Topic: College Event Feedback Analysis

Introduction:

In this project, interns will analyze **text and rating-based feedback** submitted by students after attending campus events. You'll work with **simulated or real Google Forms data (CSV)** and use basic **Natural Language Processing (NLP)** to understand satisfaction levels and identify areas for improvement.

Tools used:

Python Programming

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.
It is used for:

- **web development (server-side),**
- **software development,**
- **mathematics,**
- **system scripting.**

Why Python?

- **Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).**
- **Python has a simple syntax similar to the English language.**
- **Python has syntax that allows developers to write programs with fewer lines than some other programming languages.**
- **Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.**
- **Python can be treated in a procedural way, an object-oriented way or a functional way**

Pandas Dataframe:

- Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data. The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.

Data visualization:

Matplotlib.pyplot

Most of the Matplotlib utilities lies under the pyplot submodule, and are usually imported under the plt alias: import matplotlib.pyplot as plt Now the Pyplot package can be referred to as plt.

Matplotlib is a powerful and versatile open-source plotting library for Python, designed to help users visualize data in a variety of formats. Developed by John D. Hunter in 2003, it enables users to graphically represent data, facilitating easier analysis and understanding

Seaborn

Seaborn is a powerful Python library built on top of Matplotlib, designed specifically for creating informative and aesthetically pleasing statistical graphics. It simplifies the process of generating complex visualizations by providing a high-level interface and offering beautiful default styles and color palettes

Linear Regression using sklearn

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between the dependent and independent variables, they are considering and the number of independent variables being used. This article is going to demonstrate how to use the various Python libraries to implement linear regression on a given dataset. We will demonstrate a binary linear model as this will be easier to visualize.

imported libraries

- import numpy as np
- import pandas as pd
- import seaborn as sns
- import matplotlib.pyplot as plt
- from sklearn import preprocessing, svm
- from sklearn.model_selection import train_test_split
- from sklearn.linear_model import LinearRegression

Steps involved

Exploratory Data Analysis

1.Data collection: Gathering and collecting data from various sources.
2.Data cleaning: Checking the data for missing values, outliers, and inconsistencies and handling them appropriately. It may involve imputing missing values, removing outliers, or transforming the data.
3.Data visualization: Creating visual representations of the data using graphs, charts, and other visual aids. It helps to identify patterns and relationships in the data and gain insights into its characteristics.
4.Data exploration: Analyzing the data to identify trends, relationships, and patterns. It may involve computing summary statistics such as mean, median, standard deviation, and correlation coefficients.
5.Data modeling: Building statistical or machine learning models to make predictions or draw conclusions from the data.
6.Data communication: Presenting the analysis results to stakeholders clearly and concisely

Sentiment Analysis using VADER

- VADER stands for Valence Aware Dictionary and sentiment Reasoner.
- It’s a rule-based sentiment analysis tool built specifically for text from social media, customer feedback, and informal communication.
- Comes preloaded with a sentiment lexicon (a dictionary of words with positive/negative intensity scores).

Importing for

- from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
-
- analyzer = SentimentIntensityAnalyzer()
-
- df["sentiment_score"] = df["Questions"].astype(str).apply(lambda x: analyzer.polarity_scores(x)["compound"])
- df["sentiment"] = df["sentiment_score"].apply(lambda x: "Positive" if x>0.05 else ("Negative" if x<-0.05 else "Neutral"))
-
- print(df["sentiment"].value_counts())

NLTK:

NLTK stands for Natural Language Toolkit.
It’s one of the most widely used Python libraries for Natural Language Processing (NLP).
Think of it as a toolbox for working with text data: it helps you clean, process, analyze, and even model human language.

Import for

- import nltk
- from nltk.sentiment.vader import SentimentIntensityAnalyzer

Features engineering

Feature Engineering is the process of transforming raw data into meaningful inputs (features) for machine learning models.

Why is it Important?

- Raw data isn’t enough → ML models don’t understand dates, text, or categories directly.
- Better features = better models → Sometimes a simple model with great features beats a complex model with poor features.
- Helps with improving accuracy, reducing bias, and faster training.

Logistic Regression

- Logistic Regression is a supervised machine learning algorithm used for classification (not regression, despite the name).
- It predicts the probability of a data point belonging to a certain class (e.g., Yes/No, Spam/Not Spam, 0/1).

Import for

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
  from sklearn.metrics import classification_report
```

Conclusion & Future Work

This project analyzed student satisfaction survey data to uncover key insights:

- Sentiment Analysis showed the proportion of positive, neutral, and negative feedback.
- Word Clouds & Visualizations highlighted recurring themes in student opinions.
- A TF-IDF + Logistic Regression model was built to classify feedback sentiment automatically.
- Insights can guide educators to improve teaching quality, event organization, and student support

Future Development

To make this project more powerful, future work could include:

- **Advanced NLP Models**
- **Use deep learning models like BERT, RoBERTa, or DistilBERT for more accurate sentiment detection.**
- **Aspect-Based Sentiment Analysis (ABSA)**
- **Instead of just “positive/negative,” analyze specific aspects (e.g., faculty, facilities, events, curriculum).**
- **Integration with Dashboards**
- **Build an interactive Power BI or Streamlit dashboard so administrators can monitor student sentiment in real-time.**
- **Feedback Recommendation System**
- **Suggest improvements based on common student complaints.**
- **Multilingual Support**
- **Extend analysis to handle feedback in multiple languages using translation + NLP models.**
- **Predictive Analytics**
- **Predict overall student satisfaction score or dropout risk based on survey patterns.**