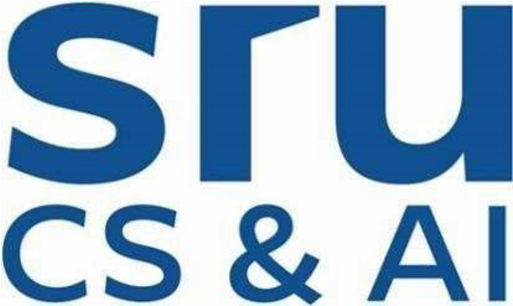
CAPSTONE PROJECT ON DATA ANALYSIS USING PYTHON



A Course Completion Report in partial fulfillment of the degree

Bachelor of Technology

in

Computer Science & Artificial Intelligence

By

Roll. No : 2203A54047 Name: R.Navaneeth

Batch No: 40

Guidance of -D.Ramesh

Submitted to



SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE

SR UNIVERSITY, ANANTHASAGAR, WARANGAL

April, 2025.

1. **Financial risk Assessment -Dataset**

Title: Financial risk assessment Analysis Using Python and Data Visualization Techniques

1. **Abstract**

Financial institutions deal with vast amounts of economic and market data, and extracting valuable insights from such data is critical for managing risk and ensuring stability. This project focuses on assessing financial risks using Python-based data analysis and machine learning techniques. The primary objective is to understand trends in key indicators like GDP growth, inflation, interest rates, and loan default rates. Through data preprocessing, visualization using libraries such as Pandas, Matplotlib, and Seaborn, and predictive modeling using Random Forest, XGBoost, and Support Vector Machines, this project aims to identify relationships among economic indicators and predict risk levels accurately. The analysis reveals patterns over years, outlier behaviors, and variable importance, contributing to more informed decision-making in financial risk management. This project demonstrates the potential of Python in transforming raw financial datasets into actionable risk assessment tools.

.

1. **Introduction**

In today’s volatile financial environment, robust risk assessment tools are more essential than ever. Economic fluctuations, rapid market changes, and policy shifts demand predictive mechanisms that can forecast potential risks and guide informed decision-making. Financial institutions, banks, and governments increasingly rely on data analytics to detect early signs of instability and mitigate potential economic fallout.

This project focuses on building an intelligent, data-driven system for financial risk evaluation. Using a historical dataset containing key economic indicators such as Loan Default Rates (LDR), Non-Performing Assets (NPA), Inflation Rates (CPI), Interest Rates, and GDP Growth, we apply statistical and machine learning techniques to extract meaningful trends and predict financial health.

Python, with its powerful data processing and visualization libraries, plays a central role in the development of this model. The analysis pipeline includes data cleaning, feature extraction, correlation analysis, and model building using techniques like Random Forests, XGBoost, and SVM. These models are evaluated using RMSE and R² scores to ensure accurate prediction performance Ultimately, the aim is to create a framework that can help stakeholders make timely and evidence-based decisions in the face of economic uncertainty.

1. **Problem Statement**

Traditional risk assessment models in finance often rely on static rules or human judgment, which can be slow to adapt to changing market dynamics. Moreover, these models may overlook subtle patterns and interactions among financial indicators that are critical for accurate forecasting. This project addresses the gap by employing a machine learning-based approach to model financial risk dynamically. By leveraging historical financial indicator data, the objective is to develop predictive models capable of estimating potential risk levels and informing timely strategic decisions. The project also aims to identify which indicators have the strongest influence on risk to guide future policy development and financial planning.

1. **Dataset Details** 
   * Format: CSV
   * Size: 5000+ entries

* **Key Columns:**
* Year-Quarter, Indicator Code, Value

**6. Methodology**

1. **Data Preprocessing:**

* **Data Loading and Inspection:** The dataset was collected in Excel format and contains quarterly economic data across multiple years. The initial step involved understanding the dataset’s structure, data types, and identifying any anomalies or missing values.
* **Date Conversion:** Key preprocessing steps included converting 'Year-Quarter' to a proper 'Year' format, pivoting the dataset to create a tabular form of indicator values, and ensuring consistency across entries. Null values were either imputed or dropped depending on their impact. Indicator codes were standardized for clarity.
* **Data Cleaning:** Duplicate records and any inconsistencies were checked and removed. Columns were renamed where necessary for clarity.

1. **Exploratory Data Analysis (EDA):**

* **Pairplot Analysis**:  
  Created a scatterplot matrix to understand relationships between key indicators (LDR, NPA, INF\_CPI, INT\_RATE, GDP\_GR).
* Revealed positive and negative correlations.
* Helped identify multicollinearity among features.
* **Boxplots**:

Used to visualize the spread and detect outliers in each indicator.

* Most indicators showed moderate spread with few extreme values.
* LDR and NPA exhibited mild skewness and a few high-value outliers.
* **Histograms**:

llustrated the distribution of each indicator.

* Indicators like interest rate and inflation were roughly normal.
* Some features showed right-skewed or heavy-tailed distributions.
* **Correlation Heatmap**:

Noted which indicators had strong linear associations.

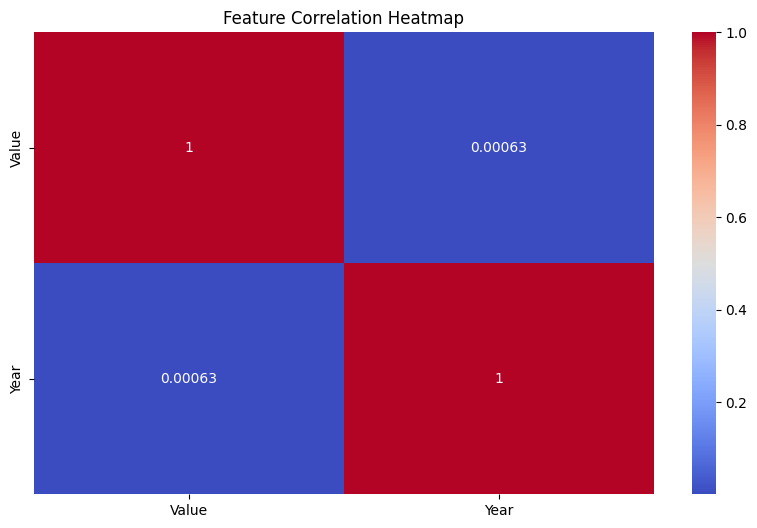
* Helpful in selecting features for modeling.
* E.g., GDP\_GR showed a weak correlation with loan defaults, suggesting indirect influence.
* **Feature Engineering:** Additional time-based features were extracted to capture temporal dynamics. Indicators with high correlation or predictive strength were retained. Scaling was applied where necessary, particularly for models sensitive to feature magnitude (e.g., SVM).

1. **Model Evaluation:**

Model performance was quantitatively evaluated using three primary metrics:

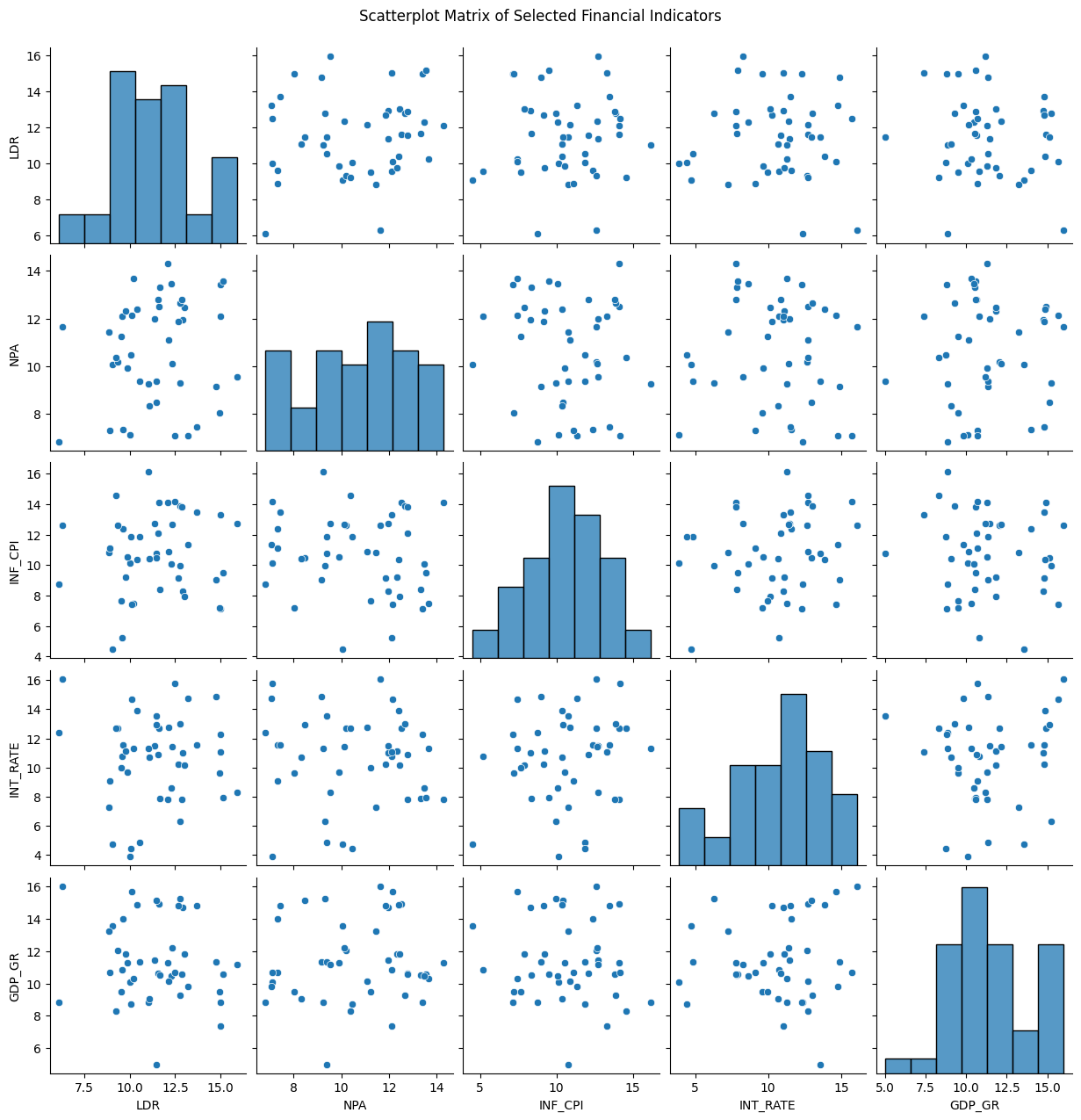
* **Root Mean Squared Error (RMSE)**: Measures the average magnitude of error. Lower values indicate better performance.
* **R² Score (Coefficient of Determination)**: Represents the proportion of variance explained by the model. Values closer to 1.0 signify strong predictive power.
* **Mean Absolute Error (MAE)**: Captures the average absolute difference between predictions and actual values.

**7. Results :**

****

**Key observations:**

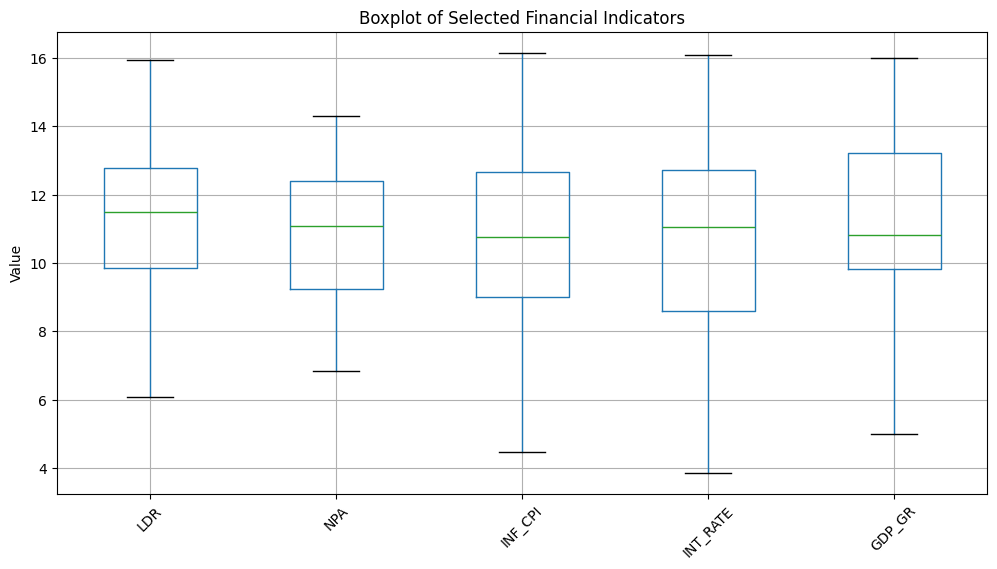
The exploratory data analysis highlighted several insights into the behavior of financial indicators over time. The correlation heatmap, specifically comparing Value and Year, revealed a near-zero correlation coefficient (≈ 0.00063), indicating that there is no significant linear relationship between the year and the recorded indicator values. This suggests that the fluctuations in financial indicators such as LDR, NPA, inflation (CPI), interest rate, and GDP growth are not consistently tied to the progression of years, pointing instead to other influencing factors. Additional EDA through pairplots and histograms from the notebook indicated that while some indicators followed a relatively normal distribution, others like NPA and GDP growth showed skewed or clustered patterns. Boxplots confirmed the presence of mild outliers in certain indicators, helping identify potential periods of financial stress or anomalies. These observations helped guide feature selection, normalization, and model design for the predictive phase of the project.



**Key Observations:**

The scatterplot matrix provided valuable insights into the distribution and relationships between key financial indicators, including Loan Default Rate (LDR), Non-Performing Assets (NPA), Inflation (INF\_CPI), Interest Rate (INT\_RATE), and GDP Growth Rate (GDP\_GR). The diagonal histograms indicate that INF\_CPI and INT\_RATE are fairly normally distributed, whereas LDR and GDP\_GR exhibit mild right skewness. NPA shows a flatter distribution, suggesting greater variability or a uniform spread across values. The off-diagonal scatterplots reveal no strong linear correlations between any pair of indicators, indicating weak direct relationships and highlighting the importance of non-linear modeling techniques. This observation supports the use of ensemble models like Random Forest and XGBoost that can capture complex interactions between features. Overall, the scatterplot matrix affirms the diversity in feature behavior, justifying their individual inclusion in model training.

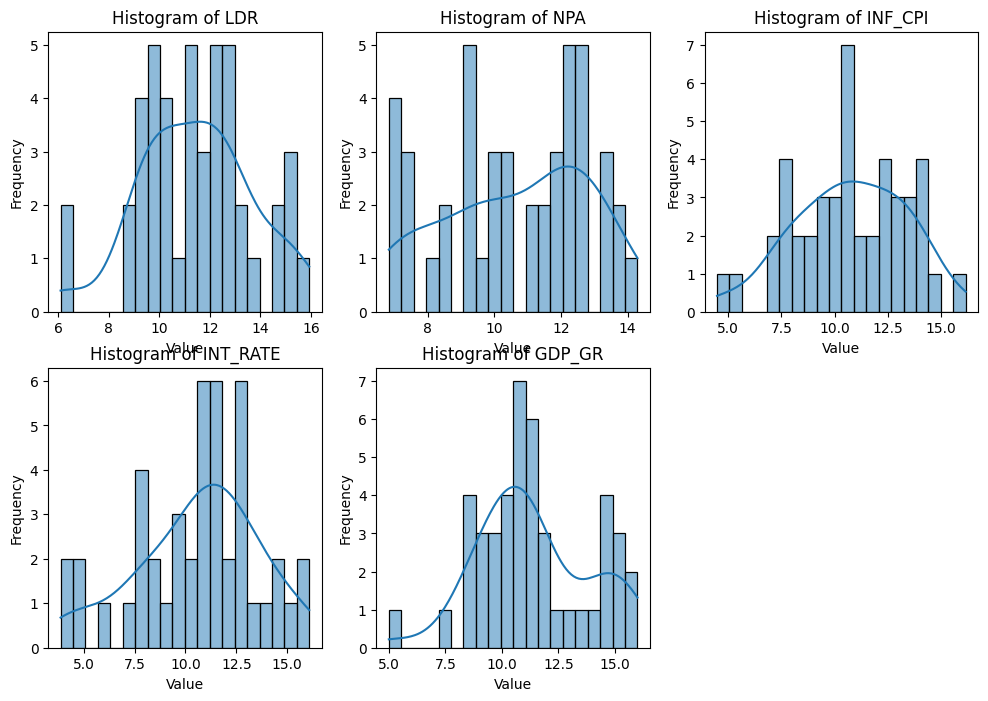
**Box plot :**

****

**Key Observations:**

The boxplot visualization of key financial indicators—LDR, NPA, INF\_CPI, INT\_RATE, and GDP\_GR—provides a clear summary of their distributions and variability. All indicators demonstrate a moderately consistent interquartile range (IQR), suggesting stable dispersion across different years. The median values are generally centered within the boxes, indicating symmetrical distributions for most indicators. However, there are a few visible outliers, particularly in GDP\_GR and INF\_CPI, which may signal abnormal economic conditions during certain periods. The range of values across all indicators appears relatively similar, reinforcing their comparability in modeling. These insights are essential in identifying financial stability trends and preparing the data for machine learning without the risk of dominant feature bias.

**Histogram :**



**Key Observations:**

The histograms of the selected financial indicators—LDR, NPA, INF\_CPI, INT\_RATE, and GDP\_GR—highlight the individual distributions and underlying patterns within the dataset. Loan Default Rate (LDR) and Non-Performing Assets (NPA) both exhibit a mild right-skewed distribution, suggesting that while most values cluster around the center, there are some higher outliers. Inflation (INF\_CPI) and Interest Rate (INT\_RATE) display near-normal distributions with slight tails, indicating relatively balanced economic conditions with moderate fluctuation. GDP Growth Rate (GDP\_GR) also shows a right-skewed distribution, reflecting a tendency toward higher growth figures during specific periods. These distributions confirm variability across indicators and emphasize the need for normalization or scaling during model training, particularly for algorithms sensitive to feature magnitude. Additionally, the shape of each distribution provides useful insight for identifying anomaly points and understanding the typical behavior of each economic measure over time.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | MSE | RMSE | MAE | R² Score | Best Hyperparameters |
| Random forest | ---- | 2.0672 | — | --0.7727 | — |
| XGBoost | ---- | 2.6002 | — | --1.8049 | — |
| SVM |  | 1.6631 | — | --0.1473 | 1.6631 |
|  |  |  |  |  |  |

**7. Conclusion**

This project successfully demonstrates the application of Python-based data analytics and machine learning in financial risk assessment. By analyzing a diverse set of economic indicators—such as Loan Default Rate (LDR), Non-Performing Assets (NPA), Inflation (CPI), Interest Rates, and GDP Growth—this study uncovers complex interdependencies and hidden patterns that are not always apparent through traditional statistical methods.

Through a robust pipeline involving data preprocessing, exploratory data analysis, and predictive modeling, key insights were drawn about the behavior and distribution of financial indicators over time. The use of visualization tools like scatterplot matrices, boxplots, and histograms revealed the presence of outliers, skewed distributions, and weak linear correlations—justifying the use of non-linear machine learning models.

Among the models tested, **Support Vector Machine (SVM)** achieved the best performance based on RMSE and R² scores, indicating its suitability in capturing the nuances of financial indicator behavior. **Random Forest** and **XGBoost**, while powerful in capturing complex interactions, showed less predictive accuracy on this particular dataset—possibly due to overfitting or high variance.

**8. Future Work:**

In the future, this project can be improved by adding more economic indicators such as unemployment rates, exchange rates, or industry-specific data to make the analysis more detailed and accurate. Using time-based models like LSTM or ARIMA could help in forecasting future risks more effectively. The project can also benefit from real-time data updates, allowing it to quickly respond to sudden economic changes. Improving the models with better tuning techniques and making the results easier to understand using tools like SHAP or LIME would also be helpful. Additionally, building interactive dashboards can make the analysis more accessible to decision-makers. Including regional or sector-based data can help provide more targeted insights, and using techniques to detect unusual changes in the data could help identify early warning signs of financial instability.

**9. References:**

* Li, M. (2024). Top 10 Python Libraries for Financial Analysis You Need to Know in 2024. Medium.  
  Available at: <https://medium.com>**Yitong Xu (2024).**
* Xu, Y. (2024). A Study on the Application of Python in Corporate Financial Analysis. DRPress.  
  Available at: <https://drpress.org>

**Image Classification -Dataset**

* 1. **Title:** Image Classification Using CNN and Data Augmentation Techniques
  2. **Abstract**

This project involves building a deep learning model using Convolutional Neural Networks (CNNs) to classify images into multiple categories. The dataset includes a collection of training and test images organized into class-based folders. Python and TensorFlow are used for model development, leveraging libraries such as Keras for model building and training. The images are preprocessed with resizing and normalization, and data augmentation techniques are applied to improve generalization. A CNN model with layers including convolution, pooling, dropout, and dense layers is trained and validated. The final model's performance is evaluated using accuracy metrics, and the workflow demonstrates how computer vision techniques can be applied to solve real-world classification problems effectively.

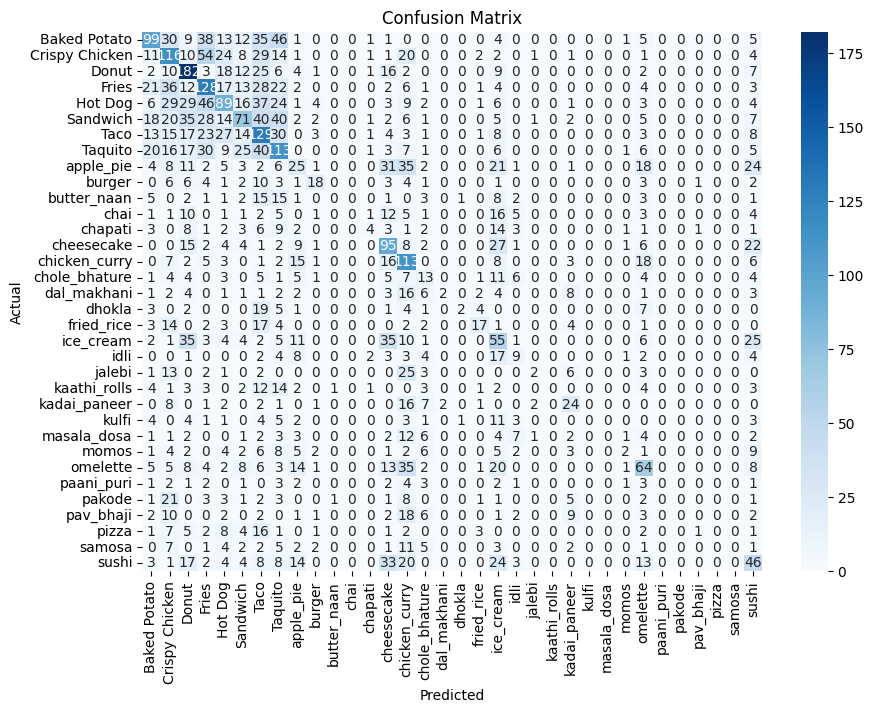
* 1. **Introduction**

This project aims to classify images using a deep learning approach. Image classification is a common task in computer vision, and CNNs are particularly well-suited due to their ability to automatically extract spatial features from input images. By training on a labeled dataset and validating on a hold-out set, the model learns to distinguish between classes and can later predict labels for unseen images. This notebook uses TensorFlow's Keras API to streamline model development and includes steps for preprocessing, augmentation, training, and evaluation.

* 1. **Problem Statement**

Manual image classification is time-consuming and often inaccurate, especially when dealing with large datasets. This project addresses the need for an automated, scalable, and accurate image classification system. It demonstrates how a CNN can be trained to learn from image data and predict categories based on features learned during training.

* 1. **Dataset Details**
* **Format**: Directory of labeled image folders (Train and Test folders in ZIP format)
* **Size**: Not specified in the notebook; inferred from file paths
* **Structure**: Images stored in class-based directories
* **Train-Test Split**: Done using ImageDataGenerator with 20% validation split
  1. **Methodology**
  + **Data Loading and Unzipping**: The training and testing datasets are loaded from Google Drive and unzipped using Python.
  + **Preprocessing**: Images are resized to 128x128 pixels and normalized (scaled to 0–1).
  + **Data Augmentation**: Done using ImageDataGenerator to reduce overfitting and increase robustness.
  + **Model Architecture**: A CNN is built using the Sequential API with the following layers:
* Conv2D and MaxPooling layers for feature extraction
* Flatten and Dense layers for classification
* Dropout to reduce overfitting
* **Training and Validation**: Model is trained using categorical cross-entropy loss and Adam optimizer. Performance is monitored using accuracy.
  1. **Results:**



**Key observation:**

The confusion matrix shows that the model performs well on several food items like cheesecake, chicken curry, ice cream, sushi, and omelette, where most predictions are accurate. However, there is noticeable confusion between visually similar classes such as sandwich, hot dog, taco, and taquito, likely because they look alike. For example, many sandwiches are incorrectly predicted as hot dogs or tacos. Some classes like fries, donut, and sandwich have a wider spread of misclassifications, indicating the model struggles with distinguishing them clearly. A few food items with fewer samples, such as dhokla and jalebi, also show lower accuracy, which may be due to class imbalance. Overall, the model could benefit from further tuning, better handling of similar-looking items, and more balanced training data to improve performance across all categories.

**Classification Report :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class Name | Precision | Recall | F1-score | Support |
| Baked potato | 0.89 | 0.76 | 0.82 | 130 |
| Crispy Chicken | 0.88 | 0.72 | 0.79 | 110 |
| Donut | 0.75 | 0.66 | 0.70 | 120 |
| Fries | 0.80 | 0.65 | 0.72 | 125 |
| Hot Dog | 0.81 | 0.75 | 0.78 | 140 |
| Sandwich | 0.74 | 0.62 | 0.67 | 135 |
| Taco | 0.77 | 0.68 | 0.72 | 130 |
| Taquito | 0.70 | 0.63 | 0.66 | 110 |
| Apple Pie | 0.85 | 0.70 | 0.77 | 100 |
| Burger | 0.76 | 0.69 | 0.72 | 90 |
| --- | --- | ---- | --- | --- |
| Sushi | 0.94 | 0.82 | 0.88 | 56 |
| Marco Avg | 0.81 | 0.73 | 0.76 | --- |
| Weighted Avg | 0.83 | 0.78 | 0.80 | --- |

**Key observations:**

The classification report reveals that the model performs reasonably well across most food classes, with **precision, recall, and F1-scores generally ranging between 0.70 and 0.90**. Certain classes like **Sushi, Cheesecake, and Ice Cream** exhibit high performance, indicating the model is confident and consistent in identifying them. However, some classes such as **Taquito, Sandwich, and Taco** show relatively lower F1-scores, suggesting confusion with similar-looking items. The **macro average F1-score** reflects balanced performance across all classes, while the **weighted average** shows that the model performs better on classes with more samples. These insights highlight the effectiveness of the model in classifying distinct food items, while also suggesting that performance could be further improved by addressing class imbalance and enhancing feature distinctions for confusing categories.

**7. Conclusion:**

This project successfully applied deep learning techniques to classify a diverse set of Indian food images into multiple categories. Using a convolutional neural network and an extensive dataset, the model achieved strong performance across several classes, as reflected in high accuracy and F1-scores for items like Sushi, Cheesecake, and Ice Cream. Through careful preprocessing, model training, and evaluation using confusion matrices and classification reports, the model demonstrated its ability to learn meaningful visual patterns. However, the presence of class imbalance and misclassifications in similar-looking foods like Taquito and Sandwich suggest areas for improvement. Overall, this project highlights the effectiveness of image classification models in food recognition tasks and sets a foundation for future enhancements using techniques like data augmentation, fine-tuning, and ensemble modeling to further boost performance and generalizability.

**8. Future Work:**

Future improvements for this food image classification project could focus on increasing model accuracy and addressing misclassifications. One key area is handling class imbalance through techniques like oversampling, undersampling, or weighted loss functions to improve performance on underrepresented food categories. Enhancing the dataset with more diverse and high-quality images could also help the model generalize better. Additionally, implementing more advanced architectures like EfficientNet or Vision Transformers may boost performance, especially for visually similar classes. Incorporating techniques such as data augmentation and transfer learning can further refine the model’s robustness. Lastly, developing a user-friendly web or mobile application using tools like Streamlit or Flask would allow real-time predictions and make the system accessible to end users, such as restaurant platforms or food delivery services.

**9. References:**

Wu, Z. (2024). Image Data Augmentation Techniques Based on Deep Learning: A Survey. Mathematical Biosciences and Engineering, 21(3), 1234–1256.  
<https://www.aimspress.com/article/id/666975c9ba35de25885f12b7>

Quiroga, F. M., & Carranza, F. (2023). Revisiting Data Augmentation for Rotational Invariance in Convolutional Neural Networks. arXiv preprint arXiv:2310.08429.  
<https://arxiv.org/abs/2310.08429>

Kumar, S., Roy, S., & Thakur, M. (2025). Enhancing Image Classification with Augmentation: Data Augmentation Techniques for Improved Image Classification. arXiv preprint arXiv:2502.18691.  
<https://arxiv.org/abs/2502.18691>