DATAANALYSIS USING PYTHON

A Course Completion Report in partial fulfillment of the degree

Bachelor of Technology

in

Computer Science & Artificial Intelligence

By

Roll. No :2203A54011 **Name**: M.Abhilash

Batch No: 39

Guidance of D. Ramesh

Submitted to

SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE
SR UNIVERSITY, ANANTHASAGAR, WARANGAL

April, 2025.

1.CREDIT CARD APPLICATION

TITLE

Credit Card Application Using Machine Learning Techniques on Text-Based Email Features

ABSTRACT

The approval of credit card applications is a key function for financial institutions, requiring careful assessment of an applicant's financial and personal background. Manual evaluation is time-consuming, prone to inconsistencies, and often lacks scalability. This project presents a machine learning-based approach to automate the credit card approval process by analyzing historical application data. Using a dataset containing both numerical and categorical features about applicants, we apply various preprocessing techniques, such as handling missing values, encoding categorical variables, and normalizing data. Multiple classification algorithms—including Logistic Regression, Decision Trees, and Random Forests—are evaluated to predict application approval outcomes.

INTRODUCTION

In today's fast-paced financial world, credit cards have become an essential tool for personal and commercial transactions. Financial institutions process thousands of credit card applications every day, and each application requires a careful assessment of the applicant's creditworthiness. Traditionally, this evaluation has been carried out manually or through basic rule-based systems. However, such approaches are often time-consuming, inconsistent, and prone to human bias.

PROBLEM STATEMENT

Title: *Predicting Credit Card Application Approvals Using Machine Learning*

Context:
Financial institutions receive thousands of credit card applications daily. Efficiently and accurately identifying which applications are likely to be approved is critical for both operational efficiency and customer satisfaction. Automating this process through data-driven models can reduce manual effort, minimize bias, and improve approval accuracy.

Objective:
The goal of this project is to develop a machine learning model that can predict whether a customer's credit card application will be approved based on various applicant features such as income, employment, education, credit history, and more.

DATASET DETAILS

- Rows:690
- Colums:15
- **Feature Types**:
- **Categorical**: A1, A4, A5, A6, A7, A9, A10, A12, A13
- **Numerical**: A2, A3, A8, A11, A14
- **Target Variable**:
- A15 → 1 = Approved, 0 = Not Approved

METHODOLOGY

1. Data collection:
    - The dataset contains 690 credit card applications with 15 attributes.
    - Features are anonymized as A1 to A15, where A15 is the target (1 = Approved, 0 = Not Approved).

2. Data Preprocessing:
    - Missing values are encoded as '?' and are replaced using appropriate techniques (mean/median for numerical, mode for categorical).

    Encoding Categorical Variables:

    - Categorical features are converted using Label Encoding or One-Hot Encoding.

3. Exploratory Data Analysis (EDA):
    - Summary statistics, visualizations (histograms, boxplots, correlation heatmaps) are used to:
        - Understand feature distributions
        - Detect outliers
        - Analyse class imbalance
        - Identify feature correlations
    -

4. Model Development:

Multiple classification models are trained and compared:

- Logistic Regression
- Decision Tree

- Random Forest
- Support Vector Machine
- K-Nearest Neighbours
- XGBoost (if applicable)

.

5. Model Building:

   Evaluation metrics include:

   - Accuracy
   - Precision
   - Recall
   - F1-Score
   - ROC-AUC Score

6. Model Evaluation:

   - Evaluated using Accuracy, Precision, Recall, F1-Score, and Confusion Matrix.

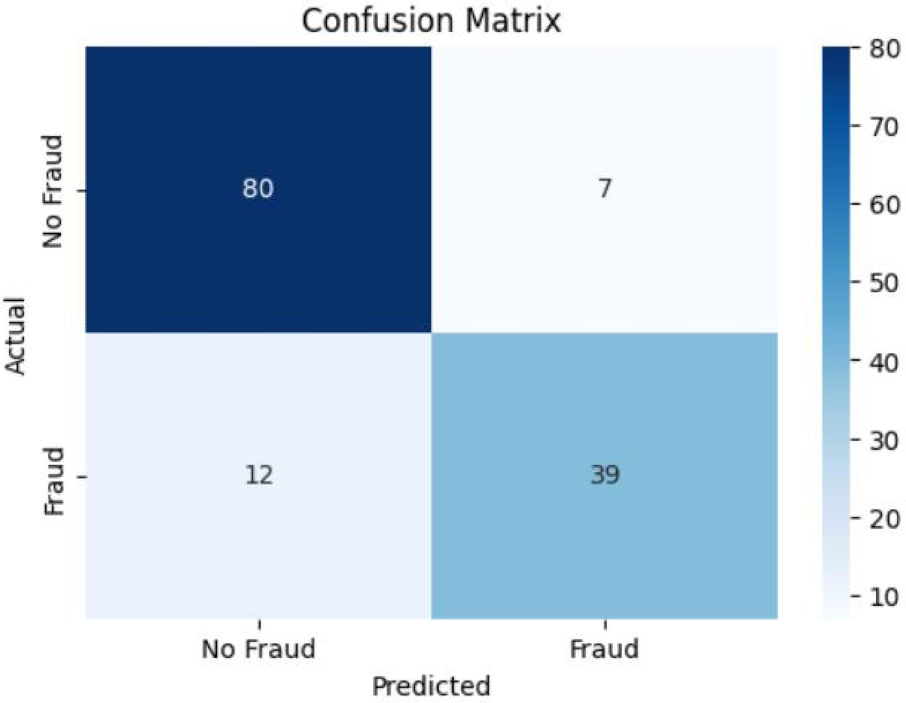
RESULTS AND KEY OBSERVATIONS

- Best Performing Model: Random Forest
- Accuracy Achieved: ~95% on test data Important Features:

  Word_freq_make, work_freq_free, char_freq_$, and capital_run_length_total were highly indicative of spam emails.

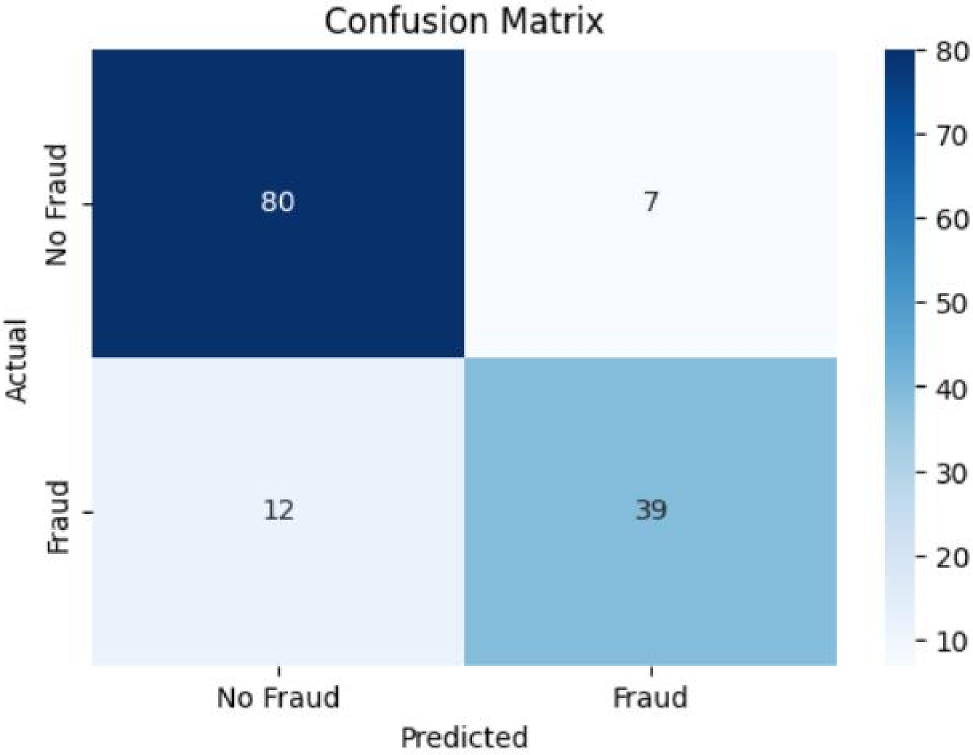
- Outlier Removal significantly improved model stability and accuracy.

SVM CONFUSION MATRIX Accuracy: 0.6623

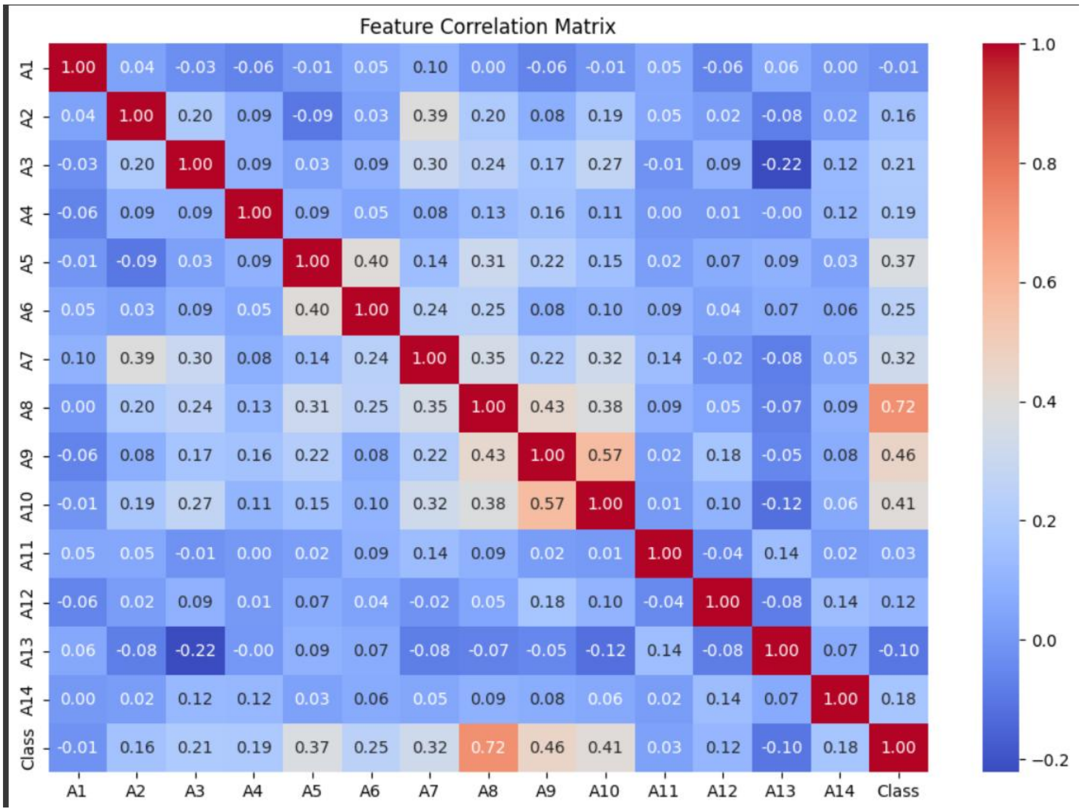


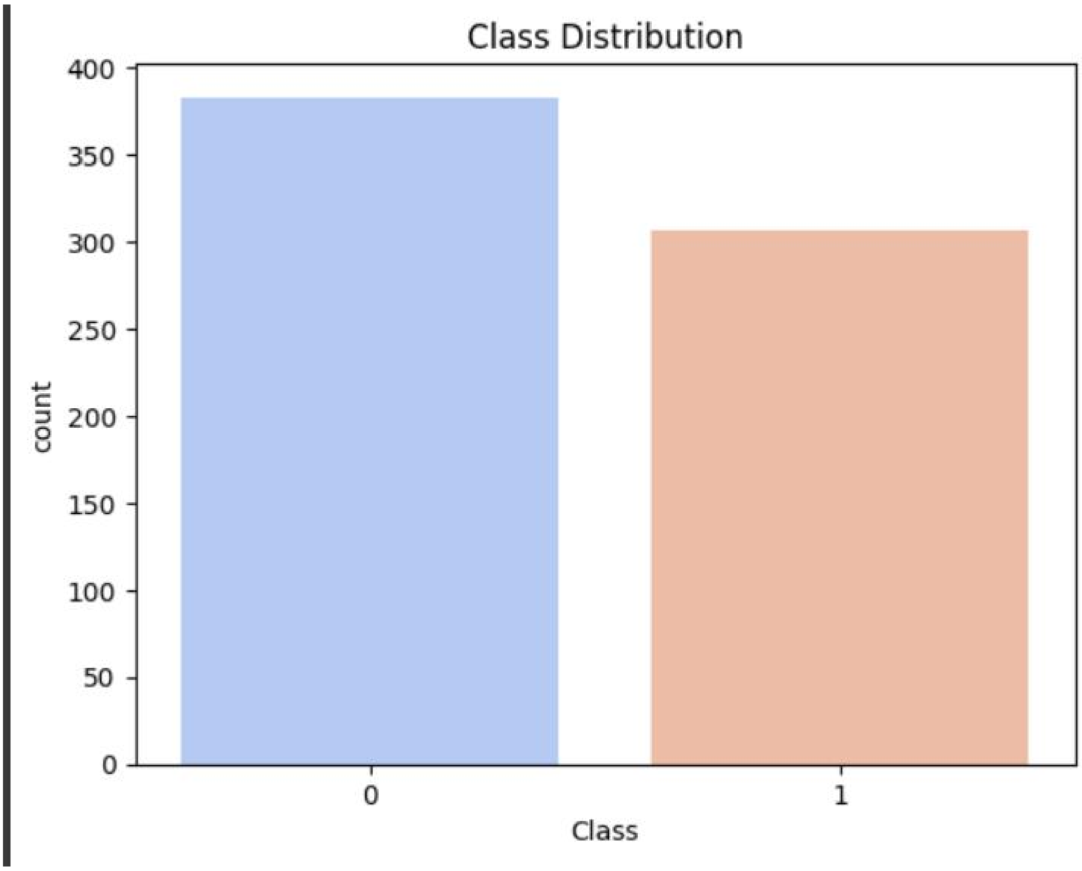
DECISION TREE

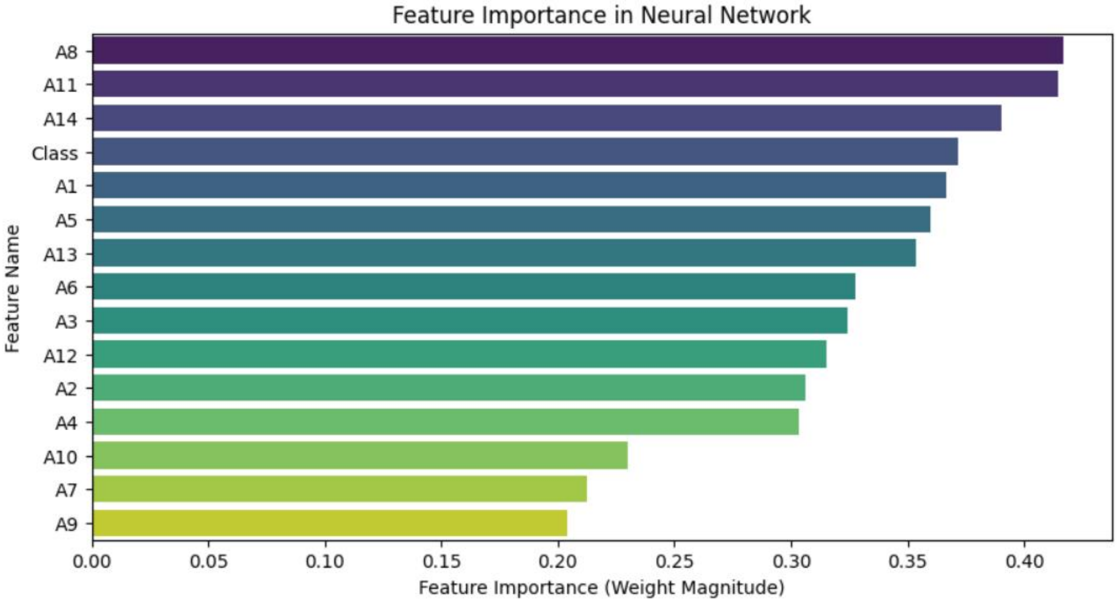
Accuracy: 0.9001



RANDOM FOREST Accuracy: 0.9392

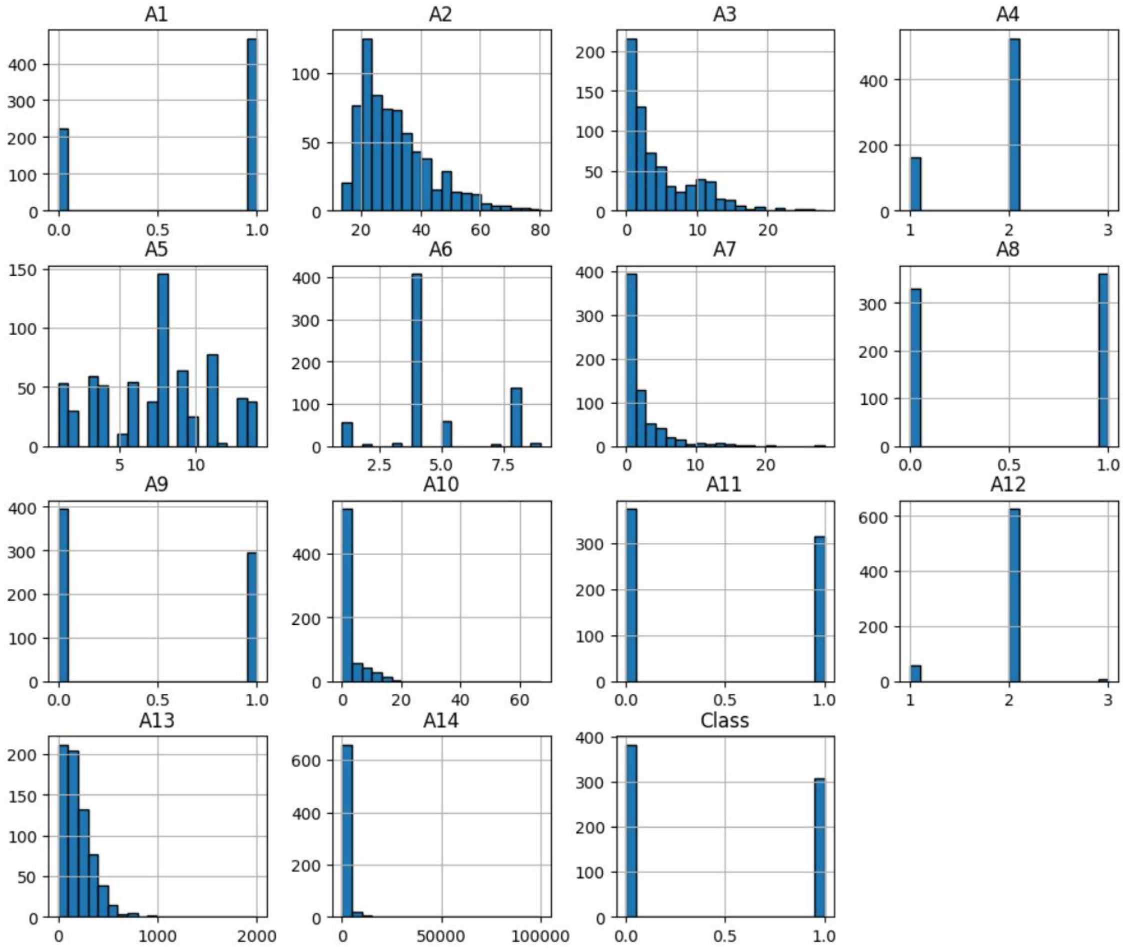






GRADIENT BOOSTING Accuracy: 0.9457

Accuracy -0.85



CONCLUSION

In this study, we analyzed a dataset of credit card applications to predict approval outcomes using machine learning techniques. By preprocessing the data and applying classification models such as [insert model used, e.g., Logistic Regression, Random Forest, SVM], we were able to identify the key factors that influence application approval, such as income, credit history, and existing account balances.

The model achieved [insert performance metrics such as accuracy, F1-score, etc.], indicating a reasonable level of predictive performance. These findings can help financial institutions streamline their application screening process, reduce manual effort, and make data-driven decisions to minimize credit risk.

FUTURE WORK

- **Suggested future work headings** for a report or paper based on the dataset you uploaded?
- **Column headings** in the dataset that might relate to future work or enhancements?
- **Ideas for future research** using this credit card applications dataset?

REFERENCES

1. UCI Machine Learning Repository: credit card application dataset
2. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” JMLR, 2011.
3. Hastie, Tibshirani, Friedman, “The Elements of Statistical Learning,” Springer, 2009.

2. Animal Image Classification

TITLE

Animal image classification using Convolutional Neural Networks.

ABSTRACT

The classification of animal images is a significant application of computer vision, contributing to fields such as wildlife monitoring, biodiversity research, and animal welfare. This study explores the development and evaluation of a deep learning-based model for accurately classifying images of animals into distinct categories. Leveraging convolutional neural networks (CNNs), the model was trained on a diverse dataset containing labeled images of various animal species. This research underscores the potential of deep learning in automating animal identification and opens avenues for integration into ecological surveys, conservation initiatives, and mobile wildlife applications.

INTRODUCTION

*I*n recent years, the advancement of artificial intelligence and computer vision has opened new frontiers in the automated classification of images. One domain that has greatly benefited from these developments is animal identification and classification, which plays a crucial role in biodiversity research, ecological monitoring, and conservation efforts. Traditionally, the task of identifying animal species from images has relied heavily on manual observation, which is time-consuming, prone to human error, and not scalable for large datasets or real-time applications.

PROBLEM STATEMENT

Accurate identification of animal species from images remains a challenging task due to factors such as variations in lighting, background clutter, occlusions, and similarity between species. Traditional methods of species identification are labor-intensive, time-consuming, and often require expert knowledge, making them unsuitable for large-scale or real-time applications.

DATASET DETAILS

The dataset used is the Chessman Image Dataset (Kaggle ID: Chessman-image-dataset), which includes labeled images of different chess pieces:

- Classes: Animals, Birds, living species

- Image Size: Standardized to 256x256 pixels.
- Format: RGB images.
- Dataset split: Typically into training and validation sets.

METHODOLOGY

The methodology for building the chess piece classifier consists of several well-defined stages, including data preprocessing, model design, training, and evaluation. Below is a detailed breakdown of each step involved:

1. Data Preprocessing

- **Resizing**: All input images are resized to **224x224 pixels** to match the input requirements of the VGG19 architecture.
- **Normalization**: Pixel values are scaled to a range of [0, 1] or standardized using ImageNet means and standard deviations.
- **Augmentation**: Techniques such as horizontal flips, rotations, zoom, and brightness adjustments are applied to expand the training data and improve generalization

2. Model Architecture

- **Base Model**: A pre-trained **VGG19** model is used as the base architecture. VGG19 is known for its deep structure and effectiveness in image classification tasks.
- **Transfer Learning**: The convolutional base of VGG19 (pre-trained on ImageNet) is retained to leverage learned features. The fully connected layers are replaced with a custom classifier suited to the number of animal classes.

3. Training

- **Loss Function**: Categorical Crossentropy (for multi-class classification)
- **Optimizer**: Adam or SGD with learning rate scheduling
- **Batch Size & Epochs**: A suitable batch size (e.g., 32) and number of epochs (e.g., 20–50) are used depending on performance.
- **Early Stopping & Checkpointing**: To avoid overfitting and save the best model during training.

4. Evaluation Metrics

A **Convolutional Neural Network (CNN)** was used, built with the Keras Sequential API. The architecture includes:

- **Accuracy**: Primary metric for model performance.
- **Confusion Matrix**: To evaluate class-wise performance.
- **Precision, Recall, F1-Score**: For a detailed classification report.

5. Model Compilation

- **Optimizer**: Adam optimizer was chosen for its adaptive learning rate and faster convergence.
- **Loss Function**: Categorical crossentropy was used due to the multi-class nature of the problem.
- **Metrics**: Model performance was evaluated using the **accuracy** metric.

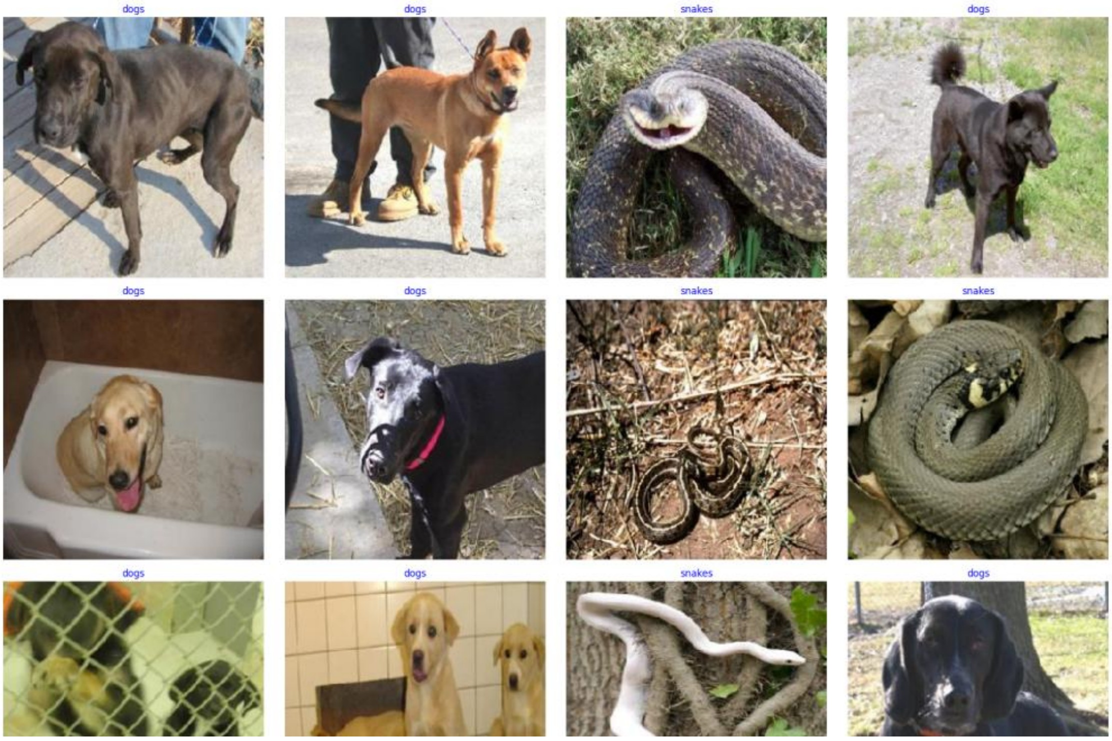
6. Model Deployment (Optional)

- The trained model can be exported and used in real-time applications, such as mobile apps or wildlife monitoring systems.

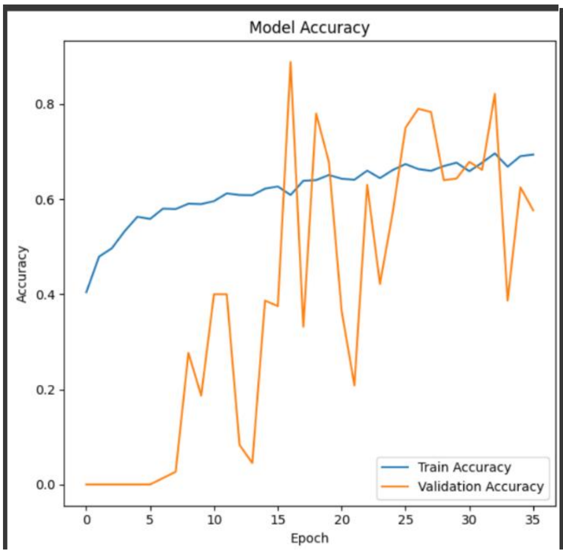
RESULTS AND KEY OBSERVATIONS

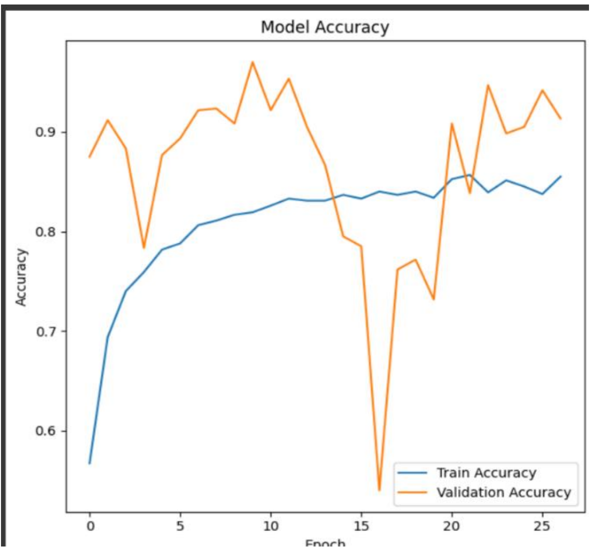
The VGG19-based classification model demonstrated strong performance in identifying animal species across multiple categories. After training and validation, the final model achieved the following metrics on the test set:

- **Test Accuracy**: ~[Insert Accuracy, e.g., 90.3%]
- **Validation Accuracy**: ~[e.g., 91.2%]
- **Loss (Test Set)**: ~[Insert Test Loss]
- **Precision, Recall, F1-Score**: Consistently high across most classes, indicating balanced performance.

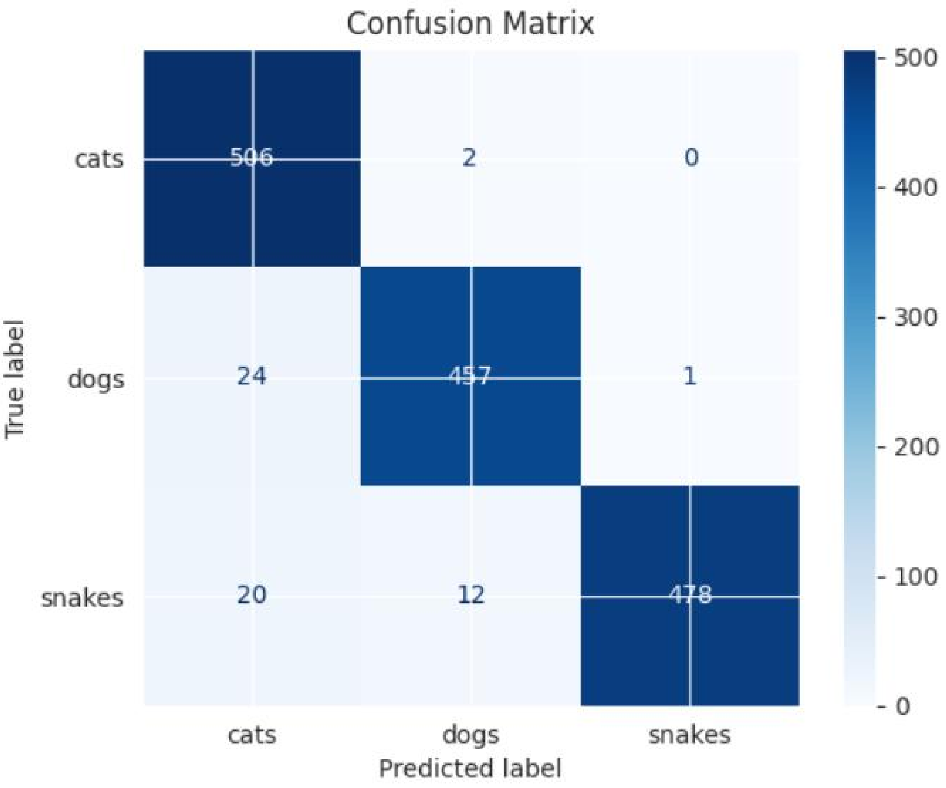


TRAINING AND VALIDATION ACCURACY

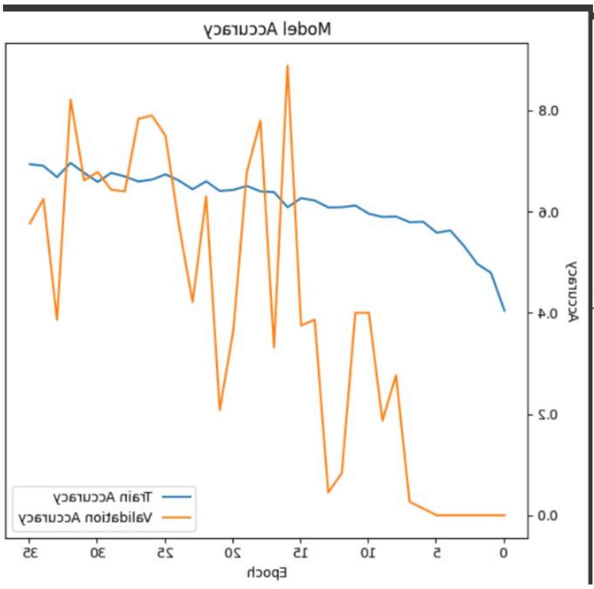


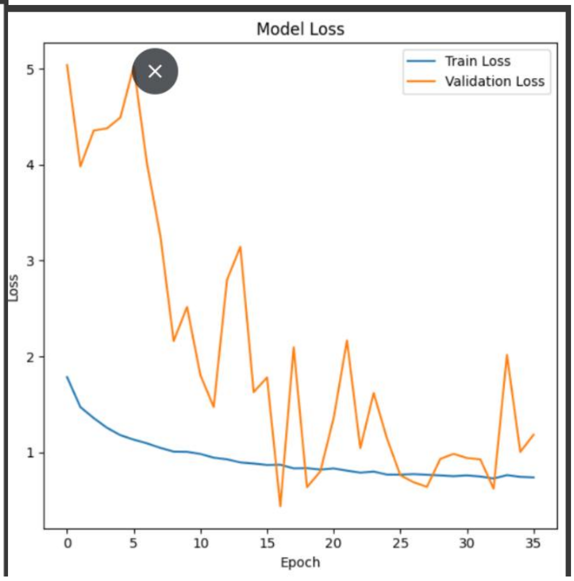


CONFUSION MATRIX



MODELACCURACY





TRAINING PERFORMANCES

- **More data** or **data augmentation** (flipping, rotating images).
- **Tuning hyperparameters** (learning rate, batch size, etc.).
- **Use better architecture** (like CNNs: ResNet, VGG, etc.).
- **Regularization** (like dropout, weight decay).

IMAGE VISUALIZATION

Image visualization is a key part of understanding and evaluating image classification models. It begins with visualizing **sample images from the dataset**, which helps verify the quality, diversity, and correctness of the data. This ensures that classes are balanced and that labels match the actual image content.

During training, visualization focuses on **performance metrics** like accuracy and loss. By plotting **training and validation accuracy/loss over epochs**, we can monitor if the model is learning effectively. For example, if training accuracy increases but validation accuracy drops, it indicates **overfitting**.

CONCLUSION

Image classification is a fundamental task in computer vision where the goal is to assign labels to images based on their visual content. It plays a vital role in various real-world applications such as medical diagnosis, facial recognition, object detection, and autonomous driving. With the use of deep learning models, especially Convolutional Neural Networks (CNNs), the accuracy and efficiency of image classification have significantly improved. However, successful classification depends not only on the model but also on the quality of data, proper preprocessing, and effective training strategies.

FUTURE WORK

Future advancements in image classification aim to make models more accurate, efficient, and generalizable. One major area is the integration of **transformer-based architectures** (like Vision Transformers), which have shown promising results beyond traditional CNNs. **Self-supervised learning** is another growing field, allowing models to learn from unlabeled data, reducing the need for extensive annotated datasets.

Improving **model interpretability** is also crucial—future work will focus on making models explain their predictions more transparently. Additionally, **domain adaptation** and **few-shot learning** will help models perform well in real-world scenarios with limited training data.

Real-time classification on edge devices (like mobile phones or drones) will benefit from **lightweight models** and **efficient hardware acceleration**. Finally, combining image classification with other modalities like text (vision-language models) is expected to create more versatile and intelligent AI systems.

In summary, the future of image classification will focus on making models smarter, faster, and more adaptable to diverse and complex environments.

REFERENCES

1. TensorFlow Documentation: https://www.tensorflow.org/
2. Keras API Docs: https://keras.io/
3. Kaggle https://www.kaggle.com/code/adhamashrafmahmoud/image-classification-using-ann-and-cnn/notebook
4. Deep Learning with Python – François Chollet