completed (90%)

To do (10%)

Psychomotor Domain

Pehesarani W.K.A

Introduction

Psychomotor skills involve the coordinated function of the brain and muscles, enabling movements and physical activities.

Gross Motor Skills

Walking
Running
Jumping
Swimming
Climbing
Throwing and Catching

Fine Motor Skills

Writing
Drawing
Coloring
Buttoning Clothes
Typing

Background

- Target Problem: Autistic children often experience difficulties in enhancing their psychomotor skills, such as fine and gross motor abilities. Existing methods lack the ability to assess psychomotor levels and provide recommendations suitable for their developmental stage.

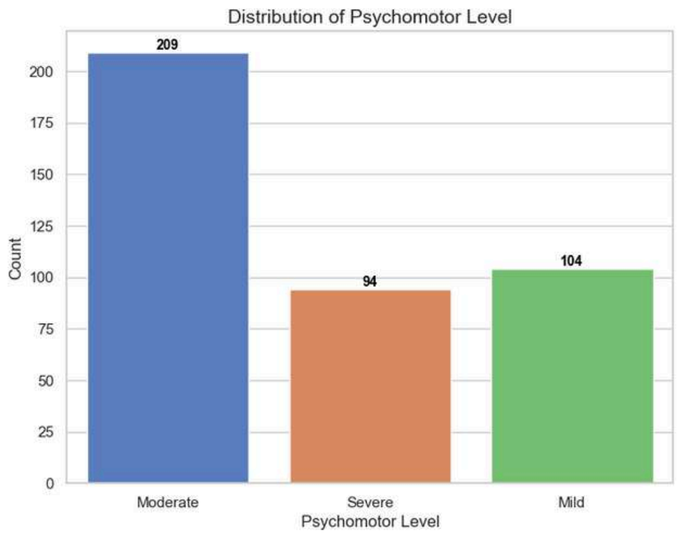
- Proposed Solution: The proposed system predicts the psychomotor level using ensemble methods and generates customized recommendations based on the child's level and age. With the support of parents, teachers, or peers, children can complete these tasks, and their progress is tracked to refine future recommendations.

Dataset

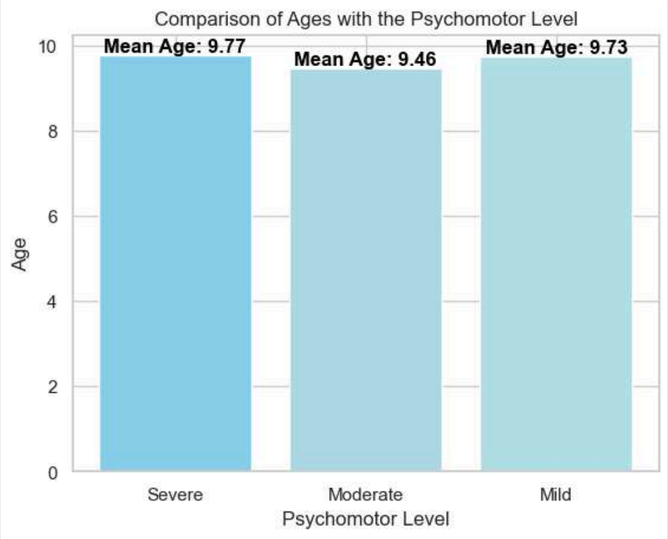
- Features:
  - Demographic Features: Gender, Age, Family history of ASD
  - Psychomotor Skills: Balance and stability, Grip strength, Large mussels coordination, Hand-eye coordination, Object manipulation, Utensil use, and clothing independence.
- Target variable:
  - The target variable is the psychomotor skill level, categorized as Mild, Moderate, or Severe
- Data volume:
  - Total records: 407
  - Number of features/columns: 11 features

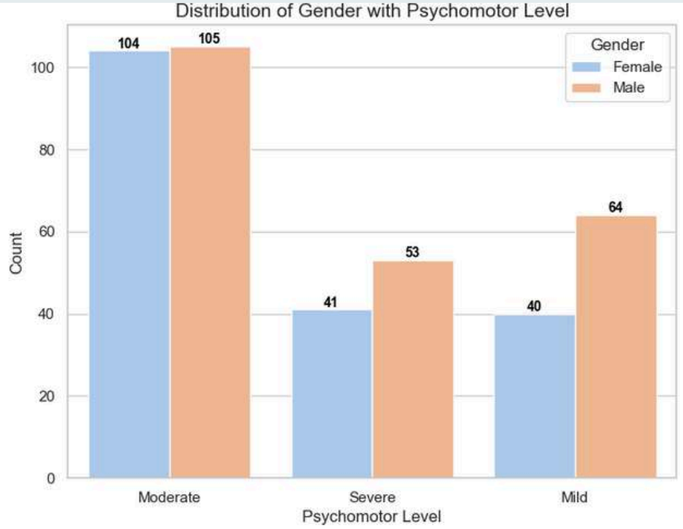
	A	B	C	D	E	F	G	H	I	J	K
1	**Gender**	**Age**	**Family_ASD_History**	**Balance_and_Stabilit**	**Grip_Strength**	**Coordination**	**Hand_Eye_Coordinati**	**Object_ Manipulation**	**Independent_Use_Ut**	**Button_Zip_Clothes**	**Psychomotor_Level**
2	Female	8	No	Rarely	Maybe	Often	Rarely	Often	Maybe	Never Have	Moderate
3	Female	14	Yes	Always	Often	Rarely	Maybe	Maybe	Maybe	Never Have	Moderate
4	Female	10	No	Often	Maybe	Never Have	Rarely	Never Have	Never Have	Maybe	Severe
5	Male	14	No	Always	Never Have	Maybe	Often	Maybe	Often	Never Have	Moderate
6	Male	12	No	Always	Never Have	Rarely	Never Have	Rarely	Never Have	Never Have	Severe
7	Male	14	Yes	Maybe	Never Have	Often	Maybe	Often	Maybe	Rarely	Moderate
8	Male	13	No	Always	Rarely	Often	Often	Always	Maybe	Often	Mild
9	Male	12	Yes	Never Have	Maybe	Often	Never Have	Rarely	Maybe	Always	Moderate
10	Male	7	Yes	Often	Often	Often	Often	Maybe	Often	Always	Mild
11	Female	12	No	Always	Rarely	Always	Rarely	Never Have	Maybe	Often	Moderate
12	Male	6	Yes	Rarely	Always	Rarely	Never Have	Maybe	Rarely	Often	Moderate
13	Male	6	No	Always	Rarely	Maybe	Never Have	Rarely	Rarely	Rarely	Moderate
14	Female	13	Yes	Maybe	Rarely	Always	Rarely	Rarely	Never Have	Often	Moderate
15	Male	8	Yes	Always	Often	Often	Rarely	Always	Often	Often	Mild

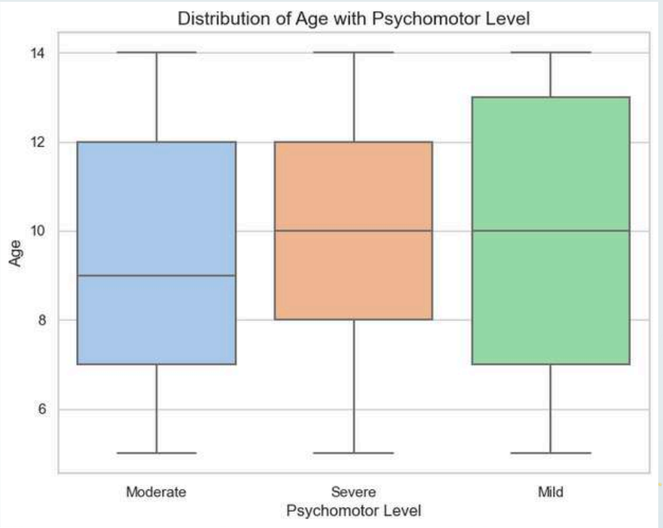
Exploratory Data Analysis



Distribution of the dataset according to different attributes







Data Preprocessing

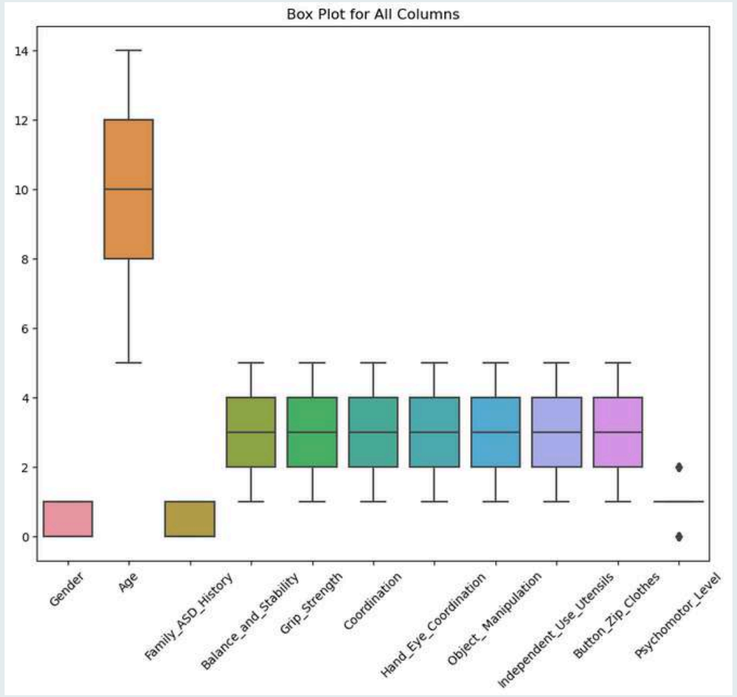
```
df.isna().sum()
```

```
Gender                          0
Age                             0
Family_ASD_History              0
Balance_and_Stability           0
Grip_Strength                   0
Coordination                    0
Hand_Eye_Coordination           0
Object_Manipulation             0
Independent_Use_Utensils        0
Button_Zip_Clothes              0
Psychomotor_Level               0
dtype: int64
```

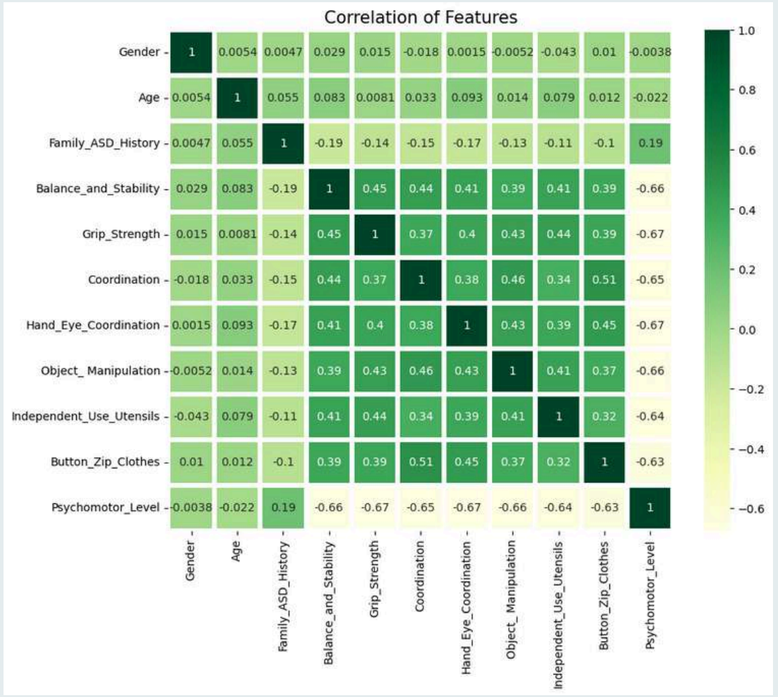
Checking missing values

Mapped categorical variables

```
mappings = {
    "Gender": {"Male": 1, "Female": 0},
    "Family_ASD_History": {"Yes": 1, "No": 0},
    "Balance_and_Stability": {"Never Have": 1, "Rarely": 2, "Maybe": 3, "Often": 4, "Always": 5},
    "Grip_Strength": {"Never Have": 1, "Rarely": 2, "Maybe": 3, "Often": 4, "Always": 5},
    "Coordination": {"Never Have": 1, "Rarely": 2, "Maybe": 3, "Often": 4, "Always": 5},
    "Hand_Eye_Coordination": {"Never Have": 1, "Rarely": 2, "Maybe": 3, "Often": 4, "Always": 5},
    "Object_ Manipulation": {"Never Have": 1, "Rarely": 2, "Maybe": 3, "Often": 4, "Always": 5},
    "Independent_Use_Utensils": {"Never Have": 1, "Rarely": 2, "Maybe": 3, "Often": 4, "Always": 5},
    "Button_Zip_Clothes": {"Never Have": 1, "Rarely": 2, "Maybe": 3, "Often": 4, "Always": 5},
    "Psychomotor_Level": {"Mild": 0, "Moderate": 1, "Severe": 2}
}

# Map features to numerical values
for column, mapping in mappings.items():
    if column in df.columns:
        df[column] = df[column].map(mapping)

df
```



Outliers Boxplot



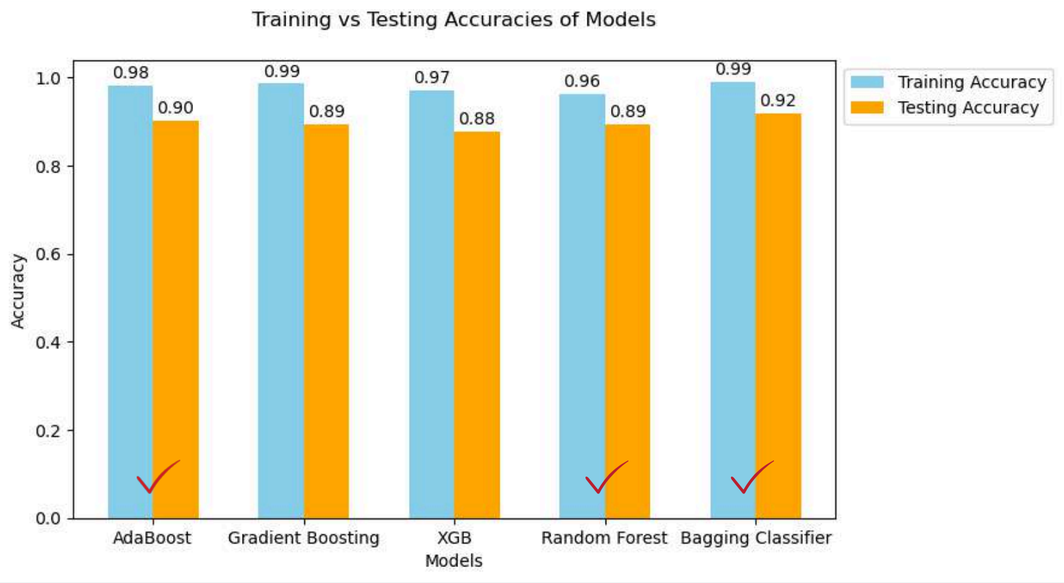
Correlation matrix

Model Building

1. Algorithm Selection
    - Used ensemble methods for better accuracy and reliability.
    - Bagging Algorithms: Bagging classifier, Random Forest
    - Boosting Algorithms: AdaBoost, GBM, XGBM

2. Model Training and Validation
    - Split data into training and testing sets.
    - Optimized model parameters using cross-validation to prevent overfitting.

3. Evaluation Metrics
    - Evaluated models performance using metrics like accuracy, precision, recall, and F1-score.
    - Chose the best-performing models based on predictive accuracy.

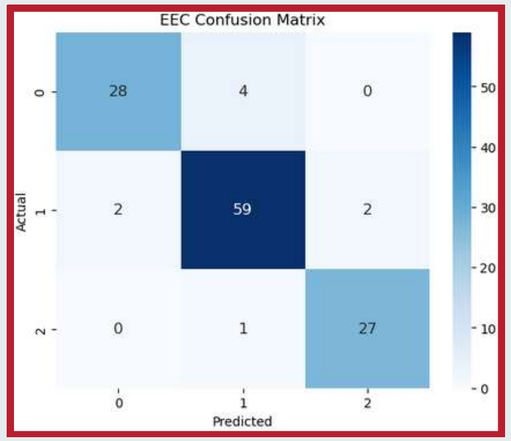


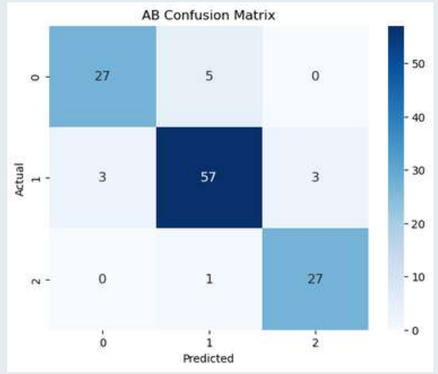
Model Building

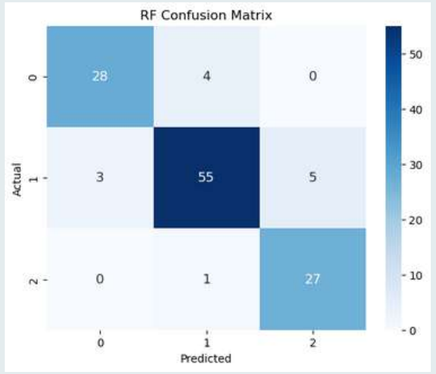
4. Model Aggregation
   - Used a Voting Classifier to combine the predictions of the best-performing models (Bagging and Boosting algorithms) to achieve improved accuracy and robustness.
   - The ensemble approach aggregated the strengths of individual models, resulting in more reliable and consistent predictions.

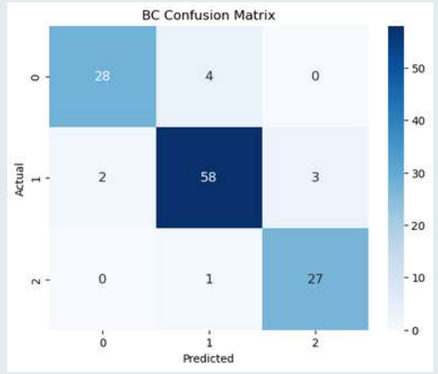
```
# Create a VotingClassifier with the best hyperparameters for voting='hard'
ensemble_classifier = VotingClassifier(
    estimators=[('rf', best_rf_model), ('ab', optimized_ada_model), ('bag', optimized_bagging_model)],
    voting='hard'  # Uses majority rule voting, where the class with the most votes is selected
)
```

```
Training Accuracy: 0.986301
Test Accuracy: 0.926829
```

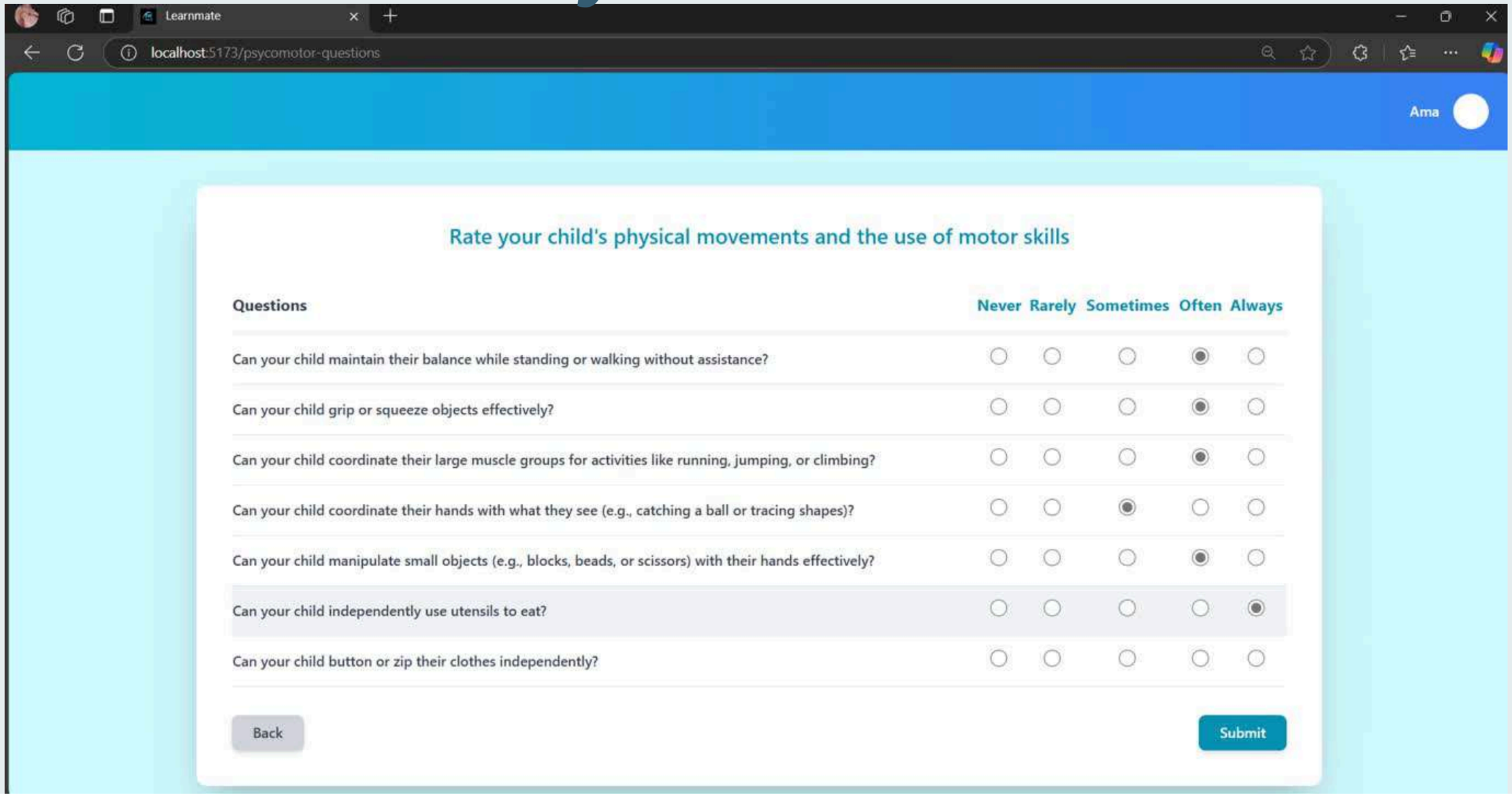




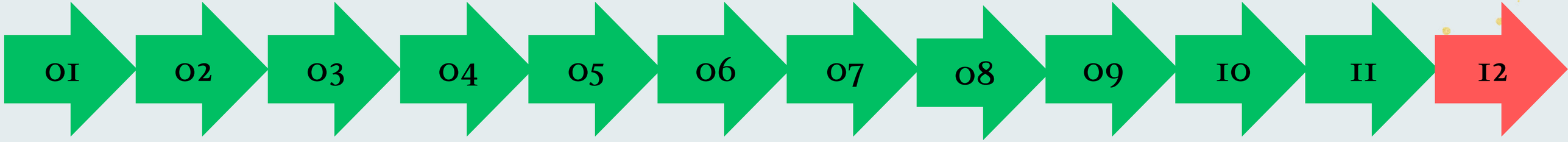




Demo - Psychomotor Domain



Project Completion



01. Background Search
02. Data Gathering
03. Data Analysis & Pre-processing
04. Model Building
5. Best Model Selection
06. Define recommendations according to level and age.
07. Psychomotor Domain Level Prediction
08. Customized Recommendations
09. Progress Tracking
10. Building a web app & integrating the models
11. Testing
12. Deployment and releasing to the required institutes

Completed 90%

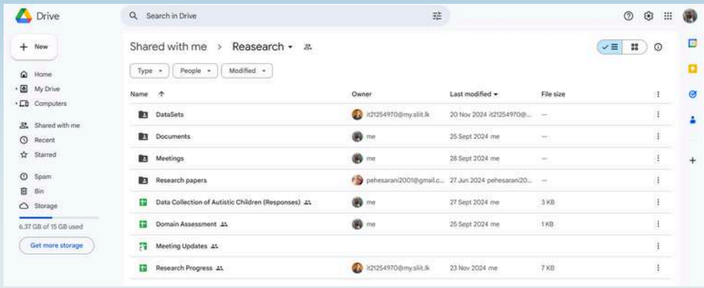
To do 10%

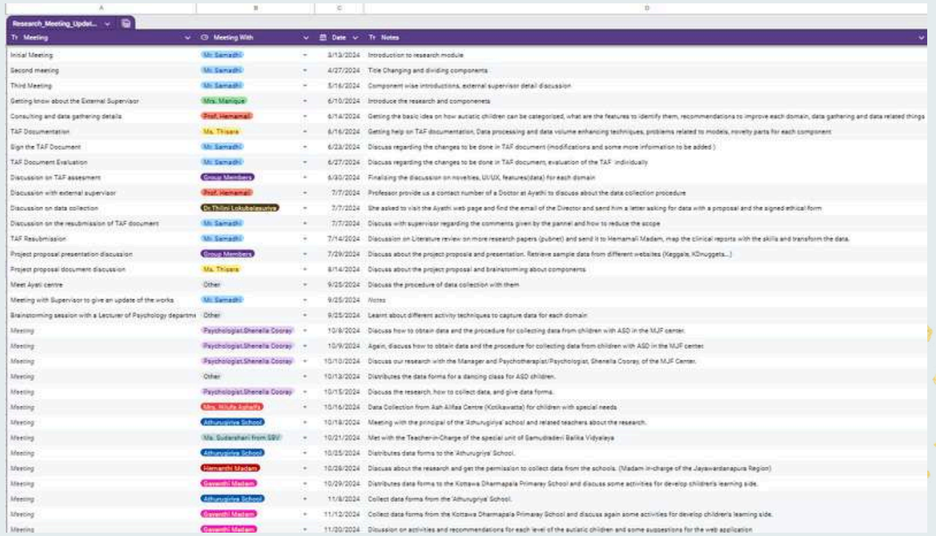
Standards and Knowledge Utilization

- Key pillars of data science and IT utilized in the implementation:
  - Data Management
  - Usage of Classification models and Ensemble methods
  - Progress Tracking Functionality

- Technologies: Python, Flask, React, MongoDB and GitHub.

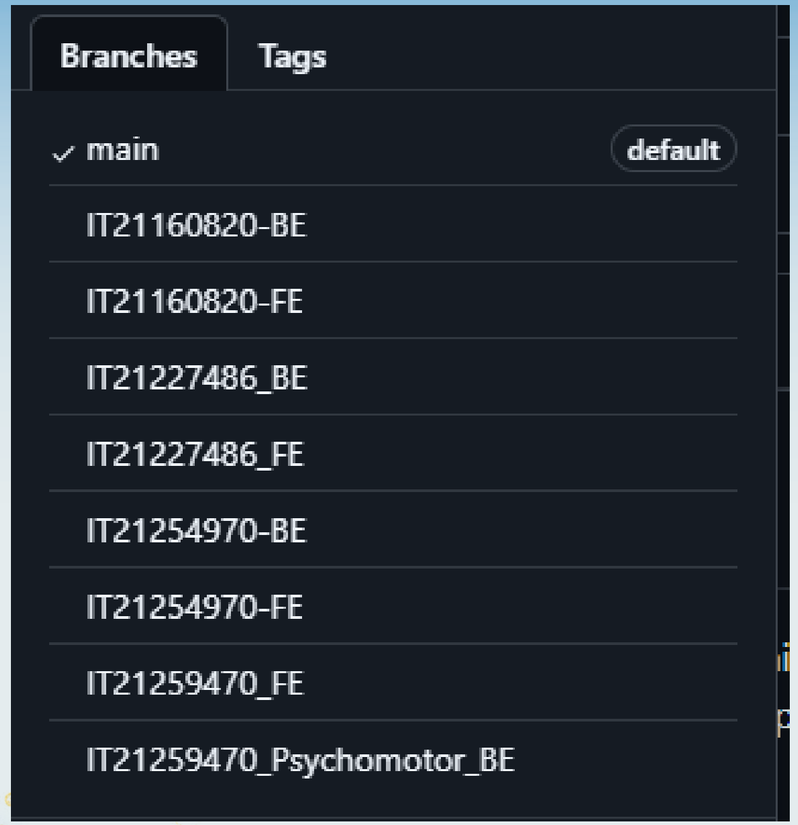
- Best Practices:
  - Followed coding standards (indentations, commenting, use of functions)
  - Efficient version control (GitHub)
  - Managing a log to track progress
  - Using cloud based platform to store related documents, images and references.



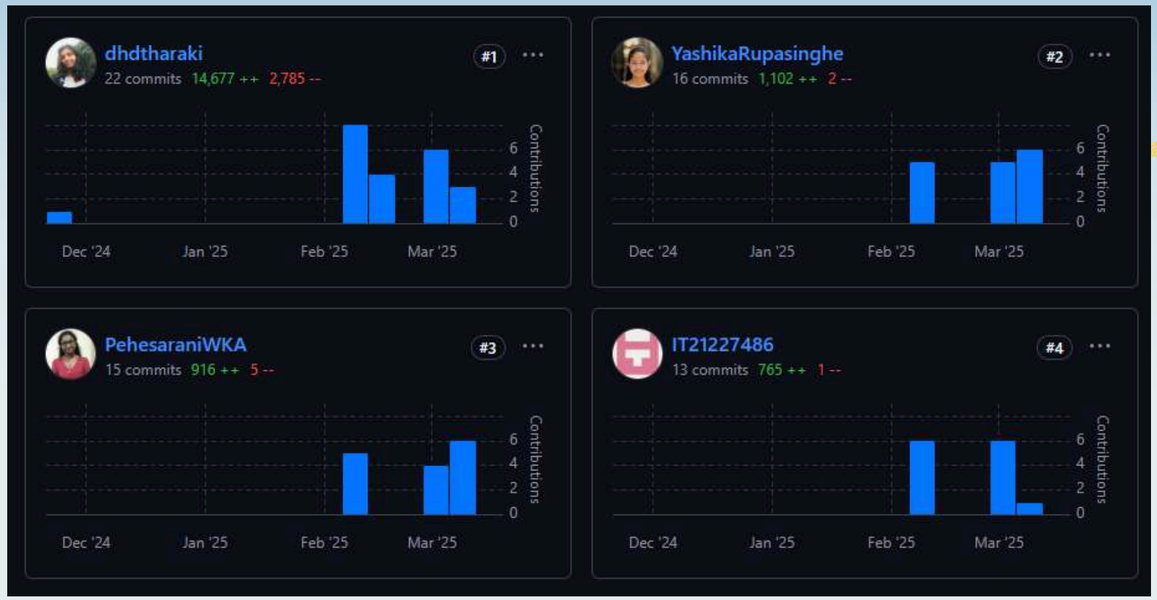


- For risk mitigation:
  - Expert validation on recommendations.
  - Feedback from educators and parents.
  - Weekly reviews and discussions with the team for consistent progress.

Version Controlling (GitHub)



Branches



Commits

Folder Structure

Commercialization

- Release the pilot version to data-providing centers.
- Collect feedback from these centers.
- Develop the application further with advanced features based on feedback.
- Release the improved application to the public.

System Design and Implementation

Web Application

- Tailored for parents, peers, and educators to access recommendations and progress reports.

Design Approach

- Minimal and user-friendly design for easy accessibility.

Development Methodology

- React-based web application.
- MongoDB database used for data storing.
- Hosting on Microsoft Azure platform for reliable deployment and scalability.

Integration

- Backend support for machine learning based techniques and progress-tracking functionality.

Q & A

THANK YOU!