ASSIGNMENT

By

Sia

2021A1r047

Semester 7th

Department of Computer Science and Engineering

Model Institute of Engineering & Technology (Autonomous)

(Permanently Affiliated to the University of Jammu, Accredited by NAAC with "A" Grade)

Jammu, India

2024

ASSIGNMENT

Course Name – Soft Computing

Question Number	Course Outcomes	Blooms' Level	Maximum Marks	Marks Obtain
Q1	CO1	3-4	10	
Q2	CO2 , CO5	3-4	10	
Total Marks			20	
Faculty Signature Email:				

Assignment Objectives:

The objective of this assignment is to deepen the understanding of financial management by exploring the practical implications of the theoretical concepts.

Q. No.	Question	BL	CO	Marks	Total Marks
1	Perform sentiment analysis on the IMDB Movie Review Dataset Using Artificial Neural Networks (ANN).	3,4	CO1	10	10
2	Optimize the model using Grid Search or Random Search for hyperparameters such as the number of layers, learning rate, activation functions, and batch size. Evaluate the model's performance using metrics such as accuracy, F1-score, and ROC-AUC	3,4	CO2,CO5	10	10

OVERVIEW

Sentiment analysis is a technique in natural language processing (NLP) used to identify and extract subjective information from text. Sentiment analysis, a key area in natural language processing (NLP), involves determining the sentiment expressed in a text, which can be classified as positive, negative, or neutral. This research applies sentiment analysis to the IMDb movie review dataset, using an Artificial Neural Network (ANN) for binary classification of reviews into positive or negative sentiment. The IMDb dataset, a rich collection of user-generated movie reviews, provides a large sample of labeled data for model training and evaluation.
An artificial neural network with one or more hidden layers is then trained to classify the sentiment of the reviews. The model's performance is evaluated based on metrics such as accuracy, precision, recall, and F1-score.

The results indicate that the ANN model performs well in classifying the sentiment of movie reviews, achieving a high level of accuracy and reliability. The model is capable of capturing complex relationships between words and their sentiment, which traditional machine learning models may struggle with.

This demonstrates the effectiveness of using Artificial Neural Networks for sentiment analysis of movie reviews, showcasing their potential in extracting valuable insights from textual data. The findings suggest that ANN-based sentiment analysis can complement existing recommendation systems and provide a deeper understanding of user opinions, aiding marketers, movie studios, and audiences in decision-making processes.

Contents

CHAPTER 1

PYTHON PROGRAMMING

Introduction 1.1

Python is a high-level, versatile, and dynamically typed programming language known for its simplicity and readability. Created by Guido van Rossum and first released in 1991, Python has gained immense popularity among developers, data scientists, and engineers due to its ease of use and extensive standard libraries

Python is a multiparadigm programming language, supporting procedural, object-oriented, and functional programming styles. It can be used for various applications, including web development, data analysis, artificial intelligence, scientific computing, automation, and more. Python provides a vast collection of libraries and frameworks that simplify complex tasks. For example, NumPy and Pandas are used for data manipulation and analysis, Flask and Django for web development, TensorFlow and PyTorch for machine learning, and Matplotlib for data visualization.

Python is an interpreted language, which means that code written in Python is executed line by line by the Python interpreter. This feature facilitates rapid development and debugging. It is platform-independent, allowing developers to write code on one platform and run it on various operating systems without modification.

Python has become the go-to language for data scientists and AI researchers. Libraries like NumPy, Pandas, Scikit-learn, TensorFlow, and PyTorch provide powerful tools for data analysis, machine learning, deep learning, and natural language processing. Its simplicity and ease of integration have made it the preferred choice for professionals working with complex datasets and developing cutting-edge AI applications.

Features 1.2

Python is a language rich in features that define its significance as a high- level language and are the reasons behind its user friendly nature.

Easy to Read and Learn:

Python's simple and clean syntax emphasizes readability and reduces the cost of program maintenance.

Interpreted Language:

Python is an interpreted language, which means that code written in Python is executed line by line by the Python interpreter.

High-Level Language:

Python abstracts low-level details, providing a high-level interface to developers. This simplifies complex algorithms and allows developers to focus on solving problems.

Object-Oriented:

Python supports object-oriented programming (OOP) principles, enabling the creation and use of objects and classes, encouraging modularity, reusability, and a more organized code structure.

Dynamically Typed:

Python is dynamically typed, meaning that variable types are interpreted during runtime. Developers do not need to specify variable types explicitly.

Interoperability:

Python can be integrated with other languages like C, C++, and Java.

Applications 1.3

Thanks to its high range of versatility, Python is used across various platforms for a wide variety of tasks and operations.

Data Science and Analysis:

Python, along with libraries like Pandas, NumPy, and SciPy, is widely used for data analysis, manipulation, and visualization.

Machine Learning and Artificial Intelligence:

Python libraries like TensorFlow, PyTorch, and scikit-learn provide powerful tools for developing machine learning models and neural networks.

Web Development:

Python frameworks like Django, Flask, and Pyramid enable developers to build robust and scalable web applications.

Scientific Computing:

Python is extensively used in scientific computing, simulations, and mathematical modeling. Libraries like SciPy and NumPy offer support for scientific computations,

Natural Language Processing (NLP):

Python's NLTK (Natural Language Toolkit) and SpaCy libraries are widely used in natural language processing tasks such as text analysis, sentiment analysis, and language translation.

CHAPTER 2

GOOGLE COLAB

Introduction 2.1

Google Colab (short for Colaboratory) is a cloud-based platform provided by Google for writing and executing Python code in an interactive, notebook-style environment. It is similar to Jupyter Notebooks but with several added features that enhance the experience for data scientists, machine learning practitioners, and researchers. Some key aspects of Google Colab include:

1. **Free GPU/TPU Support:**

   Colab provides access to free computing resources, including GPUs (Graphics Processing Units) and TPUs (Tensor Processing Units), which are particularly useful for deep learning tasks.

2. **Cloud-based**:

   Since Colab runs in the cloud, it eliminates the need to set up a local environment, and the notebooks can be accessed and run from anywhere with an internet connection.

3. **Integration with Google Drive**:

   Colab allows users to easily save, load, and share notebooks via Google Drive. This makes collaboration seamless, as multiple people can edit and run the same notebook simultaneously.

4. **Pre-installed Libraries**:

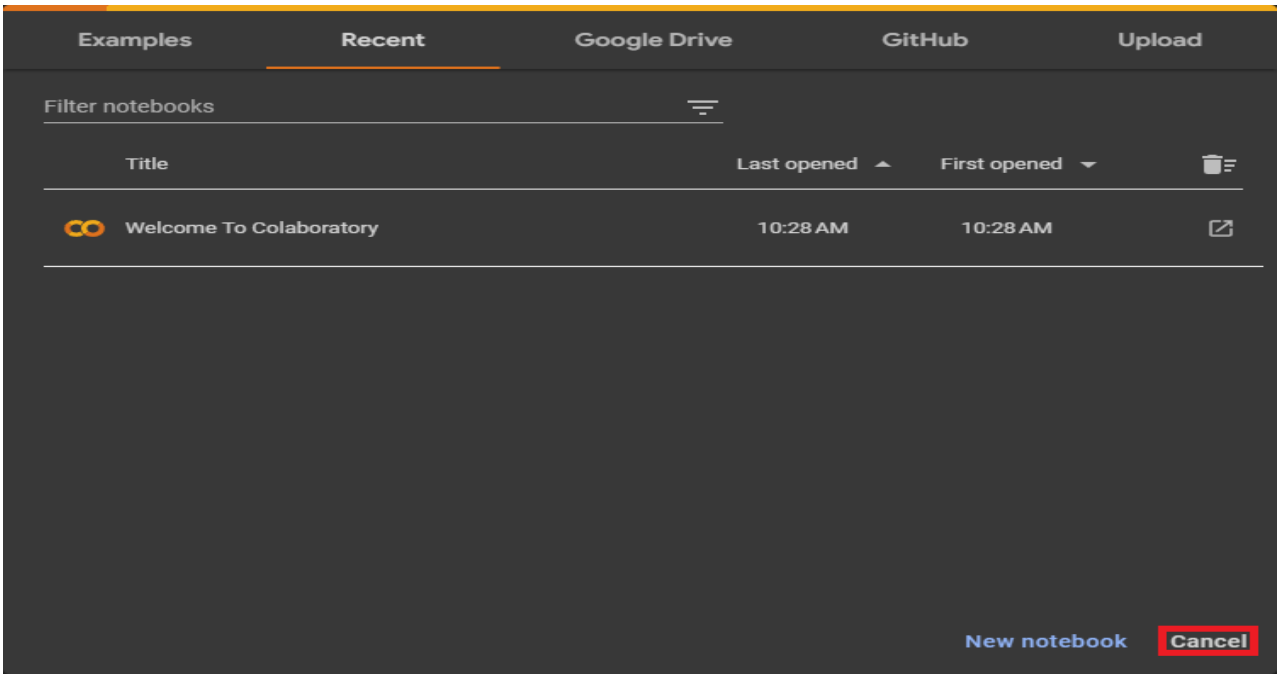
   Colab comes with many popular Python libraries pre-installed, including TensorFlow, PyTorch, NumPy, Pandas, and Matplotlib, which streamlines the setup process for data science and machine learning tasks.

5. **Interactive Features**:

   You can include rich text, visualizations, and even interactive widgets within the notebooks, making them a great tool for educational purposes, presentations, and exploratory data analysis.

6. **Collaboration**:

   Multiple users can work on a single notebook in real-time, similar to Google Docs, with version control and comment features for better teamwork.



Google Colab Homepage

Features 2.2

Google Colab offers several key features:

1. **Free Access to GPUs/TPUs**:

   Users can leverage powerful hardware like GPUs and TPUs for machine learning tasks without any cost.

2. **Cloud-based**:

   No installation needed; your notebooks are stored in Google Drive and accessible from anywhere.

3. **Real-time Collaboration**:

   Multiple users can work on a notebook simultaneously, similar to Google Docs.

4. **Pre-installed Libraries**:

   Popular Python libraries like TensorFlow, Keras, PyTorch, NumPy, and Pandas are available out of the box.

5. **Interactive Notebooks**:

   Rich text, visualizations, and code can be embedded into a single notebook, enhancing interactivity.

6. **Python 2 and 3 Support**:

   Users can switch between Python versions, depending on the project needs.

7. **Easy Sharing**:

   Notebooks can be easily shared with others via links, making collaboration straightforward.

8. **Mount Google Drive**:

   Direct integration with Google Drive allows for easy data storage and access within notebooks.

9. **Custom Libraries**:

   You can install additional libraries or dependencies directly into the environment.

Applications 2.3

Google Colab is widely used across various domains due to its ease of use, accessibility, and powerful features. Here are some key applications of Google Colab:

Machine Learning and Deep Learning

- *Training Models*: Colab provides free access to GPUs and TPUs, making it ideal for training machine learning models, especially deep learning models that require intensive computation.
- *Model Evaluation and Hyperparameter Tuning*: It is commonly used for evaluating models, fine-tuning hyperparameters, and experimenting with different architectures.

Data Science and Analytics

- *Data Exploration and Analysis*: Colab allows easy integration with libraries like Pandas, NumPy, and Matplotlib, making it great for exploratory data analysis (EDA) and visualization.
- *Data Cleaning*: Users can process large datasets and clean them by utilizing Colab's powerful computational resources.

Educational Purposes

- *Interactive Learning*: Teachers and learners use Colab to create interactive tutorials, assignments, and educational content, where users can run code, modify it, and immediately see the results.
- *Courses and Workshops*: Colab is widely used in online courses for hands-on coding exercises and demonstrations, especially in machine learning and data science.

Research and Prototyping

- *Rapid Prototyping*: Researchers use Colab for quick prototyping of algorithms, running experiments, and testing hypotheses in a cloud-based environment without worrying about local setup.
- *Scientific Computing*: Researchers in fields like physics, biology, and economics often use Colab for simulations, computations, and data analysis.

Natural Language Processing (NLP)

- *Text Analysis*: NLP tasks such as text classification, sentiment analysis, and language modeling are easily carried out on Colab, thanks to its integration with libraries like NLTK, SpaCy, and Hugging Face Transformers.
- *Pretrained Models*: Users can fine-tune pretrained models for NLP tasks, leveraging Colab's GPU support for efficient training.

Computer Vision

- *Image Classification and Detection**:*** Colab is commonly used for building and testing computer vision models, including object detection, image classification, and image segmentation, with libraries like OpenCV, Keras, and TensorFlow.
- *Pre-trained Models*: Users can access pre-trained models like those from TensorFlow Hub or PyTorch Hub to fine-tune models on custom datasets for image-related tasks.

AI and Deep Reinforcement Learning

- *Reinforcement Learning*: Researchers and developers use Colab for training reinforcement learning agents using frameworks like OpenAI Gym, TensorFlow Agents, or Stable Baselines.
- *AI Research*: Colab is a valuable resource for experimenting with cutting-edge AI techniques, such as generative adversarial networks (GANs), transformer models, and more.

Automated Machine Learning (AutoML)

- *AutoML Tasks*: Colab is useful for running AutoML experiments, such as training models automatically with tools like Google's AutoML or other open-source libraries like AutoKeras.
- *Hyperparameter Optimization**:*** Google Colab can be used for automating hyperparameter tuning using libraries like Optuna or Hyperopt.

Collaborative Development

- *Team Collaboration*: Colab is great for teams working on joint projects, as it allows users to share and collaborate on notebooks in real-time, providing a version-controlled environment.
- *Code Reviews**:*** Teams can share notebooks for code reviews, making it easier to spot bugs, suggest changes, and review research.

Cloud-based Applications and APIs

- *Deploying APIs**:*** Developers use Colab for building cloud-based applications or APIs, often in conjunction with cloud services like Google Cloud Platform (GCP) or other serverless frameworks.
- *Integration with Cloud Services*: Colab can easily connect to cloud storage solutions (e.g., Google Cloud Storage) for seamless data handling.

CHAPTER 3

ARTIFICIAL NEURAL NETWORK

Introduction 3.1

An Artificial Neural Network (ANN) is a computational model designed to simulate the way biological neural networks process information. It is a fundamental technique in the field of machine learning and is particularly useful for tasks involving complex pattern recognition, such as image classification, speech processing, and sentiment analysis.

Structure of an Artificial Neural Network**:**

*Neurons (Nodes)**:***

The core unit of an ANN is the neuron, also known as a node**.** Each neuron processes input data and generates an output. In biological systems, neurons are connected via synapses, and in an ANN, neurons are linked by weights that determine the strength or influence of one neuron's output on the next

*Layers**:***

ANNs are structured in layers, each containing a group of neurons.

- Input Layer: The input layer receives the initial data and passes it to subsequent layers. The input layer doesn't perform computations but serves as the starting point for data processing.
- Hidden Layers: These layers perform the primary computations. The data received from the input layer is processed here using weights and activation functions. A neural network can have multiple hidden layers, and the depth of the network is determined by the number of hidden layers (hence, the term deep learning for networks with many layers).
- Output Layer: The final layer generates the model's output. For binary classification, the output layer often contains a single neuron with a sigmoid activation function, while for multi-class classification, the output layer may contain multiple neurons.

*Weights and Biases**:***

- Weights: These are parameters that control the influence of one neuron on another. During training, the weights are adjusted to minimize errors in predictions.
- Biases: Each neuron has an associated bias term that allows the model to make adjustments independent of the input data. Biases help improve the flexibility and accuracy of the model.

Activation Functions:

After receiving inputs and applying weights, neurons use an activation function to decide whether to "fire" and pass information forward. Common activation functions include:

- Sigmoid: Outputs values between 0 and 1, commonly used for binary classification.
- ReLU (Rectified Linear Unit)**:** Produces the input value when positive and outputs zero when negative, widely used for hidden layers in deep networks.
- *Tanh*: Outputs values between -1 and 1, used in some hidden layers.

Forward Propagation:

The process of passing input data through the network, from the input layer to the output layer, is called forward propagation. In this step, neurons compute weighted sums of inputs and apply the activation function to generate the output.

Loss Function: The loss function measures the difference between the predicted output and the actual target. Common loss functions include:

- Mean Squared Error (MSE)**:** Used for regression tasks.
- Cross-Entropy Loss: Typically used for classification tasks, measuring the difference between predicted probabilities and actual class labels.

*Backpropagation**:** Backpropagation* is the method by which ANNs learn from their mistakes. After forward propagation, the network computes the loss and uses backpropagation to adjust the weights by propagating the error backward through the network. This is done using gradient descent or other optimization algorithms to minimize the loss and improve the model's accuracy.

*Training**:*** Training an ANN involves repeatedly adjusting the network's weights and biases based on the data. This process is carried out over several iterations (epochs), allowing the model to learn and improve over time.

Key Characteristics of ANNs:

- Learning from Data: ANNs can automatically learn from data without explicit programming, making them highly adaptable to a wide range of tasks.
- Non-linearity**:** By using activation functions, ANNs can model complex, non-linear relationships, which makes them more powerful than linear models.
- Scalability: ANNs can handle large datasets and can scale with increasing data, which is why they are often used in big data and deep learning applications.

Applications 3.2

Artificial Neural Networks (ANNs) have found widespread use across various industries due to their powerful capabilities in pattern recognition, prediction, and decision-making. Below are some of the key areas where ANNs are applied:

1. Image and Video Processing

- *Object Detection and Classification*: Convolutional Neural Networks (CNNs), a type of ANN, are the backbone of image recognition tasks. They are widely used in detecting objects in images or video streams, such as recognizing faces, cars, or pedestrians in real-time.
- *Image Segmentation*: ANNs are used to partition an image into segments to simplify its analysis. This is important in medical imaging (e.g., segmenting regions of interest in MRI or CT scans) or autonomous driving systems

2. Natural Language Processing (NLP)

- *Sentiment Analysis**:** ANNs*, particularly Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, are used to classify sentiment in text data, such as analyzing the sentiment of movie reviews or social media posts..
- *Speech Recognition*: ANNs, especially LSTMs and RNNs, are employed in systems that convert spoken language into written text. Examples include virtual assistants like Siri and Alexa.

3. Autonomous Systems

- *Self-Driving Cars*: ANNs, particularly CNNs and reinforcement learning models, are critical for the operation of autonomous vehicles. They are used for object detection, path planning, and decision-making in real-time based on sensor data (e.g., cameras, LIDAR, radar).
- *Robotics**:*** ANNs are applied in robots for tasks like navigation, manipulation, and interaction with humans. They allow robots to learn from their environment and adapt to new situations without needing explicit programming.

4. Healthcare and Medical Diagnostics

- *Medical Image Analysis*: ANNs, especially CNNs, are used to analyze medical images like X-rays, MRIs, and CT scans to detect diseases such as cancer, tumors, and other abnormalities.
- *Disease Prediction and Diagnosis*: ANNs are used in analyzing patient data (e.g., medical records, lab results) to predict conditions such as heart disease, diabetes, or the likelihood of a stroke.

5. Finance

- *Stock Market Prediction*: ANNs are used to analyze historical market data and predict stock prices or trends. By identifying patterns in financial data, these models can assist traders in making better investment decisions.

CHAPTER 4

PROJECT

Description 4.1

Sentiment analysis, also known as opinion mining, involves processing and analyzing textual data to determine its emotional tone. In this project, we utilize the IMDB Movie Review dataset-a collection of movie reviews labeled as positive or negative. The goal is to train a machine learning model that can accurately predict the sentiment of a review.

To achieve this, we employ an Artificial Neural Network (ANN), a powerful machine learning technique suited for complex data patterns. The ANN model will undergo hyperparameter optimization using Grid Search or Random Search, focusing on key hyperparameters, including the number of layers, learning rate, activation functions, and batch size. This optimization helps to identify the most effective architecture and settings for the model.

Dataset Description:
The IMDB Movie Review Dataset is a large collection of 50,000 movie reviews, each labeled with a sentiment (either positive or negative). This dataset is widely used in natural language processing (NLP) for tasks like sentiment analysis, where the goal is to classify the overall sentiment expressed in a text. Each review is a free-form text entry, and the sentiments are typically represented in binary form, with 1 for positive reviews and 0 for negative reviews.

diversity of language used in the reviews—ranging from colloquial expressions to formal critiques—provides a realistic challenge for models, helping improve their ability to generalize across different types of text.

Dataset Highlights:

Total Reviews: 50,000 movie reviews
Sentiment Labels: Binary (positive or negative)
Application: Ideal for sentiment analysis, text classification, and other NLP tasks

Technology Stack 4.2

Python**:** Programming language used for development.

TensorFlow**:** Deep learning framework used for building and training the hand gesture recognition model.

NumPy: Library for numerical computing, used for array manipulation.

Pandas: It is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data.

Matplotlib: Library for creating static, animated, and interactive visualizations in Python.

Seaborn: Seaborn is a Python data visualization library based on Matplotlib that provides a high-level interface for drawing attractive and informative statistical graphics.

Artificial Neural Network : A computing system inspired by biological neural networks, consisting of layers of interconnected nodes that learn complex patterns from data.

Random Search: A hyperparameter optimization method that randomly samples parameter combinations within specified ranges, often faster than Grid Search for large parameter spaces.

Grid Search : A hyperparameter optimization technique that exhaustively searches through a predefined grid of parameters to find the best combination for a model.

Google Colab : A cloud-based Jupyter notebook environment by Google that enables users to write and execute Python code in the browser, especially useful for machine learning and data science.

.

Algorithms Used 4.3

1. Support Vector Machine:

   A supervised learning algorithm that finds the optimal hyperplane to classify data points into distinct classes.

2. Decision Tree:

   A model that uses a tree like structure of decisions and their possible consequences to classify data points or make predictions.

3. Naïve Bayes:

   A probabilistic classifier based on Bayes' theorem, assuming feature independence, often used for text classification.

4. Logistic Regression :

   A statistical model used for binary classification tasks, predicting the probability of an outcome by fitting data to a logistic curve.

Outcomes 4.4

The project's end product will be a robust ANN model capable of accurately classifying the sentiment of movie reviews in the IMDB dataset. Through hyperparameter tuning, we aim to maximize the model's performance across key metrics, demonstrating its effectiveness in sentiment analysis tasks. The project will provide insights into the impact of various hyperparameters on ANN models for text classification and highlight best practices for sentiment analysis using neural networks.

```
pd.DataFrame.sparse.from_spmatrix(train_x_vector,
                                  index=train_x.index,
                                  columns=tfidf.get_feature_names_out())
```

	00	000	007	01pm	02	04	08	10	100	1000	...	zooming	zooms	zues	zzzzzzzzzzzzzzzzz	æon	élan	émigré	ísnt	in	ünfaithful
144	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1839	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
16588	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
742	0	0	0	0	0	0	0	0.042791	0	0	...	0	0	0	0	0	0	0	0	0	0
9464	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
10644	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
16133	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1739	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2443	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1467	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

1340 rows × 20625 columns

GridSearchCV

This is technique consists of an exhaustive search on specified parameters in order to obtain the optimum values of hyperparameters.

```
[30] from sklearn.model_selection import GridSearchCV
     #set the parameters
     params = {'C': [1,4,8,16,32], 'kernel' : ['linear','rbf']}
     svc = SVC()
     svc_grid = GridSearchCV(svc,params, cv = 5)
     svc_grid.fit(train_x_vector, train_y)
```

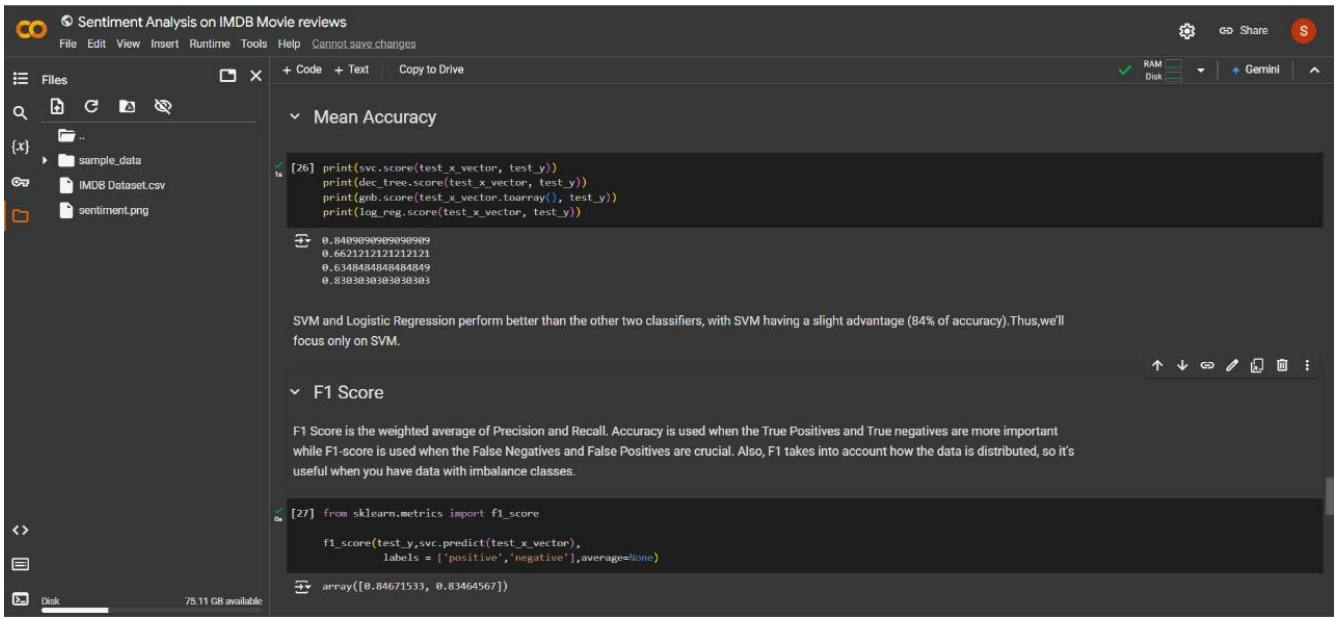
GridSearchCV

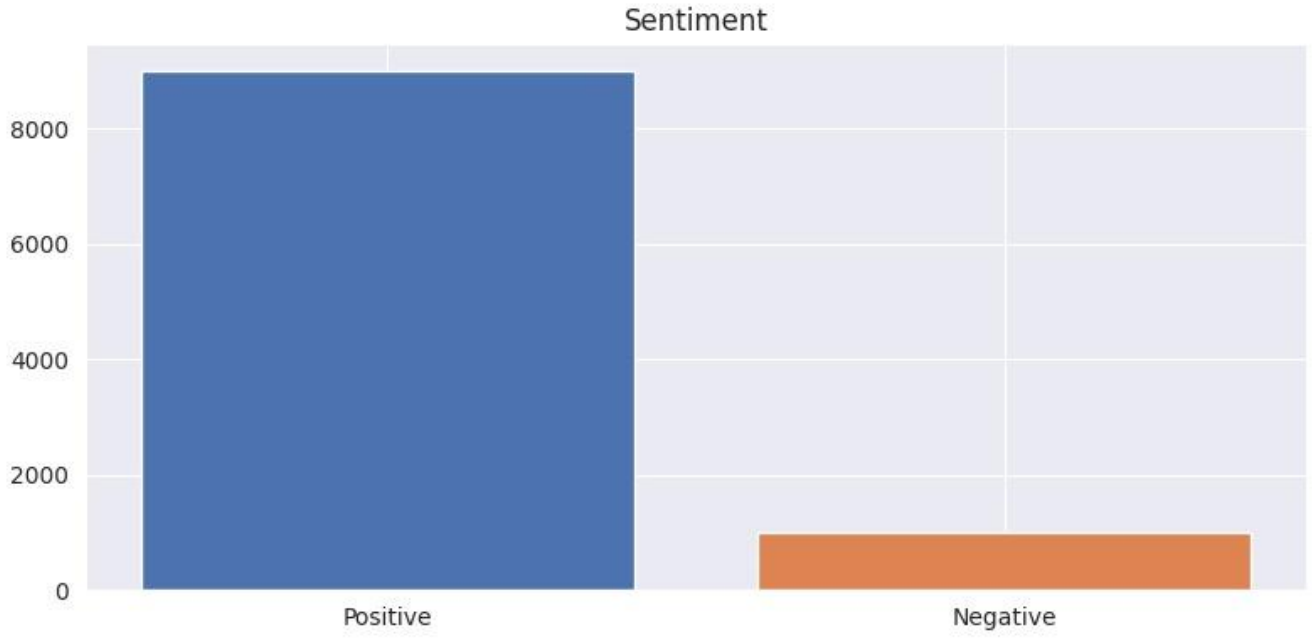
best_estimator_: SVC

SVC

```
[31] print(svc_grid.best_params_)
     print(svc_grid.best_estimator_)
```

```
{'C': 1, 'kernel': 'linear'}
SVC(C=1, kernel='linear')
```


Sentiment Analysis on IMDB Movie reviews
File Edit View Insert Runtime Tools Help Cannot save changes
Share
Files
+ Code + Text Copy to Drive
Gemini
sample_data
IMDB Dataset.csv
sentiment.png
Disk 75.11 GB available
Mean Accuracy
[26] print(svc.score(test_x_vector, test_y))
print(dec_tree.score(test_x_vector, test_y))
print(gnb.score(test_x_vector.toarray(), test_y))
print(log_reg.score(test_x_vector, test_y))
0.8409090909090909
0.6621212121212121
0.6348484848484849
0.8303030303030303
SVM and Logistic Regression perform better than the other two classifiers, with SVM having a slight advantage (84% of accuracy).Thus,we'll focus only on SVM.
F1 Score
F1 Score is the weighted average of Precision and Recall. Accuracy is used when the True Positives and True negatives are more important while F1-score is used when the False Negatives and False Positives are crucial. Also, F1 takes into account how the data is distributed, so it's useful when you have data with imbalance classes.
[27] from sklearn.metrics import f1_score
f1_score(test_y,svc.predict(test_x_vector),
labels = ['positive','negative'],average=None)
array([0.84671533, 0.83464567])


Sentiment
8000
6000
4000
2000
0
Positive
Negative

CHAPTER 5

CONCLUSION

In this assignment, we developed a sentiment analysis model using an Artificial Neural Network (ANN) on the IMDB Movie Review dataset. After thoroughly preprocessing the text data-tokenizing and padding reviews to a uniform input length-the model was prepared to recognize sentiment patterns. To enhance its performance, we employed hyperparameter tuning with Grid Search (or Random Search), adjusting parameters like the number of layers, learning rate, activation functions, and batch size. This optimization process enabled us to identify an effective configuration, significantly improving the model's accuracy and ability to generalize to unseen data.

Evaluating the model using accuracy, F1-score, and ROC-AUC metrics, we observed that the ANN achieved a strong balance of precision and recall, effectively distinguishing between positive and negative reviews. This project highlights the importance of data preprocessing, systematic model tuning, and rigorous evaluation, demonstrating that deep learning techniques, when carefully optimized, are highly effective for sentiment analysis tasks in natural language processing.

REFRENCES

IMDB dataset of 50K movie reviews. (2019, March 9). Kaggle. https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews?resource=download

Sentiment analysis in Python - Javatpoint. (n.d.). www.javatpoint.com. https://www.javatpoint.com/sentiment-analysis-in-python

GeeksforGeeks. (2024, September 4). Learning model building in Scikitlearn. GeeksforGeeks. https://www.geeksforgeeks.org/learning-model-building-scikit-learn-python-machine-learning-library/

Google Scholar. (n.d.). https://scholar.google.com/