# MOVIE REVIEW SENTIMENT ANALYSIS USING MACHINE LEARNING

*Presented By:* **Dipa Khadka & Manis Chaudhary**

- *Dataset:* IMDb 50,000 Movie Reviews
- *Models:* Logistic Regression, SVM, KNN

# INTRODUCTION

Sentiment analysis is a powerful Natural Language Processing (NLP) technique used to automatically determine whether a piece of text expresses a **positive** or **negative** opinion.
It is a form of text classification that helps computers understand human emotions.

Have you ever wanted to know whether a movie is worth watching without reading thousands of reviews?

This project solves that problem by training a machine learning model that automatically predicts whether a movie review is **positive** or **negative**.

# WHY THIS PROJECT?

Sentiment analysis helps summarize overall audience opinions

Thousands of movie reviews can be overwhelming to read manually

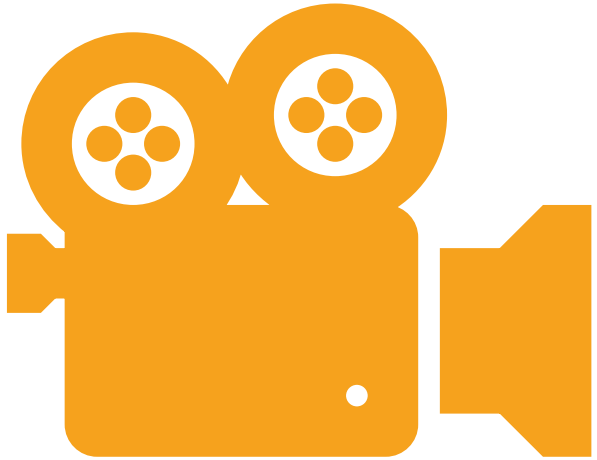
Useful for:

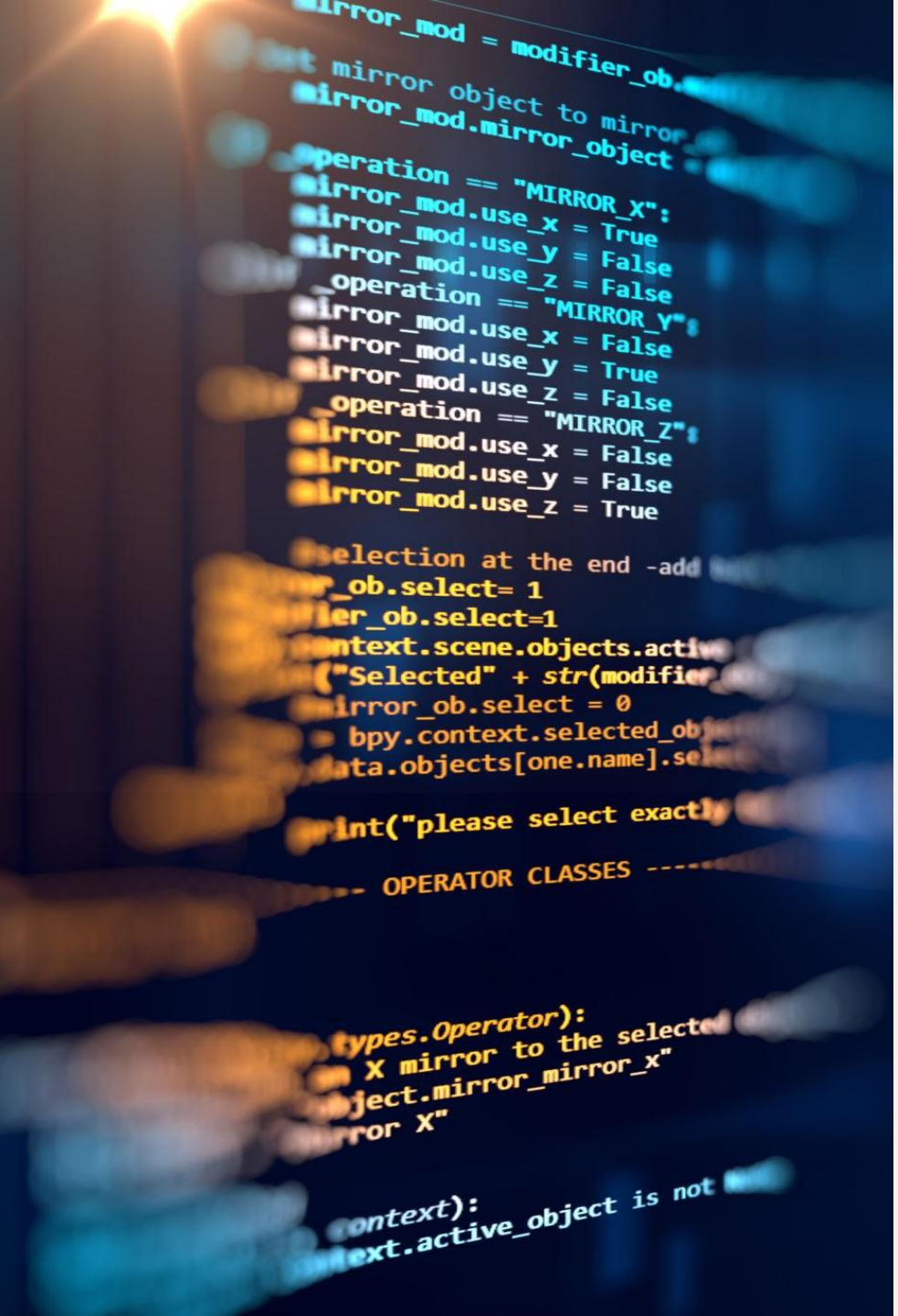
- Movie studios
- Streaming platforms
- Recommendation engines
- Viewers deciding what to watch

build a machine learning model that predicts sentiment of movie reviews.

# DATASET OVERVIEW

- We used the **IMDb Movie Review Dataset**, which contains:
- **50,000** movie reviews
- Each labeled as **positive** or **negative**
- Balanced dataset: equal number of each class
- After removing a few duplicates (~0.8%), the final dataset contains **49,582** reviews
- sentiment — positive (1) or negative (0)
- This makes it an excellent dataset for binary sentiment classification.






# DATA CLEANING

- Removal of HTML tags
- Lowercasing all text
- Removing punctuation & special characters
- Removing extra spaces
- Removing stopwords
- Creating a new column: **clean_review**

➡️ Result: Clean, standardized text ready for modeling.

# TF-IDF VECTORIZATION

TF-IDF converts text into meaningful numerical features.

**Parameters used:**

- max_features = 50,000
- Ngram range = (1,2) (unigrams + bigrams)
- Stop words = "english"

**Why TF-IDF?**

- Highlights important words
- Works extremely well for ML models
- Produces high-dimensional sparse vectors ideal for Logistic Regression & SVM

# MODELS USED IN THIS PROJECT

✓ **Logistic Regression**

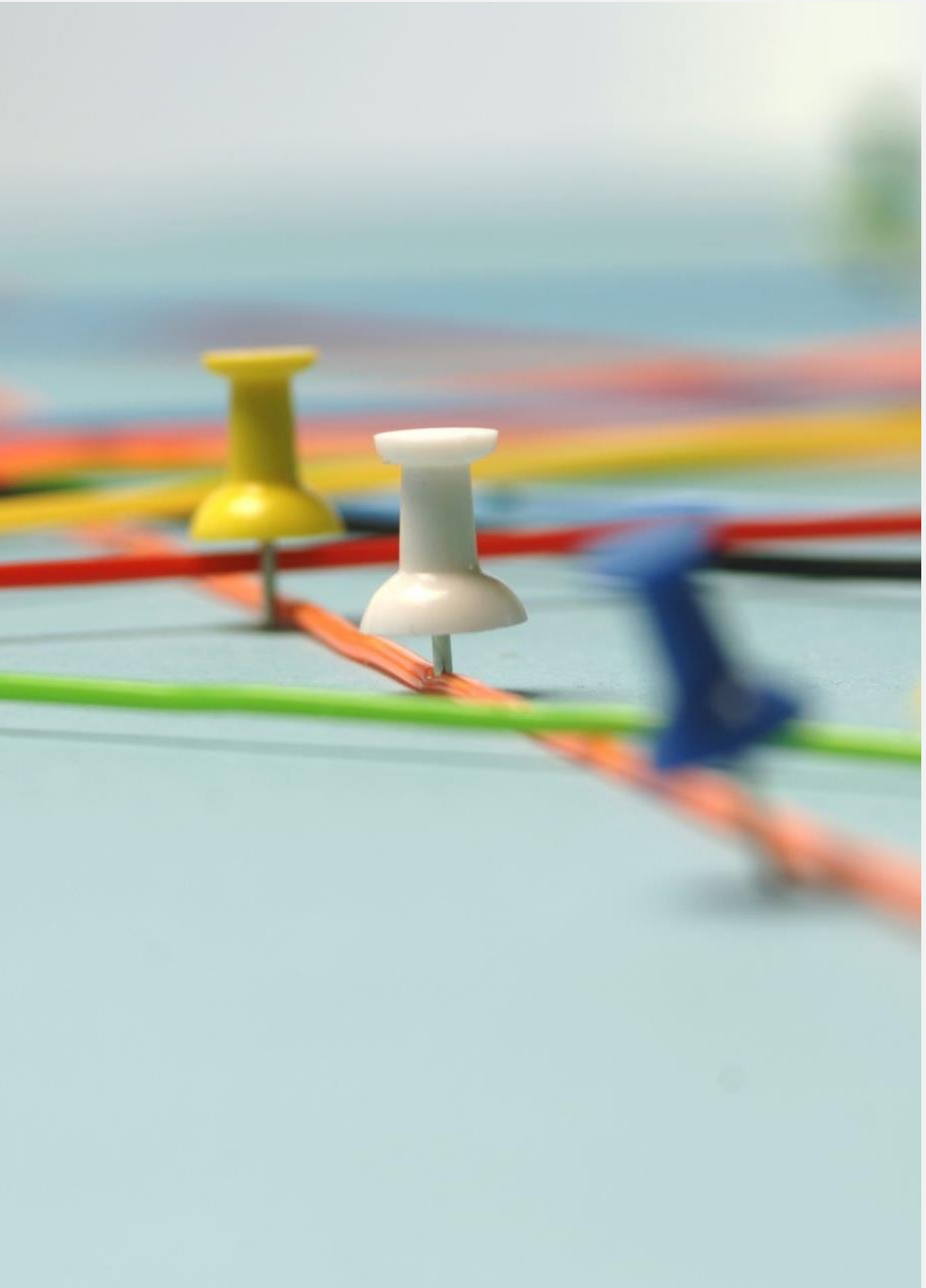
- Fast and highly effective for text data
- Performs well with TF-IDF representations

✓ **LinearSVC (Support Vector Machine)**

- Best model for high-dimensional text
- Often achieves the highest accuracy
- Works superbly with sparse TF-IDF vectors

✓ K-Nearest Neighbors (KNN)

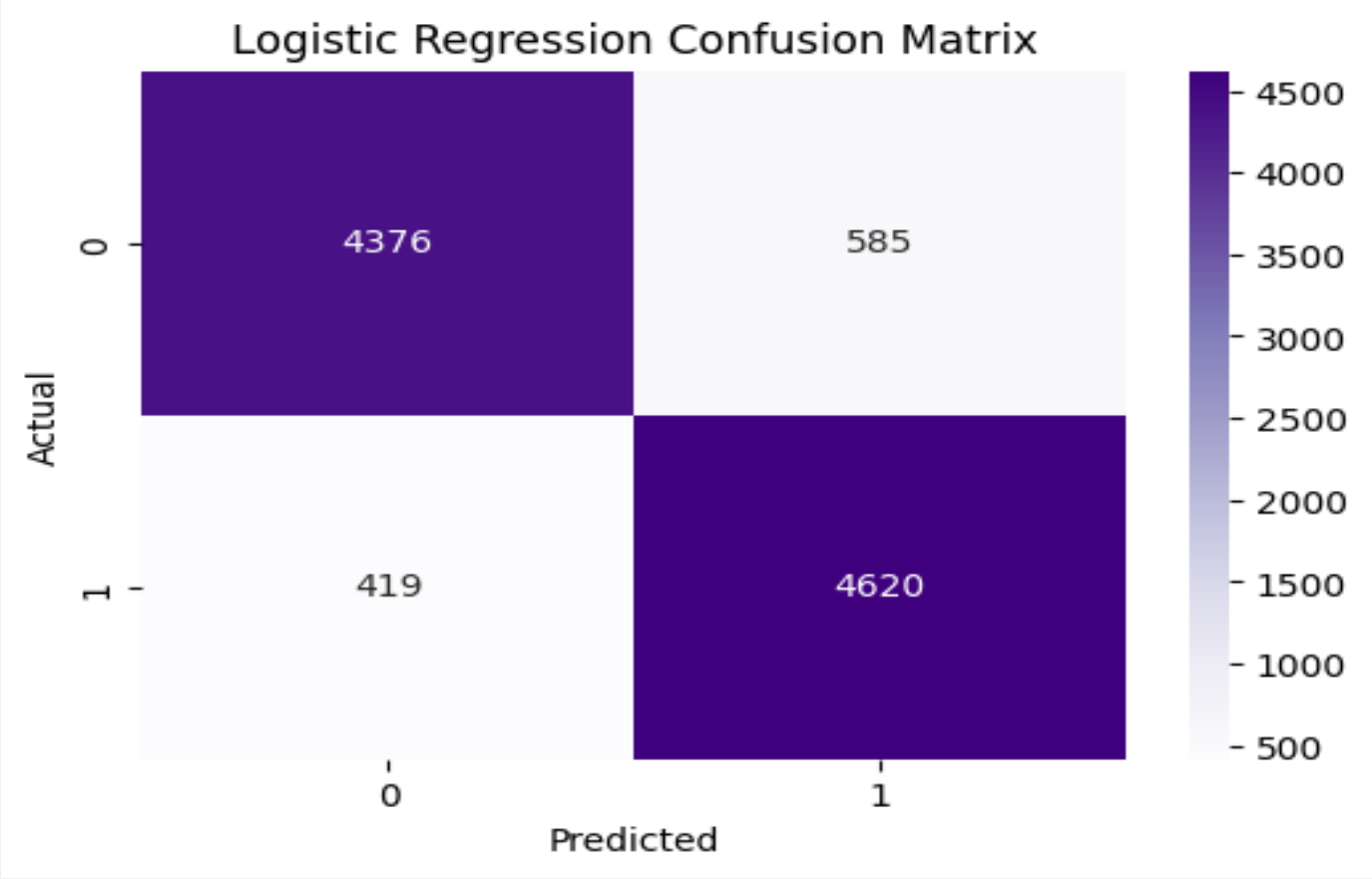
- Simple and intuitive algorithm
- Performance depends on the choice of **K value**
- Lower accuracy compared to linear models on text, but useful for demonstrating **overfitting vs. generalization behavior**
- We used **TF-IDF Vectorization** to convert text into numerical features suitable for machine learning.



# MACHINE LEARNING MODEL TRAINING

- Each model was trained using an **80/20 train-test split** on the cleaned dataset.
- **4. Evaluation Metrics**
- We calculated:
- Accuracy
- Precision
- Recall
- F1-score
- Confusion matrix
- These metrics help understand how well the model distinguishes positive vs negative reviews.

# MODEL 1- LOGISTIC REGRESSION




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.91 | 0.88 | 0.90 | 4961 |
| 1 | 0.89 | 0.92 | 0.90 | 5039 |
| accuracy | | | 0.90 | 10000 |
| macro avg | 0.90 | 0.90 | 0.90 | 10000 |
| weighted avg | 0.90 | 0.90 | 0.90 | 10000 |

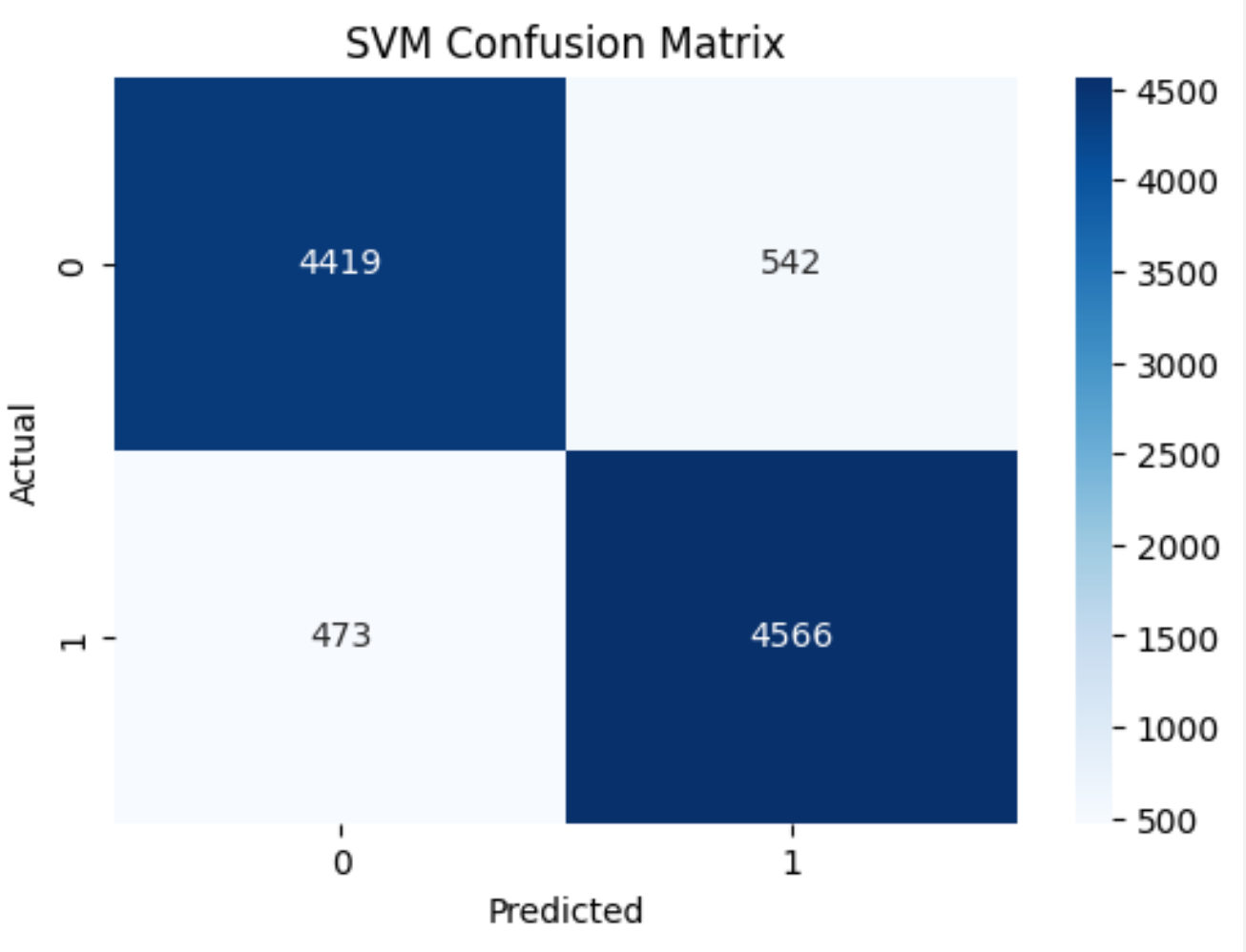
Logistic Regression Accuracy: 0.8996

**Why Logistic Regression Performs Well:**

- Works extremely well on high-dimensional TF-IDF features
- Very fast and efficient
- Avoids overfitting due to regularization
- Creates a strong baseline for text classification

**Confusion Matrix Insight:**

- Predicts both positive and negative reviews with high accuracy
- Errors are low and evenly distributed



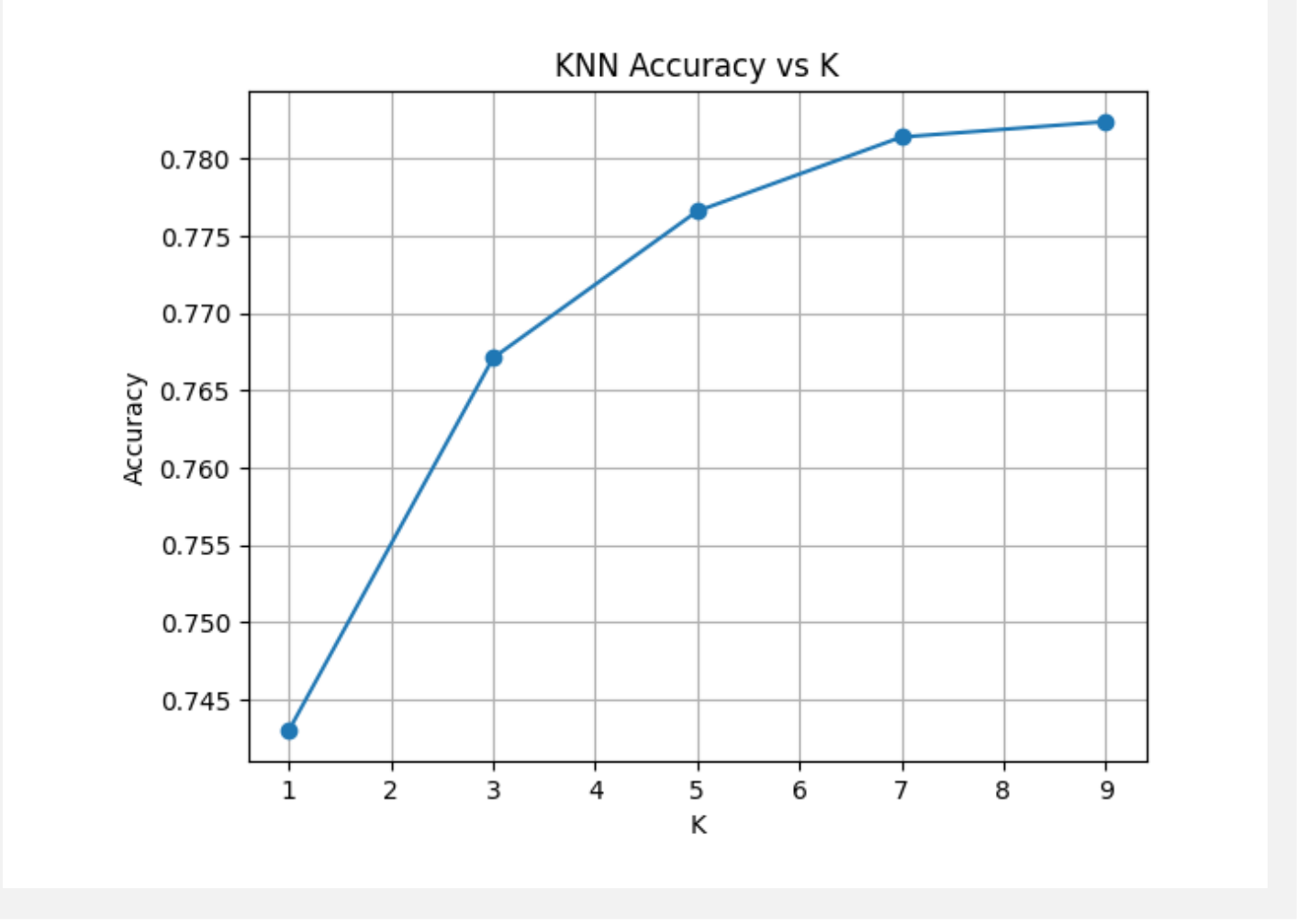

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.90 | 0.89 | 0.90 | 4961 |
| 1 | 0.89 | 0.91 | 0.90 | 5039 |
| accuracy | | | 0.90 | 10000 |
| macro avg | 0.90 | 0.90 | 0.90 | 10000 |
| weighted avg | 0.90 | 0.90 | 0.90 | 10000 |

# MODEL 2-SVM (LINEAR SVC)

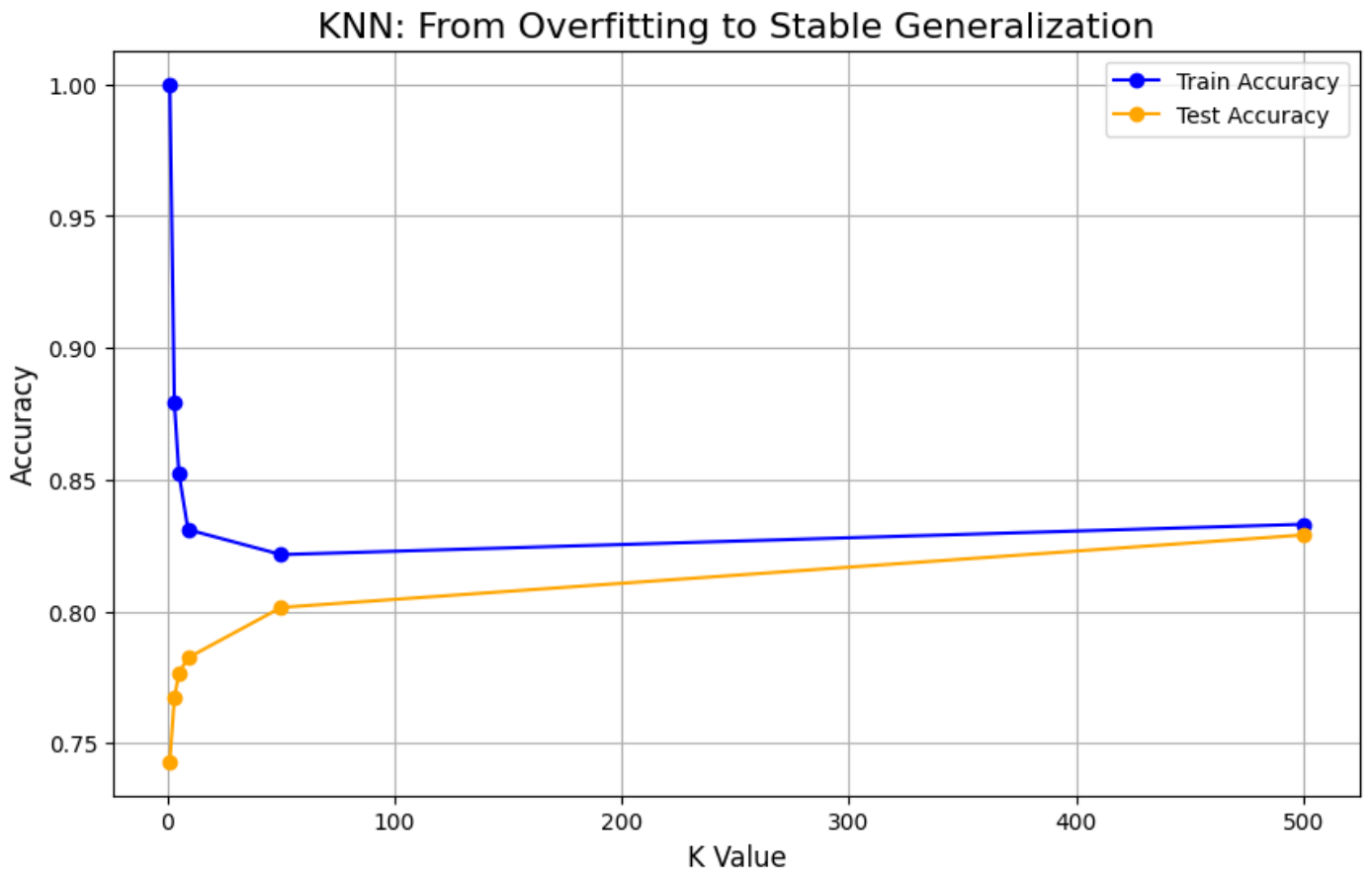
- SVM Accuracy: 0.8985
- The SVM model achieved an accuracy of **89.85%** on the test data.
- The precision and recall values for both positive and negative sentiment are close to 0.90, showing balanced and reliable performance.
- The confusion matrix indicates that SVM correctly classifies most reviews, with very few misclassifications.
- SVM performs especially well on TF-IDF text data because it handles high-dimensional features effectively.

# KNN

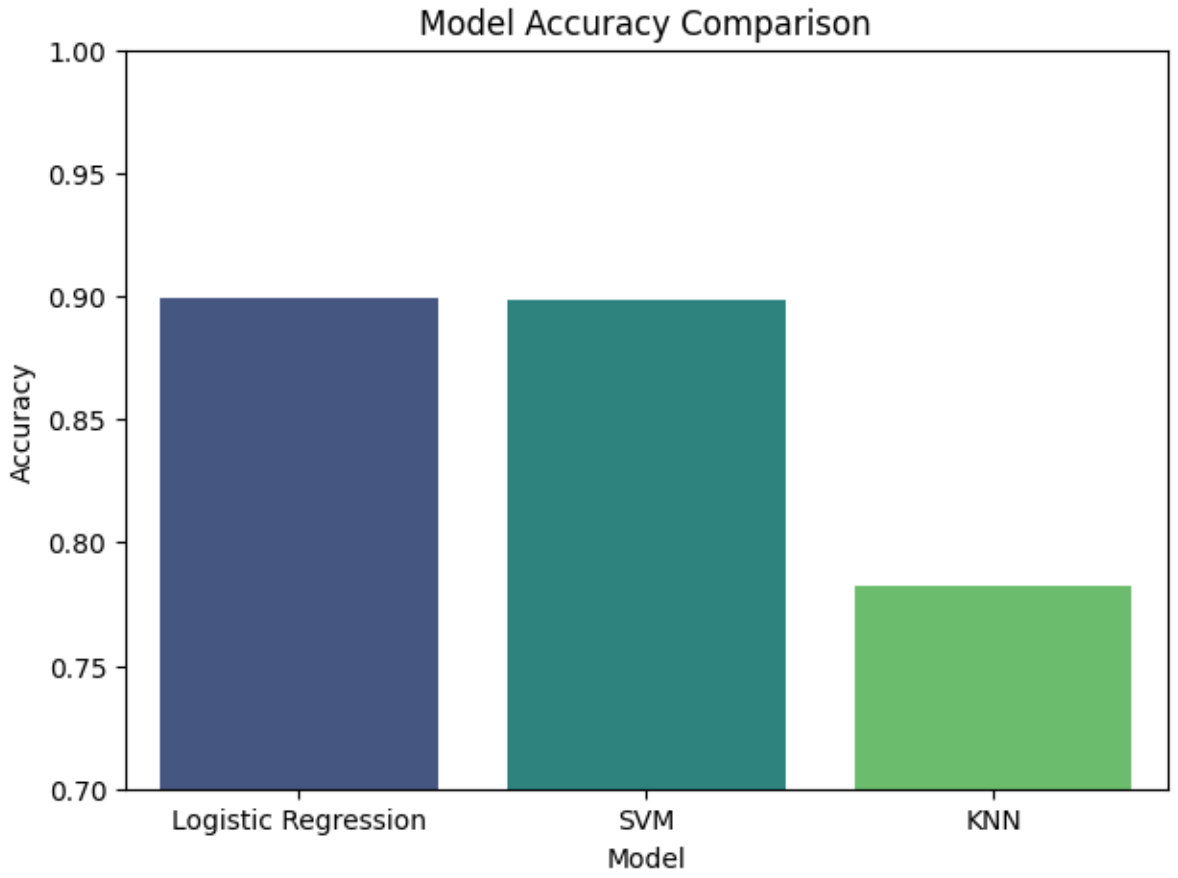
- KNN performance depends heavily on the value of **K** (number of neighbors).
- **Low K (e.g., K=1)** → High variance → Overfitting
- **Higher K** → More stable predictions
- Accuracy improved as K increased from **1 → 9**
- Best accuracy in this experiment: **K = 9 (≈ 78%)**
- KNN performs weaker compared to Logistic Regression and SVM because text data is high-dimensional.



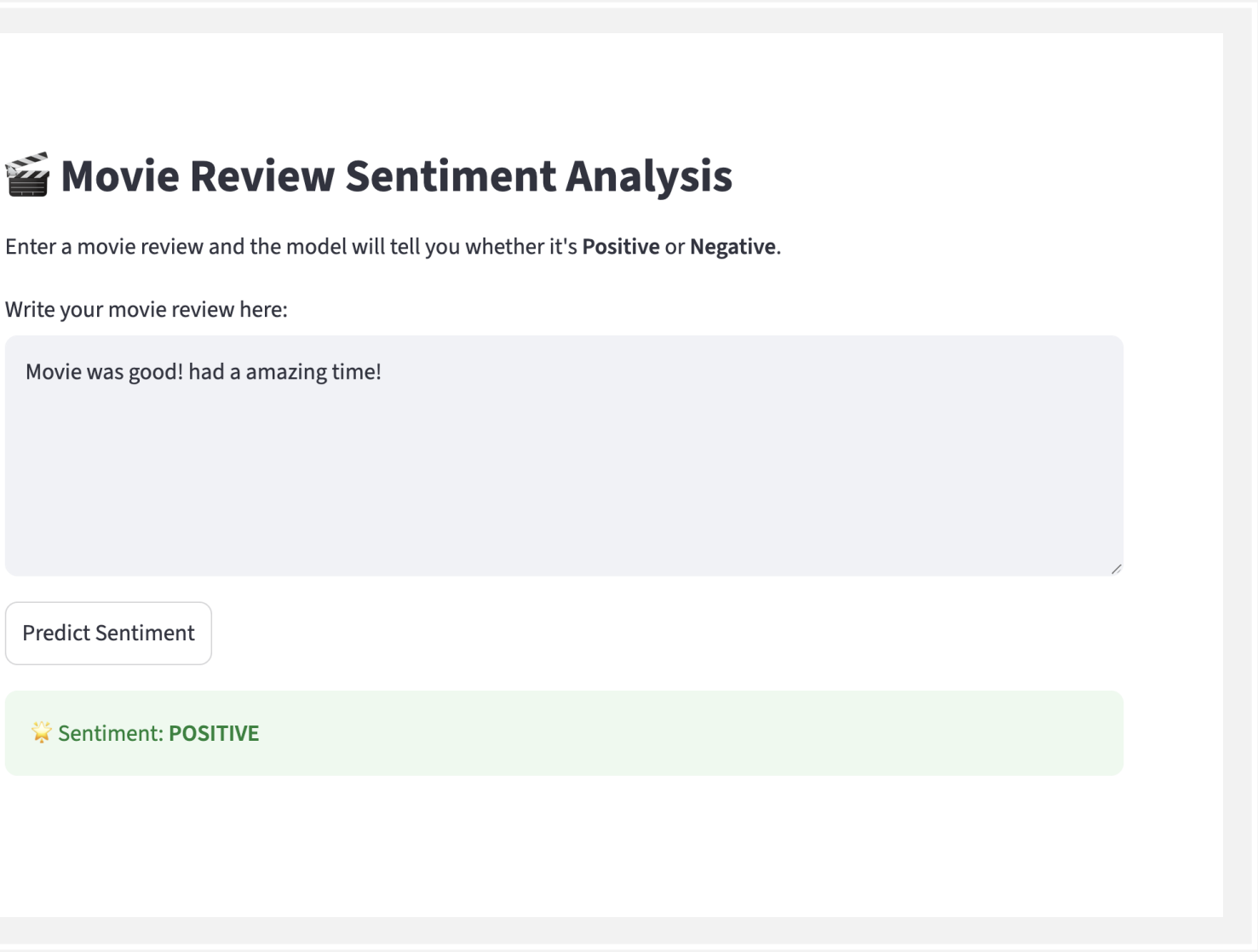
KNN


KNN: From Overfitting to Stable Generalization
Train Accuracy
Test Accuracy
Accuracy
K Value
1.00
0.95
0.90
0.85
0.80
0.75
0
100
200
300
400
500

# MODEL COMPARISON



STREAMLIT


🎬 Movie Review Sentiment Analysis
Enter a movie review and the model will tell you whether it's Positive or Negative.
Write your movie review here:
Movie was good! had a amazing time!
Predict Sentiment
🌟 Sentiment: POSITIVE

# CONCLUSION

- Cleaned and processed IMDb reviews using NLP techniques
- Built three ML models: Logistic Regression, SVM, and KNN
- Logistic Regression and SVM achieved the highest accuracy (~90%)
- KNN helped demonstrate overfitting vs. generalization
- TF-IDF proved effective for converting text into features
- Built a Streamlit app to make the model interactive
- Overall, ML models can successfully classify movie reviews into Positive or Negative

Project Link:
https://github.com/imdeepa99/movie-review-sentiment-analysis-project

# ANY QUESTIONS?

dkhad3@unh.newhaven.edu
mchau13@unh.newhaven.edu

# THANK YOU! HAVE A GOOD DAY!