Password Strength Prediction using NLP

**Overview**

This project leverages Natural Language Processing (NLP) techniques to predict the strength of passwords. The aim is to classify passwords into different strength categories (weak, medium, strong) based on various features extracted from the passwords. The project includes data cleaning, feature engineering, statistical analysis, and machine learning model training to achieve accurate predictions.

## Table of Contents

* [Overview](#overview)
* [Data Collection](#data-collection)
* [Data Cleaning](#data-cleaning)
* [Semantic Analysis](#semantic-analysis)
* [Feature Engineering](#feature-engineering)
* [Descriptive Statistics](#descriptive-statistics)
* [TF-IDF Computation](#tf-idf-computation)
* [Machine Learning Model](#machine-learning-model)
* [Model Evaluation](#model-evaluation)
* [Insights](#insights)
* [Installation](#installation)
* [Usage](#usage)

**Data Collection**

The dataset is sourced from an SQLite database named password\_data.sqlite. The passwords and their respective strengths are stored in a table called Users.

**Data Cleaning**

Basic data cleaning was performed, including removing irrelevant features and handling missing values. The order\_date column was parsed to extract the date part for further analysis.

**Semantic Analysis**

Various analyses were performed to understand the characteristics of the passwords:

* **Numeric Characters**: Count of passwords containing only numeric characters.
* **Upper-case Characters**: Count of passwords containing only upper-case characters.
* **Alphabetic Characters**: Count of passwords containing only alphabetic characters.
* **Alpha-numeric Characters**: Count of passwords containing alpha-numeric characters.
* **Title-case Characters**: Count of passwords containing title-case characters.
* **Special Characters**: Count of passwords containing special characters.

**Feature Engineering**

The following features were extracted to predict password strength:

* **Length of password**
* **Frequency of Lowercase Characters**
* **Frequency of Uppercase Characters**
* **Frequency of Numeric Characters**
* **Frequency of Special Characters**

**Descriptive Statistics**

Boxplots were used to visualize the relationship between password strength and various features. The analysis revealed:

* Higher length correlates with higher strength.
* Higher lowercase frequency is seen in low-strength passwords.
* Nicely proportioned passwords (with a mix of character types) are stronger.

fig , ((ax1 , ax2) , (ax3 , ax4) , (ax5,ax6)) = plt.subplots(3 , 2 , figsize=(15,7))

sns.boxplot(x="strength" , y='length' , hue="strength" , ax=ax1 , data=data)

sns.boxplot(x="strength" , y='lowercase\_freq' , hue="strength" , ax=ax2, data=data)

sns.boxplot(x="strength" , y='uppercase\_freq' , hue="strength" , ax=ax3, data=data)

sns.boxplot(x="strength" , y='digit\_freq' , hue="strength" , ax=ax4, data=data)

sns.boxplot(x="strength" , y='special\_char\_freq' , hue="strength" , ax=ax5, data=data)

plt.subplots\_adjust(hspace=0.6)

## TF-IDF Computation

TF-IDF values for each character in the passwords were computed to understand their significance.

## Machine Learning Model

A Multinomial Logistic Regression model was used to classify passwords into three strength categories (weak, medium, strong).

from sklearn.linear\_model import LogisticRegression

clf = LogisticRegression(multi\_class="multinomial")

## Model Evaluation

The model's performance was evaluated using precision, recall, and F1-score.

print(classification\_report(y\_test , y\_pred))

# Output:

# precision recall f1-score support

# 0 0.52 0.29 0.37 2752

# 1 0.82 0.94 0.88

**Insights**

From the descriptive statistics and modeling results, we observe:

* Higher password length generally indicates higher strength.
* Passwords with diverse character types (lowercase, uppercase, digits, special characters) are stronger.
* Lowercase frequency is higher in weaker passwords.

**Future Work**

* **Expand Dataset**: Collect more diverse and extensive password data to improve model accuracy.
* **Advanced Models**: Explore advanced NLP models like transformers for better feature extraction and prediction.
* **User Education**: Implement features to educate users on best practices for creating strong passwords.