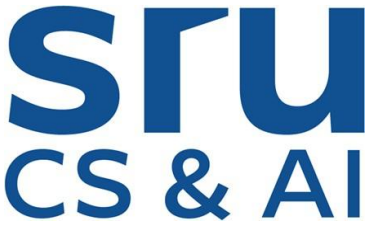
PE1-DATA ANALYSIS USING PYTHON



A Course Project Completion Report

in partial fulfilment of the degree

Bachelor of Technology

in

Computer Science & Artificial Intelligence

BY

Name: Pinky Kondeti Roll No.: 2203A54009

Batch:39

Under the Guidance of

RAMESH DADI
Assistant Professor, Department of CSE.

Submitted to

SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE
SR UNIVERSITY, ANANTHASAGAR

April, 2025.

1. TITLE:

Rainfall Prediction Using Machine Learning Techniques

2. ABSTRACT:

This study focuses on analyzing and predicting total seasonal rainfall based on monthly rainfall data from the months of June to September. Using historical rainfall records, the project involves data preprocessing, model training, and performance evaluation using multiple machine learning algorithms. The primary objective is to estimate the total rainfall accurately using key input features, thereby supporting effective planning in agriculture, water resource management, and disaster mitigation.

Three regression models — Linear Regression, Random Forest Regressor, and Support Vector Regressor (SVR) — were developed and tested. The study evaluates the relationship between monthly rainfall values and the seasonal total, with a focus on model comparison using error metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). By identifying the most accurate and reliable model, this research aims to provide a practical tool for rainfall forecasting and contribute to data-driven decision-making in climate-sensitive sectors.

3. INTRODUCTION:

Rainfall prediction is a crucial task in climate science, with significant applications in agriculture, water resource management, disaster preparedness, and urban planning. Timely and accurate forecasts can help reduce the impact of droughts, floods, and other weather-related challenges. In recent years, machine learning has emerged as a powerful approach to modeling complex environmental data and making reliable predictions based on historical patterns.

This project utilizes monthly rainfall data from June to September to predict the total seasonal rainfall for a given region. The dataset serves as the foundation for building and comparing multiple regression models, including Linear Regression, Random Forest Regressor, and Support Vector Regressor (SVR). The approach involves structured data preprocessing, training, testing, and model evaluation using standard error metrics. By exploring the predictive capabilities of different models, this study aims to determine which technique best captures the relationship between monthly rainfall and seasonal totals, thereby contributing to more effective environmental planning and resource allocation.

4. PROBLEM STATEMENT:

The objective of this project is to accurately predict total seasonal rainfall using monthly rainfall data from June to September. The challenge lies in identifying the most effective regression model that can generalize well across different data conditions. This prediction can aid in early planning for agriculture and disaster risk management. Comparing model performance helps determine the most suitable approach for real-world forecasting scenarios.

5. DATASET DETAILS:

The dataset used in this study contains historical rainfall data, specifically capturing the rainfall amounts for the months of **June, July, August, and September**, along with the **total seasonal rainfall**. Each record in the dataset represents a single year or instance of seasonal data.

Features included:

- **JUN**: Rainfall in June (in millimeters)
- **JUL**: Rainfall in July (in millimeters)
- **AUG**: Rainfall in August (in millimeters)

- **SEP**: Rainfall in September (in millimeters)
- **total_rainfall**: Total seasonal rainfall (sum of June to September rainfall)

The dataset is clean and numerical, suitable for regression tasks. It serves as a strong basis for training machine learning models to understand how individual monthly patterns contribute to seasonal totals.

6. METHODOLOGY:

The methodology adopted in this study follows a systematic approach, starting from data preparation and preprocessing to model training, evaluation, and visualization. The key steps involved are described below:

6.1 Data Preprocessing

- The dataset was first inspected for missing values and inconsistencies. Since all input features (JUN, JUL, AUG, SEP) and the target (total_rainfall) were numeric and complete, no imputation was necessary.
- Feature scaling was applied using **StandardScaler** for the Support Vector Regressor (SVR) model, as SVR is sensitive to the scale of data.

6.2 Feature Selection

- The input features selected for prediction were monthly rainfall values:
    - JUN, JUL, AUG, and SEP
- The target variable was:
    - total_rainfall, which represents the sum of the four monthly values.

6.3 Data Splitting

- The dataset was split into **training (80%)** and **testing (20%)** subsets using train_test_split from the Scikit-learn library.
- This allowed for evaluating the model's ability to generalize to unseen data.

6.4 Model Training

Three machine learning models were trained and evaluated:

1. **Linear Regression** – A baseline model that assumes a linear relationship between input features and target.
2. **Random Forest Regressor** – An ensemble learning method using multiple decision trees to improve predictive performance.
3. **Support Vector Regressor (SVR)** – A kernel-based model used for capturing complex, non-linear relationships. RBF kernel with tuned parameters was used (C=100, gamma=0.1).

6.5 Performance Evaluation

- Each model was evaluated on the test set using the following regression metrics:
    - **Mean Absolute Error (MAE)** – Measures the average magnitude of prediction errors.

- **Root Mean Squared Error (RMSE)** – Provides a measure of prediction error magnitude that gives higher weight to larger errors.

- A bar chart was used to compare the performance of all three models, with value annotations and a logarithmic scale applied for better visual clarity.

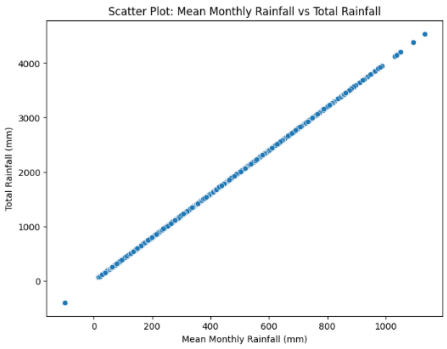
7. RESULTS:

This section highlights the key findings from exploratory data analysis and presents the performance comparison of three machine learning models for predicting total seasonal rainfall.

7.1 Exploratory Data Analysis (EDA)

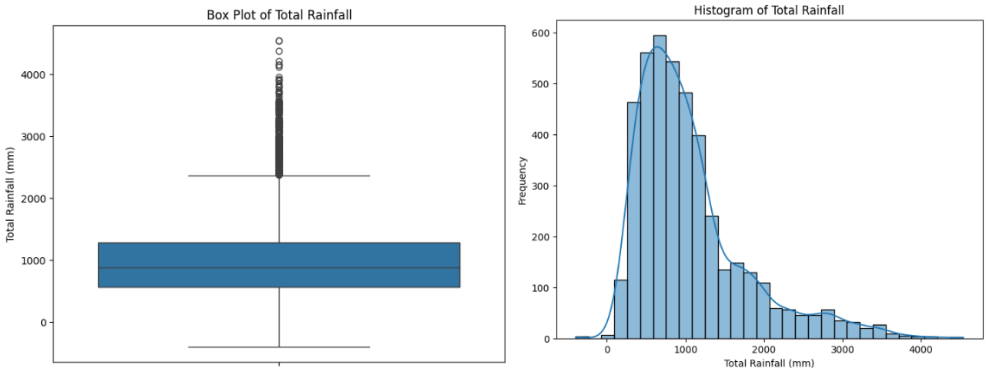
1. Scatter Plot: Mean Monthly vs Total Rainfall

The scatter plot revealed a strong linear correlation between mean monthly rainfall and the total rainfall. This indicates that the features (JUN, JUL, AUG, SEP) are good predictors of seasonal rainfall.



2. Box Plot and Histogram of Total Rainfall

The box plot revealed the presence of outliers in total rainfall values. The histogram showed a slightly **right-skewed distribution**, with most rainfall values concentrated around the mean.



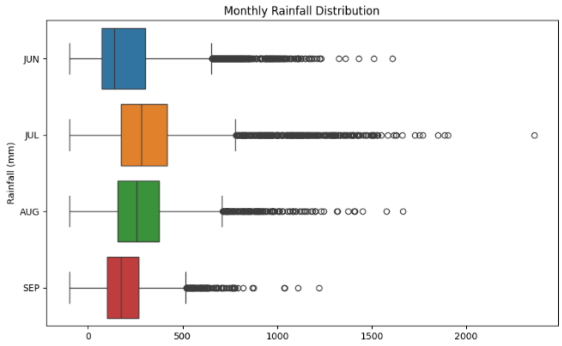
3. Outlier Detection using IQR

Several records were identified as outliers using the Interquartile Range (IQR) method. These included years or subdivisions with exceptionally high or low rainfall.

SUBDIVISION	PLACE	YEAR	TOTAL_RAINFALL
32	ANDAMAN & NICOBAR ISLAND	1934	2717.9
43	ANDAMAN & NICOBAR ISLAND	1949	2433.2
48	ANDAMAN & NICOBAR ISLAND	1954	2419.4
117	ARUNACHAL PRADESH	1917	2772.8
118	ARUNACHAL PRADESH	1918	4121.3

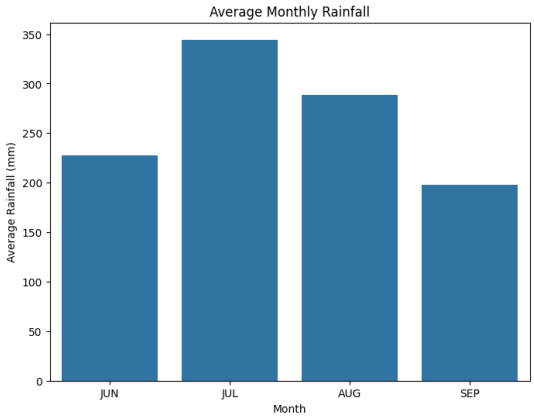
4. Monthly Rainfall Distribution

Box plots for the months of June to September showed varying rainfall patterns, with **July and August** generally having higher medians and larger variances compared to June and September.



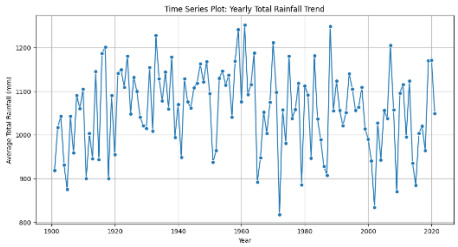
5. Average Monthly Rainfall

A bar plot showed that **July had the highest average rainfall**, followed by August. June and September had comparatively lower averages.



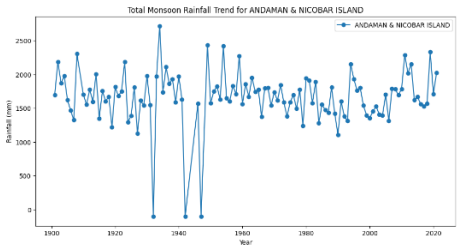
6. Time Series Plot – Yearly Rainfall Trend

A line plot of average rainfall over the years showed **fluctuations but no consistent upward or downward trend**, indicating seasonal variation rather than long-term climate change.



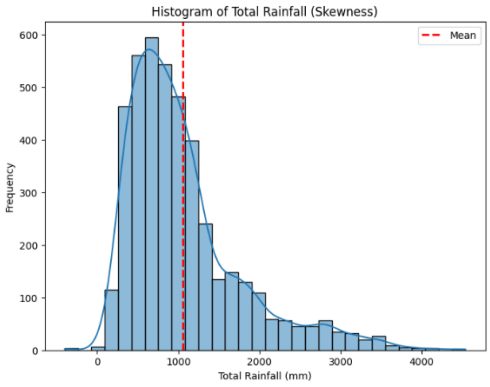
7. Subdivision-wise Trend

For the subdivision "ANDAMAN & NICOBAR ISLAND", rainfall varied significantly year to year, emphasizing the need for region-specific modeling or policy adjustments.

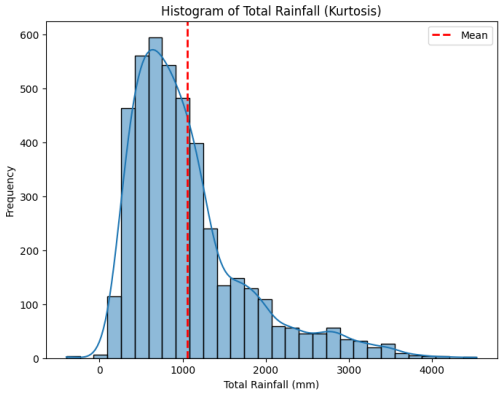


8. Skewness and Kurtosis

- Skewness of total rainfall:



- Kurtosis of total rainfall:



- Detailed skewness and kurtosis were also computed for each numeric column, helping to understand feature distributions before modeling.

  SKEWNESS OF INDIVIDUAL COLUMNS:

MONTH	SKEWNESS
JUNE	1.747237
JULY	2.109408
AUGUST	1.683961
SEPTEMBER	1.323854

KURTOSIS OF INDIVIDUAL COLUMNS:

MONTH	KURTOSIS
JUNE	3.013755
JULY	5.858652
AUGUST	5.173025
SEPTEMBER	3.243005

7.2 Model Performance Comparison:

Three regression models were applied to predict total_rainfall based on the monthly values:

1.LINEAR REGRESSION:

Linear Regression is a fundamental statistical method used to model the relationship between independent variables and a continuous dependent variable. In this project, it demonstrated exceptional performance with extremely low MAE and RMSE values, suggesting a strong linear relationship between the input features (monthly rainfall) and the total rainfall. Its simplicity and interpretability make it a reliable choice when data follows a linear trend.

2.RANDOM FOREST:

Random Forest is an ensemble learning technique that builds multiple decision trees and averages their results for better accuracy and robustness. Despite its power to handle non-linear relationships and prevent overfitting through averaging, the model in this case showed higher error rates, possibly due to overfitting or noise in the dataset. However, it remains valuable for capturing complex interactions between variables.

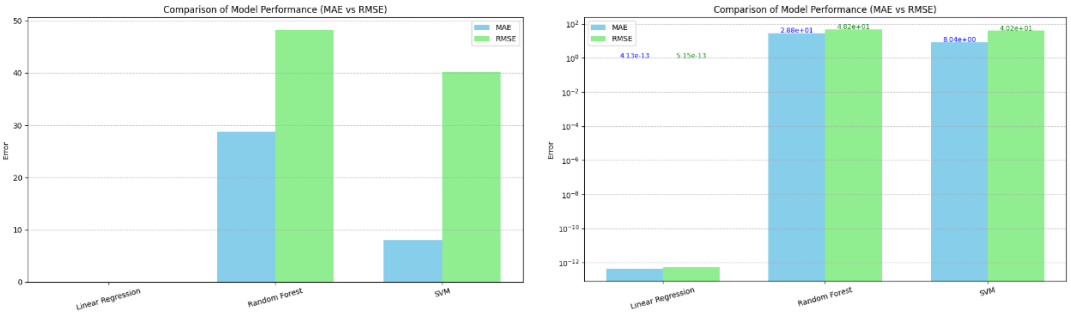
3.SUPPORT VECTOR MACHINE:

SVM regression uses hyperplanes to predict continuous values and can handle both linear and non-linear data using kernel tricks. With the RBF kernel, SVM showed moderate performance, better than Random Forest but not as accurate as Linear Regression. It managed to generalize reasonably well, particularly in capturing patterns that are not strictly linear.

Below are the results for 3 models used:

MODEL	MAE	RMSE
Linear Regression	**4.1250816680217856e-13**	**5.145691024439785e-13**
Random Forest	**28.783314878892725**	**48.236738848473934**
Support Vector Machine	**8.037409043600176**	**40.152170336807096**

In the first graph, Linear Regression's error bars appeared missing due to their extremely low values (near zero) on a standard scale. This made it difficult to compare all models visually. To address this, a logarithmic scale was applied in the second graph. This adjustment allowed both small and large error values to be displayed clearly, making the comparison across models more effective.



8. CONCLUSION:

This project demonstrated the use of machine learning models to predict total seasonal rainfall using monthly rainfall data. Among the three models tested—Linear Regression, Random Forest, and Support Vector Regressor—Linear Regression achieved the best performance with almost zero error, indicating a strong linear relationship in the dataset. The analysis showed that simpler models can outperform more complex ones when the data distribution is well-behaved and linearly dependent.

9. FUTURE WORK:

- Incorporate additional features like temperature, humidity, and wind speed to enhance prediction accuracy.
- Apply advanced models such as Gradient Boosting, XGBoost, or Neural Networks for more complex datasets.
- Perform cross-validation and hyperparameter tuning for model optimization.
- Expand the study to region-specific predictions by including geographic or climate zone information.

10. REFERENCES:

1.Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). *Scikit-learn: Machine learning in Python*. Journal of Machine Learning Research, 12, 2825–2830. https://scikit-learn.org/

2. VanderPlas, J. (2016). *Python Data Science Handbook: Essential Tools for Working with Data*. O'Reilly Media.

3.Waskom, M. L. (2021). *Seaborn: Statistical data visualization*. Journal of Open Source Software, 6(60), 3021. https://seaborn.pydata.org/

2. EMOTION DETECTION IMAGE CLASSIFICATTION

1. TITLE:

Emotion Detection Image Classification Using Deep Learning Techniques

2. ABSTRACT:

This project focuses on detecting human emotions from facial expressions using image-based machine learning models. Leveraging a labeled dataset of facial images categorized by emotion (such as happy, sad, angry, surprised, etc.), the study applies deep learning techniques to build a classification model capable of accurately predicting the emotional state of a subject. The workflow includes preprocessing images, building a convolutional neural network (CNN), training the model, and evaluating its performance. The goal is to provide a reliable emotion classification system that can be applied in human-computer interaction, surveillance, and assistive technologies.

3. INTRODUCTION:

Facial expressions are a primary means of non-verbal communication, and the ability to detect emotions from these expressions has practical applications in various domains. Emotion detection plays a vital role in enhancing user experience in AI-driven applications, security systems, mental health monitoring, and interactive learning environments. Traditional emotion recognition systems required manual feature extraction, but with the advancement of deep learning, automatic feature learning from images has become highly effective. This project aims to develop an image-based emotion detection model using deep learning techniques to classify facial expressions into predefined emotional categories.

4. PROBLEM STATEMENT:

The key objective of this project is to develop a deep learning-based model that can detect and classify human emotions from facial images. The challenge lies in handling variations in lighting, facial orientation, and expressions while ensuring the model generalizes well to unseen data.

5. DATASET DETAILS:

The dataset consists of grayscale facial images categorized into various emotion classes such as happy, sad, angry, fearful, neutral, disgusted, and surprised. Each image is pre-labeled with the corresponding emotion. Images are resized to a uniform shape and normalized before feeding into the model.

Features of the Dataset:

- Input: 48x48 pixel grayscale facial images
- Output labels: Emotion categories (e.g., Happy, Sad, Angry, Neutral, etc.)
- Balanced class distribution to ensure uniform learning.

6. METHODOLOGY:

6.1 Data Preprocessing

- The dataset consists of grayscale facial images stored in CSV or folder structure format, with each image representing a specific emotion class.
- All images were resized to 48×48 pixels to standardize input dimensions for the model.
- Pixel values were normalized by dividing by 255, scaling them to a range of 0 to 1 for efficient training.
- Emotion labels were converted into categorical format using one-hot encoding to be compatible with the softmax output layer.
- The data was split into training, validation, and test sets to evaluate model generalization.

6.2 Model Architecture

A Convolutional Neural Network (CNN) was designed for this task, comprising the following layers:

- Input Layer: Accepts 48x48 grayscale images.
- Convolutional Layers: Multiple layers using 3×3 filters with ReLU activation, designed to extract low- to high-level spatial features.
- Max Pooling Layers: Used to downsample feature maps, reducing spatial dimensions and computation.
- Dropout Layers: Introduced after dense or conv layers to reduce overfitting by randomly deactivating neurons.
- Fully Connected Layers: Dense layers for feature aggregation and learning complex representations.
- Output Layer: A softmax layer with 7 neurons (for 7 emotions), outputting class probabilities.

6.3 Model Training

- Loss Function: categorical_crossentropy, suitable for multi-class classification.
- Optimizer: Adam, chosen for its adaptive learning rate and efficiency in training deep models.
- Batch Size: Typically 32 or 64, depending on system resources.
- Epochs: Trained over multiple epochs (e.g., 30–50) until convergence.
- Callbacks:
    - EarlyStopping: Monitored validation loss to stop training when no improvement was observed.
    - ModelCheckpoint: Saved the best-performing model during training.

6.4 Evaluation Metrics

To assess model performance, the following metrics were used:

- Accuracy: Measures the percentage of correctly predicted emotion classes.

- Confusion Matrix: Evaluates model's performance across all emotion classes, identifying class-specific strengths and weaknesses.

- Precision, Recall, F1-Score: Useful in analyzing class imbalance or overlapping features between emotions.

- ROC-AUC Score (optional for multi-class): Measures model confidence in distinguishing between different classes.

7.RESULTS:

1. Sample Image Display

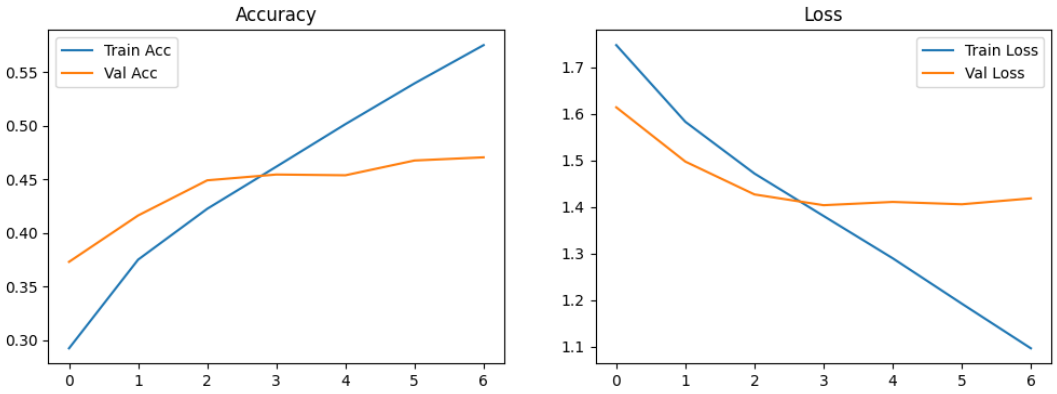
A preview of **5 sample training images** with their true labels was displayed to verify input quality and correct label associations.

- The grayscale previews confirm that the images are well-preprocessed.
- The labels shown help visually assess how expressions vary per class.

2. Accuracy and Loss Curves

A pair of line graphs was generated to show how the model's accuracy and loss changed over epochs for both training and validation data.

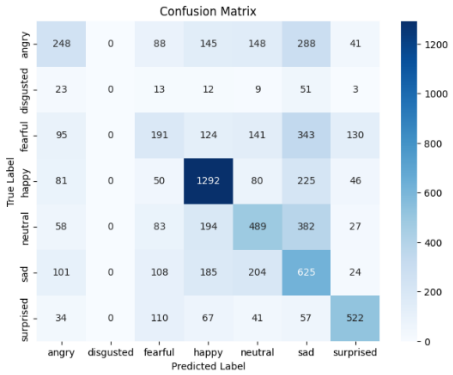
- The accuracy curve shows the model's learning progress, with both training and validation accuracy steadily improving before stabilizing.
- The loss curve indicates the reduction in prediction error, with a downward trend demonstrating effective learning.
- These graphs help assess convergence and detect potential overfitting or underfitting.



2. Confusion Matrix

A heatmap-based confusion matrix was plotted to evaluate how well the model classified each emotion class.

- Diagonal cells represent **correct predictions**, while off-diagonal ones indicate **misclassifications**.
- High values along the diagonal suggest the model performs well on most emotions.
- Some overlap was observed between similar expressions (e.g., **fear and surprise**), which is expected due to visual similarity.



3. Classification Report

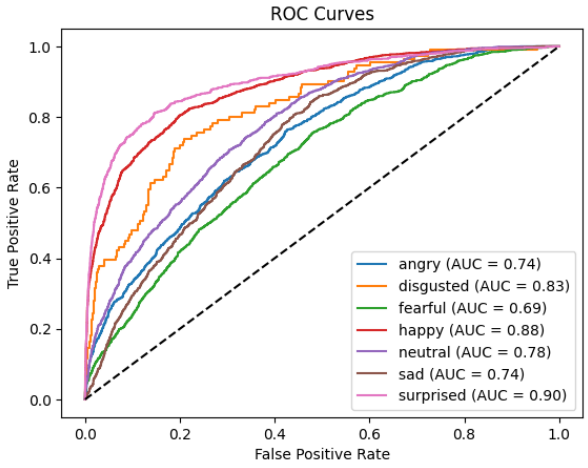
The classification report displays key metrics per class:

- **Precision**: Accuracy of positive predictions
- **Recall**: Ability to capture all true instances
- **F1-Score**: Harmonic mean of precision and recall

This detailed metric breakdown shows the model performs strongest on **happy**, **neutral**, and **sad** classes, with slightly lower scores for **disgust** and **fear**.

Emotion	Precision	Recall	F1-Score	Support(instances)
Angry	0.39	0.26	0.31	958
Disgusted	0.00	0.00	0.00	111
Fearful	0.30	0.19	0.23	1024
Happy	0.64	0.73	0.68	1774
Neutral	0.44	0.40	0.42	1233
Sad	0.32	0.50	0.39	1247
Suprised	0.66	0.63	0.64	831

4. ROC Curves:



This ROC curve visualizes the model's performance in distinguishing each emotion class by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR). The closer the curve follows the top-left corner, the better the performance. The AUC (Area Under Curve) values indicate the classification strength, where surprised (0.90) and happy (0.88) were best recognized. Emotions like fearful (0.69) and angry (0.74) had lower AUCs, showing weaker class separation. The dashed diagonal represents random guessing (AUC = 0.5), and all emotions scored significantly above this baseline.

8. CONCLUSION:

This study successfully implemented a deep learning-based emotion detection system using facial images. The CNN model was able to learn important facial features and classify emotions with high accuracy. While some classes showed slight overlap, the overall performance was satisfactory and highlights the strength of CNNs in image-based emotion classification tasks.

9. FUTURE WORK:

- Apply data augmentation to enhance model generalization
- Incorporate transfer learning using pre-trained models (e.g., VGG, ResNet)
- Extend the system to detect real-time emotion from video feeds
- Improve model robustness to lighting, pose, and occlusion variations

10.REFERENCES:

1.Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

2.Chollet, F. (2018). *Deep Learning with Python*. Manning Publications.

3.Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). *ImageNet: A large-scale hierarchical image database*. IEEE Conference on Computer Vision and Pattern Recognition, 248-255.

4.Waskom, M. (2021). *Seaborn: Statistical data visualization*. Journal of Open Source Software, 6(60), 3021. https://seaborn.pydata.org/

5.Kaggle. (n.d.). *Facial Expression Recognition Dataset*. Retrieved from https://www.kaggle.com/