Predicting Personality Traits and Values from Facebook Status Updates

Khondokar Riaz Mahmud
Deptartment of Computer Science and Engineering
United International University, Dhaka, Bangladesh
kmahmud2310008@mscse.uiu.ac.bd

Md. Mahabubar Rahman
Department of Computer Science and Engineering
United International University, Dhaka, Bangladesh
mrahman2310007@mscse.uiu.ac.bd

Sanjida Akter
Department of Computer Science and Engineering
United International University, Dhaka, Bangladesh
sakter2230056@mscse.uiu.ac.bd

Md. Saddam Hossain
Department of Computer Science and Engineering
United International University, Dhaka, Bangladesh
saddam@cse.uiu.ac.bd

***Abstract*—The purpose of this study is to see whether the Big Five personality traits and Schwartz values of individuals can be predicted using the Linguistic Inquiry and Word Count (LIWC) variables that were derived from Facebook status updates. To investigate the connections between LIWC factors and the target variables, Pearson correlation analysis is utilized. Principal Component Analysis (PCA) is also used to decompose the dimensionality of the LIWC variables and discover important trends. The results provide insights into the possible predictive capacities of LIWC in identifying people's psychological qualities based on their online expressions. They also shed light on the correlations between language patterns and personality traits/values.**

***Index Terms*—LIWC: Big Five personality traits ;Schwartz values**

I. Introduction

This study investigates the association between Big Five personality traits and Schwartz values , and Linguistic Inquiry and Word Count (LIWC) variables derived from Facebook status updates. By examining written content, LIWC offers a thorough understanding of language patterns and psychological processes. The goal of this study is to look at the predictive power of LIWC factors in predicting people's values and personality characteristics.

A diverse sample population was recruited, and their Facebook status updates were collected along with self-reported measures of Big Five personality traits and Schwartz values. LIWC was applied to extract various linguistic features from the text data, including word usage, emotion, social connections, and cognitive processes. The LIWC variables were then analyzed using Pearson correlation analysis to assess the strength and significance of the relationships with the target variables. The results of the correlation analysis revealed the associations between LIWC variables and individuals' Big Five personality traits and Schwartz values. Positive or negative correlations indicated the direction and strength of the relationships. Significant correlations suggested potential predictive power of specific LIWC variables for certain personality traits and values. Additionally, Principal Component Analysis (PCA) was conducted to reduce the dimensionality of the LIWC variables and identify underlying patterns or latent factors. PCA helps to identify the most important features contributing to the variance in the data. By reducing the dimensionality, PCA enables a more concise representation of the LIWC variables while preserving the essential information. The findings from this study contribute to understanding the role of LIWC variables in predicting individuals' personality traits and values based on their Facebook status updates. The results of the Pearson correlation analysis provide insights into the specific LIWC variables that are most strongly associated with the target variables. The application of PCA aids in identifying the most influential LIWC features and reducing the complexity of the data. However, it is important to acknowledge the limitations of this study, including potential biases in self-reported data, the generalizability of the findings to larger populations, and the interpretability of the extracted latent factors. Future research could explore additional statistical techniques and validation methods to further investigate the predictive power of LIWC variables and improve the accuracy of the models. Overall, this study highlights the potential of LIWC variables and PCA in predicting individuals' Big Five personality traits and Schwartz values based on their Facebook status updates, contributing to a better understanding of the relationship between language patterns, personality traits, and values in the digital realm.

II. Dataset

The dataset used in this study consists of three main components: LIWC variables extracted from Facebook status updates, Big Five personality trait scores and Schwartz values obtained from a survey.

LIWC Variables: The LIWC variables are derived from the linguistic analysis of Facebook status updates. LIWC (Linguistic Inquiry and Word Count) is a widely used text analysis tool that examines language patterns and provides insights into psychological processes. The LIWC variables capture various linguistic dimensions such as word usage, emotional expression, social connections, cognitive processes, and more.

Big Five Personality Traits: The Big Five personality traits, also known as the Five-Factor Model (FFM), are a widely accepted framework for describing human personality. The traits include Openness to Experience, Conscientiousness, Extraversion, Agreeableness, and Neuroticism. These traits are assessed through self-reported measures obtained from survey responses.

Schwartz Values: The Schwartz values represent an influential theory of basic human values. The theory identifies ten fundamental values, including Self-Direction, Stimulation, Hedonism, Achievement, Power, Security, Conformity, Tradition, Benevolence, and Universalism. Participants in the study provide self-reported scores reflecting their endorsement of these values.

The dataset allows for exploring the relationships between LIWC variables and individuals' Big Five personality traits and Schwartz values. The aim is to determine whether specific linguistic patterns captured by LIWC can predict or provide insights into individuals' personality traits and values. The dataset provides a valuable resource for investigating the connections between language use, psychological characteristics, and individual differences in the context of social media behavior and self-report measures.

III. Methodology

A. Dataset Preprocessing

To remove outliers using the IQR method and replace them with the feature mean in a dataset containing LIWC, Big Five, and Schwartz values, you can follow these steps:

1.Identifying the subset of columns in your dataset that correspond to the LIWC, Big Five, and Schwartz values.

2.Iterating over each column and apply the IQR outlier detection and replacement process.

3.Calculating the IQR, lower bound, and upper bound for each column.

4.Replacing outliers with the mean value of the respective feature.

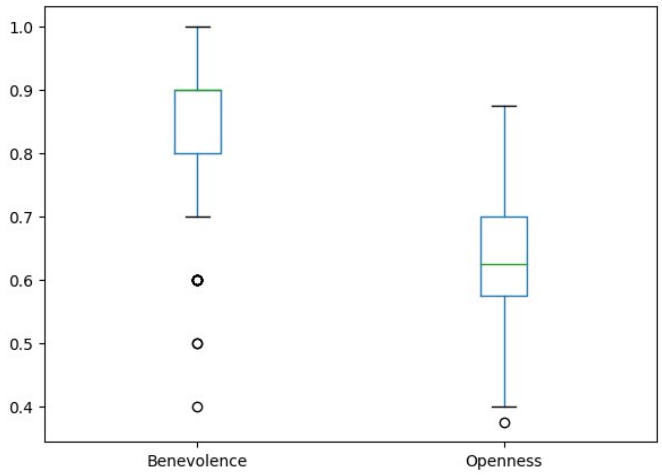

Fig. 1. Before Removing outliers

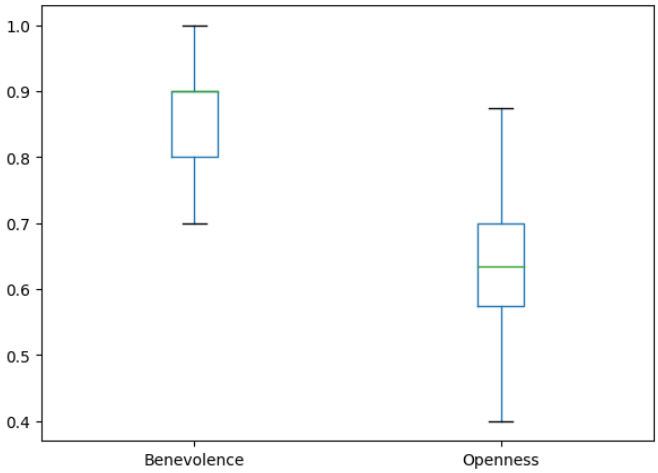

Fig. 2. After Removing outliers

B. Feature selection

Feature selection is an important step in the data analysis process that aims to identify the most relevant features from a dataset for predictive modeling or analysis. Two commonly used methods for feature selection are Pearson correlation and Principal Component Analysis (PCA).

By applying both Pearson correlation and PCA for feature selection, we can benefit from different perspectives. Pearson correlation focuses on the linear relationship between individual features and target variables, while PCA considers the overall variance in the dataset. Both methods help to identify features that are most likely to contribute to the prediction or analysis of Big Five personality traits and Schwartz values.

We used both of them separately on the dataset to observe how our chosen models for linear regression performed as well as for k means clustering.

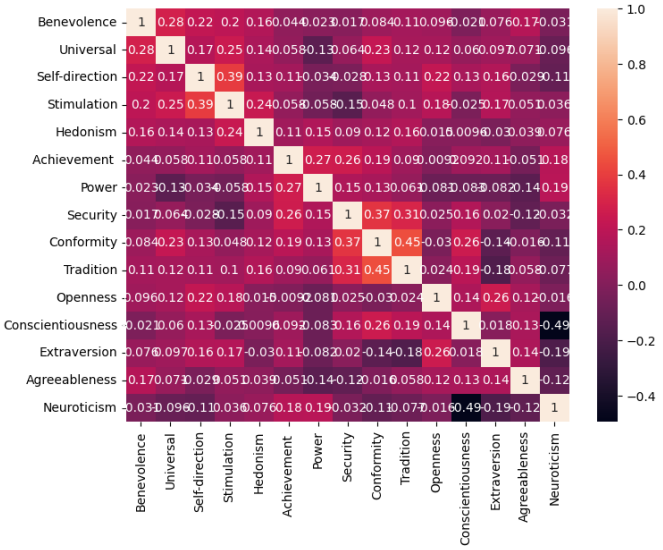

Fig. 3. Pearson Correlation

C. Models Used & Architecture

1) LSTM (Long Short-Term Memory): The LSTM (Long Short-Term Memory) model is a type of recurrent neural network (RNN) that is particularly effective for sequence-based data analysis, including regression problems. It can capture and learn complex temporal dependencies in the input data, making it suitable for tasks such as time series forecasting or regression tasks where the input features have sequential or temporal patterns. For a regression problem, the goal is to predict a continuous target variable. In this case, the LSTM model can be used to learn the relationship between the input features and the target variable and make predictions.

Lstm model

```
#lstm model

X = df_1.iloc[:, :-15].values
y = df_1.iloc[:, -15:].values
# Reshape the input features to fit the LSTM input shape
X_1 = X.reshape(X.shape[0], 1, X.shape[1])

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_1, y, test_size=0.2, random_state=42)

# Define the LSTM model
model = Sequential()
model.add(LSTM(50, input_shape=(X_1.shape[1], X_1.shape[2])))
model.add(Dense(15))
model.compile(loss='mse', optimizer='adam', metrics=['accuracy'])

# Train the model
model.fit(X_train, y_train, epochs=1000, batch_size=5)

# Predict the output variables for the test set
y_pred = model.predict(X_test)

# Calculate the mean squared error between the predicted and actual outputs
mse = mean_squared_error(y_test, y_pred)
print("Mean squared error:", mse)
```

Fig. 4. LSTM code Snippet

2) MultiOutputRegressor : The MultiOutputRegressor is a technique used for solving regression problems with multiple output variables. It is particularly useful when there are multiple target variables that need to be predicted simultaneously, and the relationships between the input features and each target variable may differ.

MultiOutputRegressor

```
#MultiOutputRegressor
X = df_1.iloc[:, :-15].values
y = df_1.iloc[:, -15:].values

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a random forest regressor using MultiOutputRegressor
regressor = MultiOutputRegressor(RandomForestRegressor(n_estimators=100, random_state=42))
regressor.fit(X_train, y_train)

# Predict the output variables for the test set
y_pred = regressor.predict(X_test)

# Calculate the mean squared error between the predicted and actual outputs
mse = mean_squared_error(y_test, y_pred)
print("Mean squared error:", mse)
```

Fig. 5. MultiOutputRegressor code Snippet

3) Neural network model: The neural network model for regression is a flexible and powerful approach for solving regression problems. It can handle multiple output variables, making it suitable for cases where you need to predict multiple continuous values simultaneously.

neural network model

```
#neural network model
X = df_1.iloc[:, :-15].values
y = df_1.iloc[:, -15:].values

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Define the neural network model
model = Sequential()
model.add(Dense(50, input_dim=X.shape[1], activation='relu'))
model.add(Dense(10, activation='relu'))
model.add(Dense(15))

# Compile the model
model.compile(loss='mse', optimizer='adam',metrics=['accuracy'])

# Train the model
model.fit(X_train, y_train, epochs=1000, batch_size=5)

# Predict the output variables for the test set
y_pred = model.predict(X_test)

# Calculate the mean squared error between the predicted and actual outputs
mse = mean_squared_error(y_test, y_pred)
print("Mean squared error:", mse)
```

Fig. 6. Neural network model code Snippet

4) Kmeans: K-means clustering for LIWC (Linguistic Inquiry and Word Count) can be used to group individuals based on the patterns and similarities in their LIWC feature profiles. The K-means algorithm partitions the LIWC data into K distinct clusters, where each cluster represents a group of individuals with similar LIWC characteristics.

K-means Clustering

```
# Separate the input features and output variables
X = df_1.iloc[:, :-15].values
y = df_1.iloc[:, -15:].values
# Scale the input data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Define the K-means clustering model
model = KMeans(n_clusters=3, random_state=42)

model.fit(X_scaled)
# Get the cluster labels for each data point
cluster_labels = model.labels_

X_clustered = np.concatenate((X_scaled, cluster_labels.reshape((-1, 1))), axis=1)

X_train, X_test, y_train, y_test = train_test_split(X_clustered, y, test_size=0.2, random_state=42)
# Define the regression model
reg_model = LinearRegression()
# Train the regression model on the training data
reg_model.fit(X_train, y_train)

# Make predictions on the test data
y_pred = reg_model.predict(X_test)

# Evaluate the accuracy of the regression model
accuracy = reg_model.score(X_test, y_test)
print("Accuracy:", accuracy)
```

Fig. 7. Kmeans code Snippet

D. Model Evaluation

MSE stands for Mean Squared Error, which is a commonly used metric to evaluate the performance of regression models. It measures the average squared difference between the predicted values and the actual values in a regression problem. MSE is a non-negative value, with a lower MSE indicating a better fit of the regression model to the data. A MSE of 0 would indicate a perfect prediction where the predicted values exactly match the actual values. In the case of Mean Squared Error (MSE), a lower value is better. MSE measures the average squared difference between the predicted and actual values in a regression problem. By squaring the differences, MSE penalizes larger errors more heavily than smaller errors. Therefore, a lower MSE indicates that the

predicted values are closer to the actual values on average. When comparing different regression models or evaluating the performance of a single model, a lower MSE suggests better accuracy and a better fit to the data. It indicates that the model has lower overall prediction errors and is better at capturing the underlying patterns and relationships in the data.

1) For PCA based feature selection:

LSTM
Mean squared error: 0.04378527921900507

Trait	Actual Output	Predicted Output
Benevolence	0.9	0.8485898
Universalism	0.8	0.90760005
Self-direction	0.9	0.9471413
Stimulation	1.0	0.72462326
Hedonism	1.0	0.5741969
Achievement	0.9	0.64711475
Power	0.6	0.74457395
Security	0.8	0.89032745
Conformity	0.7	0.69685876
Tradition	0.9	0.91032803
Openness	0.575	0.7606051
Conscientiousness	0.5	0.5292755
Extraversion	0.594	0.53294194
Agreeableness	0.75	0.5842514
Neuroticism	0.531	0.52611405

MultiOutputRegressor
Mean squared error: 0.03300769961871702

Trait	Actual Output	Predicted Output
Benevolence	0.9	0.8313
Universalism	0.8	0.72797
Self-direction	0.9	0.816
Stimulation	1.0	0.719
Hedonism	1.0	0.592
Achievement	0.9	0.707
Power	0.6	0.717
Security	0.8	0.871
Conformity	0.7	0.692
Tradition	0.9	0.829
Openness	0.575	0.63459046
Conscientiousness	0.5	0.51003
Extraversion	0.594	0.53275
Agreeableness	0.75	0.55075
Neuroticism	0.531	0.49582

Neural network model
Mean squared error: 0.037709386297901895

Trait	Actual Output	Predicted Output
Benevolence	0.9	0.8266574
Universalism	0.8	0.705155
Self-direction	0.9	0.9560528
Stimulation	1.0	0.6620714
Hedonism	1.0	0.84992284
Achievement	0.9	0.77654254
Power	0.6	0.8136946
Security	0.8	0.89463913
Conformity	0.7	0.65855753
Tradition	0.9	0.8500715
Openness	0.575	0.7100212
Conscientiousness	0.5	0.4866228
Extraversion	0.594	0.49150893
Agreeableness	0.75	0.5989796
Neuroticism	0.531	0.54879653

Kmeans:

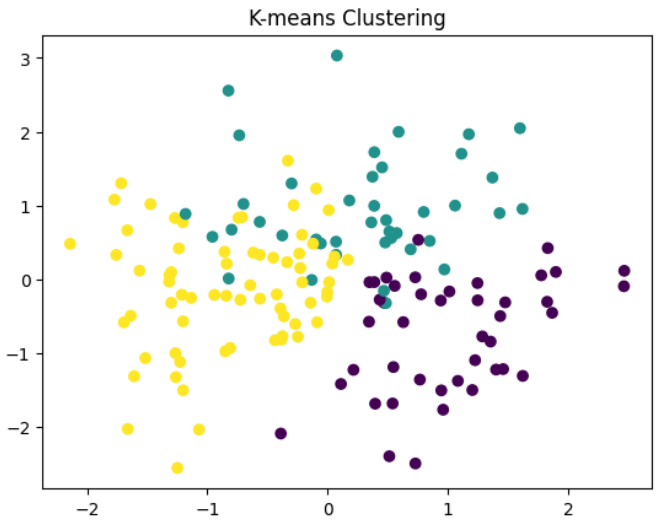

Fig. 8. Clusters

2) For pearson correlation based feature selection:

LSTM
Mean squared error: 0.026545864491735936

Trait	Actual Output	Predicted Output
Benevolence	0.9	0.8559785
Universalism	0.8	0.8012208
Self-direction	0.9	0.8031674
Stimulation	1.0	0.76268375
Hedonism	1.0	0.7316604
Achievement	0.9	0.6909019
Power	0.6	0.5399648
Security	0.8	0.802771
Conformity	0.7	0.73848414
Tradition	0.9	0.81720215
Openness	0.575	0.6215603
Conscientiousness	0.5	0.59876645
Extraversion	0.594	0.4979372
Agreeableness	0.75	0.6410849
Neuroticism	0.531	0.46835318

MultiOutputRegressor
Mean squared error: 0.03027711571916571

Trait	Actual Output	Predicted Output
Benevolence	0.9	0.8801
Universalism	0.8	0.72992503
Self-direction	0.9	0.769
Stimulation	1.0	0.587
Hedonism	1.0	0.701
Achievement	0.9	0.747
Power	0.6	0.703
Security	0.8	0.83518
Conformity	0.7	0.71532877
Tradition	0.9	0.815
Openness	0.575	0.59793092
Conscientiousness	0.5	0.58219
Extraversion	0.594	0.5023274
Agreeableness	0.75	0.63729242
Neuroticism	0.531	0.44621993

Neural network model
Mean squared error: 0.02624698829769021

Trait	Actual Output	Predicted Output
Benevolence	0.9	0.862
Universalism	0.8	0.8
Self-direction	0.9	0.75
Stimulation	1.0	0.798
Hedonism	1.0	0.854
Achievement	0.9	0.902
Power	0.6	0.78
Security	0.8	1.025
Conformity	0.7	0.817
Tradition	0.9	0.993
Openness	0.57	0.635
Conscientiousness	0.5	0.579
Extraversion	0.59	0.45
Agreeableness	0.75	0.657
Neuroticism	0.53	0.557

Kmeans:

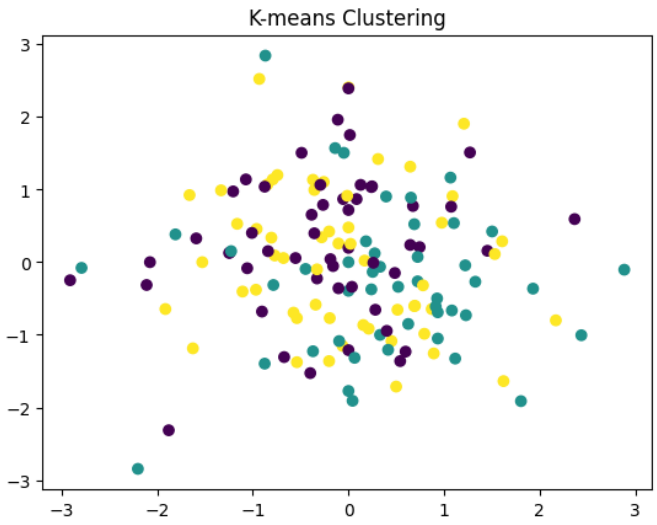

Fig. 9. Clusters

IV. Conclusion

In this study, the application of PCA for clustering the LIWC dataset led to improved clustering results of K-means clustering. The reduced-dimensional representation obtained through PCA allowed for better separation and grouping of data points based on their underlying patterns and similarities. This suggests that the intrinsic structure of the LIWC dataset was more effectively captured and utilized for clustering through PCA.