

Data Collection And Preprocessing Phase

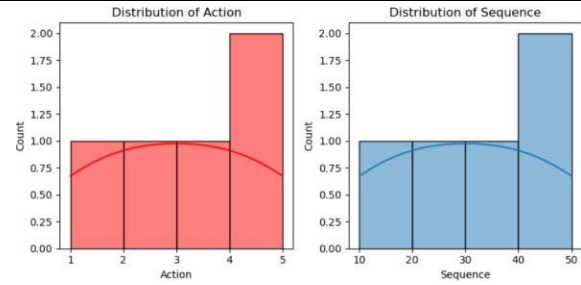
Date	06JULY2024
TeamID	739909
ProjectName	Unlocking Silent Signals: Decoding Body Language With Mediapipe
Maximum Marks	6 Marks

Data Exploration and Preprocessing Report:

Dataset variables will undergo statistical analysis to identify patterns and outliers. Python will be used for preprocessing tasks such as normalization and feature engineering. Data cleaning will address missing values and outliers to ensure data quality for subsequent analysis and modeling, forming a robust foundation for insights and predictions.

Section	Description																		
Data Overview	<div><pre>df.describe()</pre><table><thead><tr><th></th><th>Sequence</th></tr></thead><tbody><tr><td>count</td><td>3600.000000</td></tr><tr><td>mean</td><td>14.500000</td></tr><tr><td>std</td><td>8.656644</td></tr><tr><td>min</td><td>0.000000</td></tr><tr><td>25%</td><td>7.000000</td></tr><tr><td>50%</td><td>14.500000</td></tr><tr><td>75%</td><td>22.000000</td></tr><tr><td>max</td><td>29.000000</td></tr></tbody></table></div>		Sequence	count	3600.000000	mean	14.500000	std	8.656644	min	0.000000	25%	7.000000	50%	14.500000	75%	22.000000	max	29.000000
	Sequence																		
count	3600.000000																		
mean	14.500000																		
std	8.656644																		
min	0.000000																		
25%	7.000000																		
50%	14.500000																		
75%	22.000000																		
max	29.000000																		

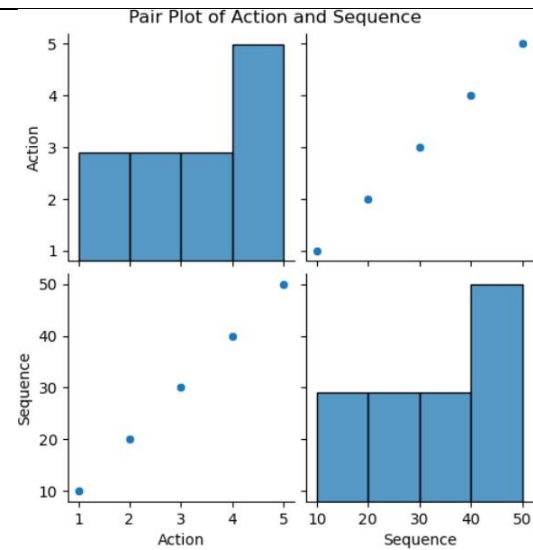
Univariate Analysis



Bivariate Analysis



Multivariate Analysis



Outliers and Anomalies

-

DataPreprocessing Code Screenshots

LoadingData	<pre>df = pd.read_csv("C:/Users/lenovo/Desktop/Body_Language_Decoder/dataset/body_language_data.csv") # Print the column names to verify the exact names print(df.columns) Index(['Action', 'Sequence', 'Data'], dtype='object') # Print the first few rows to inspect the data print(df.head())</pre> <table><thead><tr><th></th><th>Action</th><th>Sequence</th><th>Data</th></tr></thead><tbody><tr><td>0</td><td>happy</td><td>0</td><td>[0.6459652781486511 0.6242932081222534 -0.7729...</td></tr><tr><td>1</td><td>happy</td><td>0</td><td>[0.6458597779273987 0.6174453496932983 -1.0967...</td></tr><tr><td>2</td><td>happy</td><td>0</td><td>[0.7056352496147156 0.5952770709991455 -0.9480...</td></tr><tr><td>3</td><td>happy</td><td>0</td><td>[0.7088898420333862 0.5936240553855896 -0.7997...</td></tr><tr><td>4</td><td>happy</td><td>0</td><td>[0.7049200743094024 0.5924623141208797 -0.8366...</td></tr></tbody></table>		Action	Sequence	Data	0	happy	0	[0.6459652781486511 0.6242932081222534 -0.7729...	1	happy	0	[0.6458597779273987 0.6174453496932983 -1.0967...	2	happy	0	[0.7056352496147156 0.5952770709991455 -0.9480...	3	happy	0	[0.7088898420333862 0.5936240553855896 -0.7997...	4	happy	0	[0.7049200743094024 0.5924623141208797 -0.8366...
	Action	Sequence	Data																						
0	happy	0	[0.6459652781486511 0.6242932081222534 -0.7729...																						
1	happy	0	[0.6458597779273987 0.6174453496932983 -1.0967...																						
2	happy	0	[0.7056352496147156 0.5952770709991455 -0.9480...																						
3	happy	0	[0.7088898420333862 0.5936240553855896 -0.7997...																						
4	happy	0	[0.7049200743094024 0.5924623141208797 -0.8366...																						
Handling Missing Data	<pre># Handling Missing Values print(df.isnull().sum()) Action 0 Sequence 0 Data 0 dtype: int64</pre>																								
Data Transformation	-																								
Feature Engineering	Attached the codes in final submission.																								
Save Processed Data	<pre>from sklearn.ensemble import RandomForestClassifier import pickle # Assuming X_resampled and y_resampled are obtained from RandomOverSampler # Train your model model = RandomForestClassifier(random_state=42) model.fit(X_resampled, y_resampled) # Save the model model_path = 'C:/Users/lenovo/Desktop/Body_Language_Decoder/model/body_language.pkl' with open(model_path, 'wb') as f: pickle.dump(model, f)</pre>																								