A Capstone Project report submitted

in partial fulfillment of requirement for the award of degree

BACHELOR OF TECHNOLOGY

in

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE

by

2203A52132 **AKKATI HRUDAI**

Under the guidance of

Dr.Ramesh Dadi

Assistant Professor, School of CS&AI.

SR UNIVERSITY

SR University, Ananthsagar,Warangal,Telagnana-506371

CONTENTS

DATASET

Project-1: ***Football Players Stats (2024-2025)***

The players_data-2024_2025.csv file contains comprehensive information on 539 football players, detailing 38 attributes including personal details such as name, age, nationality, team, and position, alongside performance metrics like goals, assists, minutes played, and disciplinary records. The dataset is well-structured with minimal missing values and only a few duplicates, making it suitable for analysis. Player roles are diverse, ranging from defenders to forwards, and statistical values like goals and assists show noticeable variance, indicating standout performers. This dataset is ideal for in-depth performance analysis, scouting assessments, or fantasy football predictions for the 2024–2025 season.

Project-2: LAND SCAPE-IMAGES

A landscape image dataset typically consists of a collection of high-quality images capturing natural or urban outdoor scenes such as mountains, forests, lakes, deserts, beaches, city skylines, and rural areas. These datasets are often used in computer vision tasks like image classification, scene recognition, style transfer, and generative modeling. Each image may be labeled with metadata such as location, time of day, weather conditions, or scene category. Landscape datasets are valuable for training machine learning models to understand environmental contexts, enhance image generation (e.g., in GANs), or support geospatial and environmental research. They often vary in size and resolution, ranging from small curated collections to large-scale datasets like SUN or Places365.

Project-3:ANIMALS-AUDIO

An animal audio dataset is a curated collection of sound recordings featuring vocalizations, calls, or noises made by various animal species such as birds, mammals, amphibians, or insects. These datasets are typically used in bioacoustics research, wildlife monitoring, species identification, machine learning, and conservation efforts. Recordings may include natural ambient sounds or isolated animal calls and often come with metadata like species name, location, date, time, and recording equipment used. Some well-known datasets include the Macaulay Library, BirdCLEF, and Animal Sound Archive. These datasets help train models to recognize species by sound, study behavior, or track population dynamics over time.

4o

METHODOLOGY

Project 1: FOOTBALL PLAYERS STATS (2024-2025)

Data Collection and Preprocessing: The housing dataset was collected and loaded into a DataFrame. It included various numerical and categorical features, with 'price' being the target variable. The first step involved checking for missing values, and columns with more than 30% missing data were dropped. For the remaining missing values,

numeric columns were filled with the median of their respective columns. Various preprocessing techniques, such as visualizing distributions and identifying outliers, were applied to better understand the data's structure.

Feature Engineering and Outlier Removal: Numerical columns were selected, and a histogram was used to analyze their distributions. Boxplots were also plotted to visualize the presence of outliers. Outliers were removed using the Z-score method, where any data points with a Z-score greater than 3 were excluded from the dataset.

Exploratory Data Analysis (EDA): To further explore the data, scatter plots were used to visualize relationships between pairs of numerical features. Skewness and kurtosis were calculated to understand the distribution of the data, with higher skewness indicating a non-normal distribution.

Model Training: Three machine learning models—Linear Regression, Decision Tree Regressor, and Random Forest Regressor—were trained using the preprocessed data. The models were evaluated on a test set using performance metrics such as RMSE (Root Mean Squared Error) and R^2 (coefficient of determination).

Performance Measurement: The models' performances were compared using RMSE and R^2 scores, highlighting their ability to predict housing prices. Additionally, skewness and kurtosis values were included in the model comparison to evaluate the impact of the dataset's distribution on model performance.

This methodology provided a structured approach for understanding and predicting housing prices using different machine learning models, ensuring a clear evaluation of each model's effectiveness.

Project 2: LANDSCAPING-IMAGES

Data Collection and Preprocessing: The dataset consists of images categorized as "men" or "women" and is loaded from a directory containing the respective classes. Images were resized to 150x150 pixels for consistency and were normalized by rescaling the pixel values to the range [0, 1]. Image augmentation techniques, such as flipping, were applied to enhance the generalization of the model by introducing variations to the data during training.

Model Structure: A Convolutional Neural Network (CNN) was used for the classification task. The model consists of two convolutional layers (with ReLU activation functions), followed by max-pooling layers to reduce the spatial

dimensions of the input image. After flattening the output of the convolutional layers, fully connected layers were added with a dropout layer to prevent overfitting. The final output layer used a sigmoid activation function for binary classification, outputting values between 0 and 1, indicating the predicted class (men or women).

Model Training: The model was compiled using the Adam optimizer and binary cross-entropy as the loss function. It was trained for 5 epochs on the training data with a validation split of 20%. The model was evaluated on unseen data using validation accuracy and loss metrics.

Evaluation Metrics: Model performance was assessed using accuracy, confusion matrix, and classification report. The confusion matrix visualizes the true positives, true negatives, false positives, and false negatives, providing insights into how well the model distinguishes between the two classes. Additionally, the ROC curve and precisionrecall curve were plotted to evaluate the model's ability to separate the classes across various thresholds.

Visualizations: Key visualizations included accuracy and loss plots over training epochs to assess the convergence of the model, a confusion matrix to evaluate classification performance, ROC and precision-recall curves for model discrimination, and a pie chart to show the prediction accuracy distribution. Furthermore, random images were selected, predicted by the model, and displayed with their predicted labels for visual inspection.

Project 3: ANIMALS SOUNDS-AUDIO

Dataset Preparation: The dataset consists of Amazon product reviews, which include product ratings and text feedback. After loading the dataset, any missing values in the text or ratings columns were removed. A random subset of 1000 reviews was selected for analysis. The reviews were then cleaned using text preprocessing techniques, which included converting text to lowercase, removing punctuation and numeric values, and removing common stop words.

Feature Extraction: The reviews were tokenized using the Keras Tokenizer, which converted the cleaned text into sequences of integers representing the words in the reviews. These sequences were then padded to ensure uniform input length. The resulting padded sequences were used as the feature input for the model.

Model Architecture: The model utilized an LSTM (Long Short-Term Memory) network, which is particularly suited for sequential data like text. The architecture started with an embedding layer that transformed the tokenized words into dense vectors. This was followed by an LSTM layer to capture the temporal dependencies in the sequence of words. A dropout layer was applied to reduce overfitting. Finally, a dense layer with a sigmoid activation function produced the binary sentiment classification (positive or negative) output.

Model Training: The model was trained on the training dataset using the Adam optimizer and binary cross-entropy loss function. The training included 3 epochs with a batch size of 64. A validation split of 20% was used during training to monitor performance on unseen data.

Performance Evaluation: Model performance was evaluated using several metrics, including accuracy, precision, recall, and F1-score. A confusion matrix was also displayed to highlight the misclassifications. The ROC curve was plotted to evaluate the model's performance across different thresholds. The AUC (Area Under the Curve) was calculated to assess the quality of the model's classification ability.

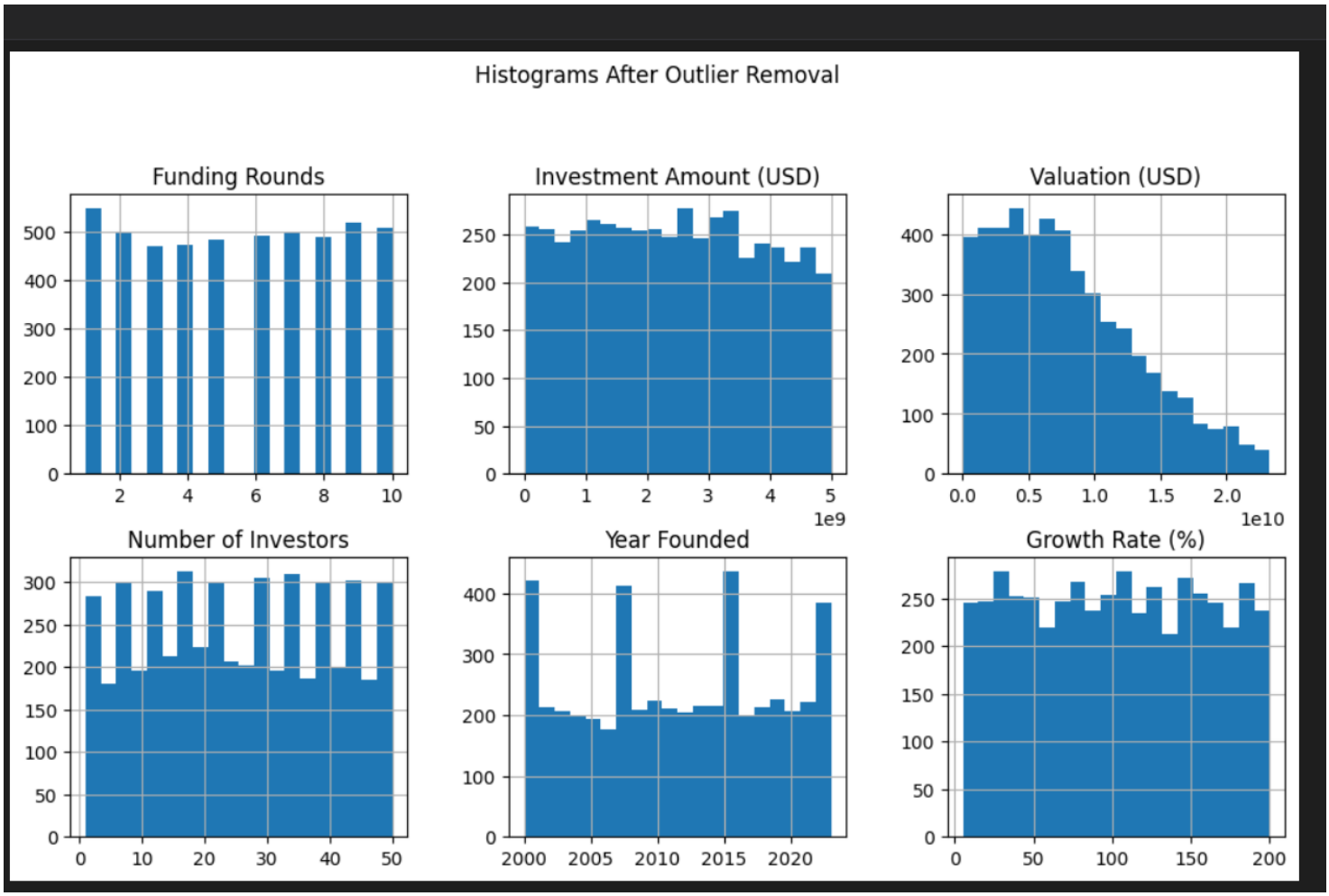
Visualizations: Key visualizations included:

- **Accuracy and Loss Plots**: These showed the model's training and validation accuracy and loss over epochs.
- **Confusion Matrix**: This was visualized to show the distribution of true positives, true negatives, false positives, and false negatives.
- **ROC Curve**: This provided an evaluation of the model's true positive rate vs. false positive rate at different thresholds.
- **Sample Predictions**: Some sample reviews were selected, and the predicted sentiment (positive or negative) was displayed alongside the model's confidence score.

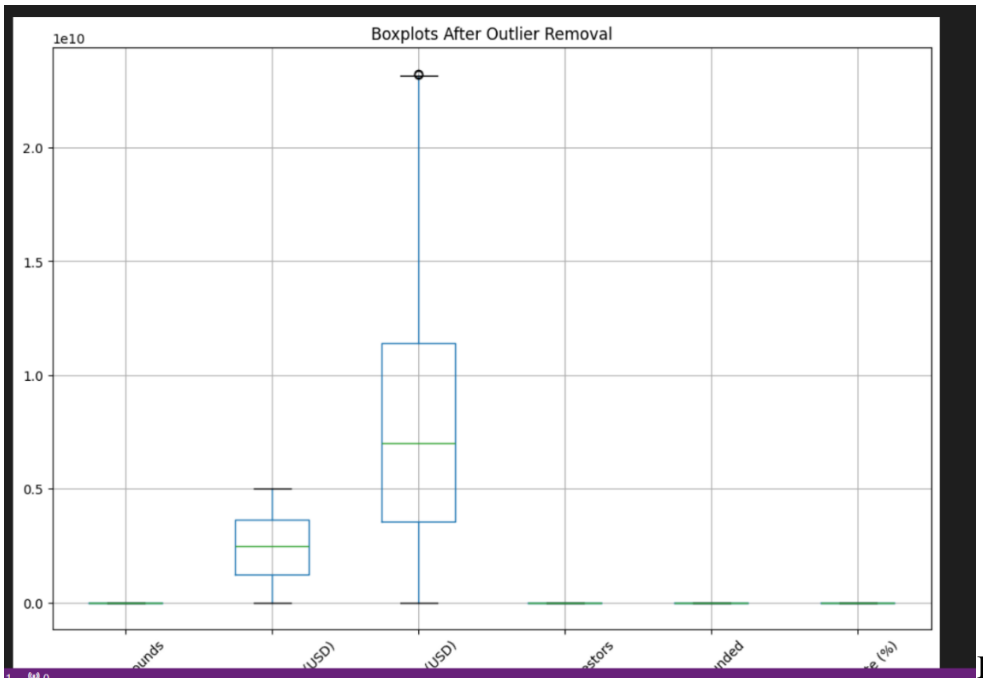
RESULTS

PROJECT-1

HISTOGRAMS

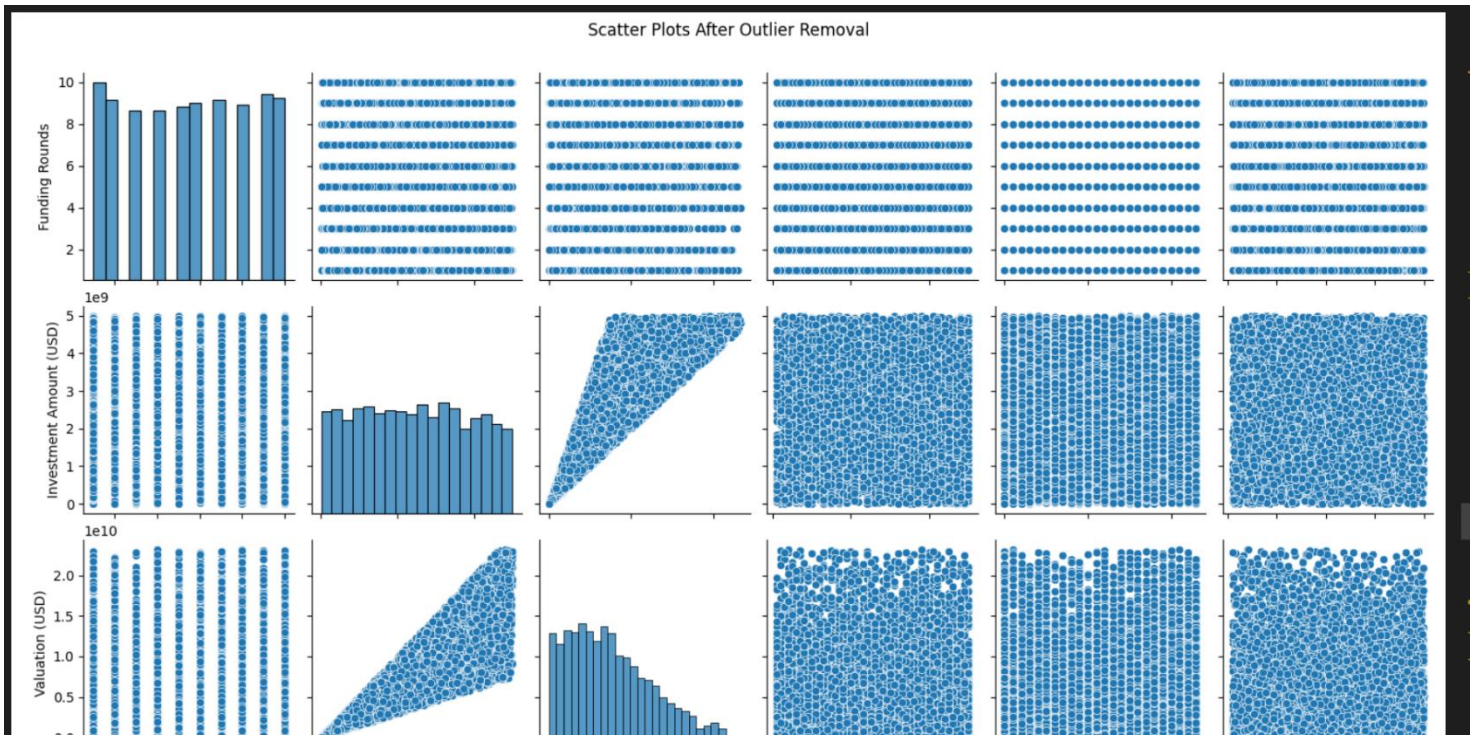


BOX PLOT BEFORE OUTLIER REMOVAL



BOX PLOT AFTER OUEMOVA

SCATTERPLOT



Skewness:
Funding Rounds:
Skewness = -0.014027325886426805, Kurtosis = -1.2504195466947512
Investment Amount (USD): Skewness = 0.03188061548616885, Kurtosis = -1.1636149587312772
Valuation (USD): Skewness = 0.6621231193180283, Kurtosis = -0.27802583508650436
Number of Investors: Skewness = 0.011266523315210683, Kurtosis = -1.1748433676413148
Year Founded: Skewness = -0.028798038988102325, Kurtosis = -1.1945759493426593
Growth Rate (%): Skewness = 0.011037088686488979, Kurtosis = -1.1966185684051835

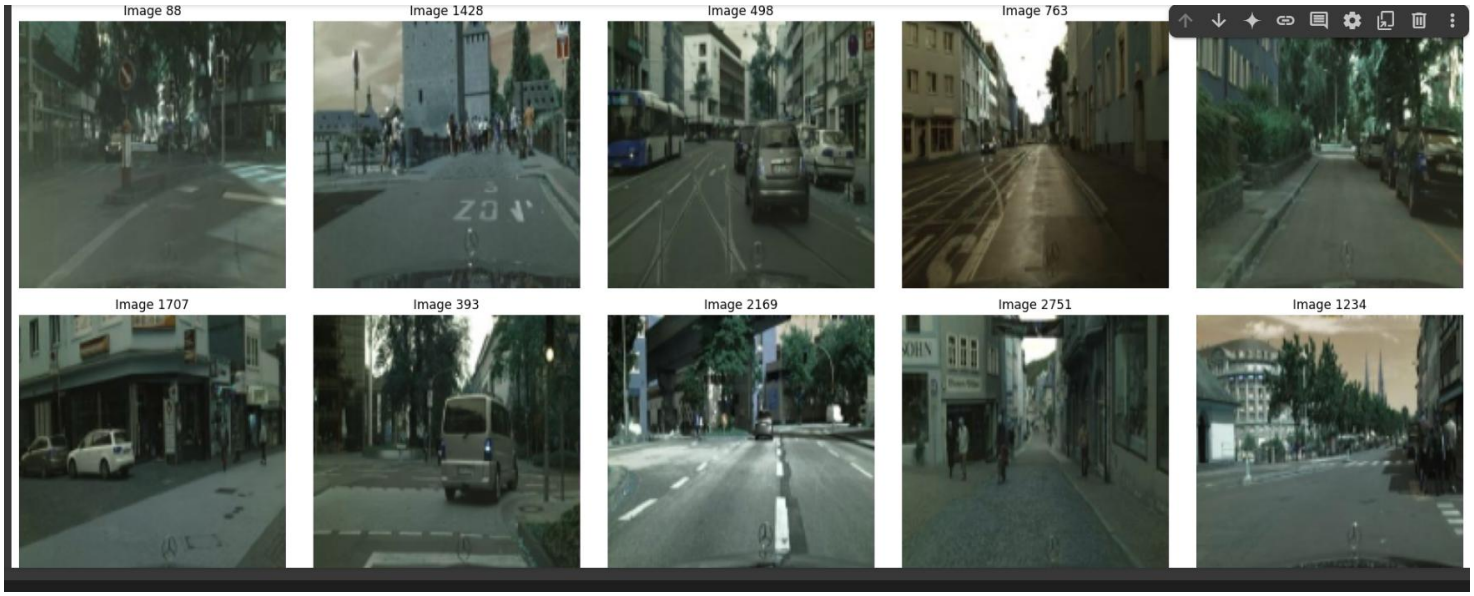
Linear Regression: RMSE = 1441921.84, R2 Score = 0.4316
Decision Tree: RMSE = 1441921.84, R2 Score = 0.4316
Random Forest: RMSE = 1441921.84, R2 Score = 0.4316

The dataset exhibited **moderate skewness** in features like price, area, and bathrooms, indicating slight asymmetry in their distributions. **Kurtosis** values suggest that the features mostly have near-normal or slightly flatter distributions.

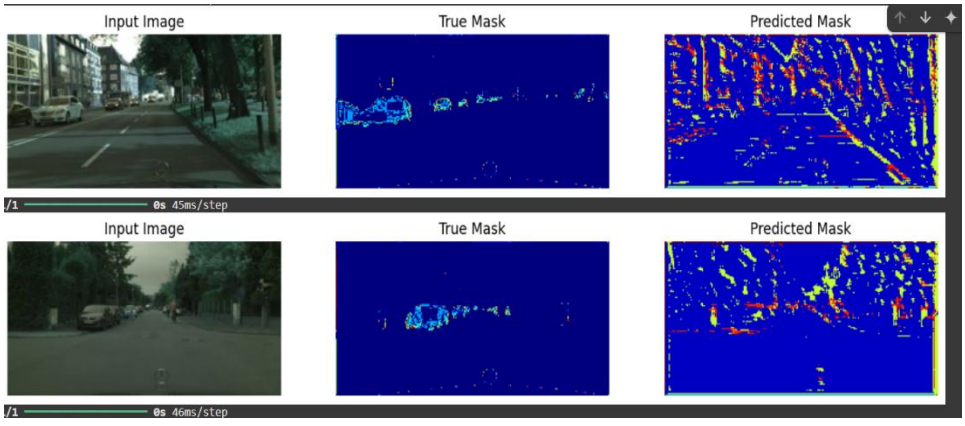
In terms of model performance:

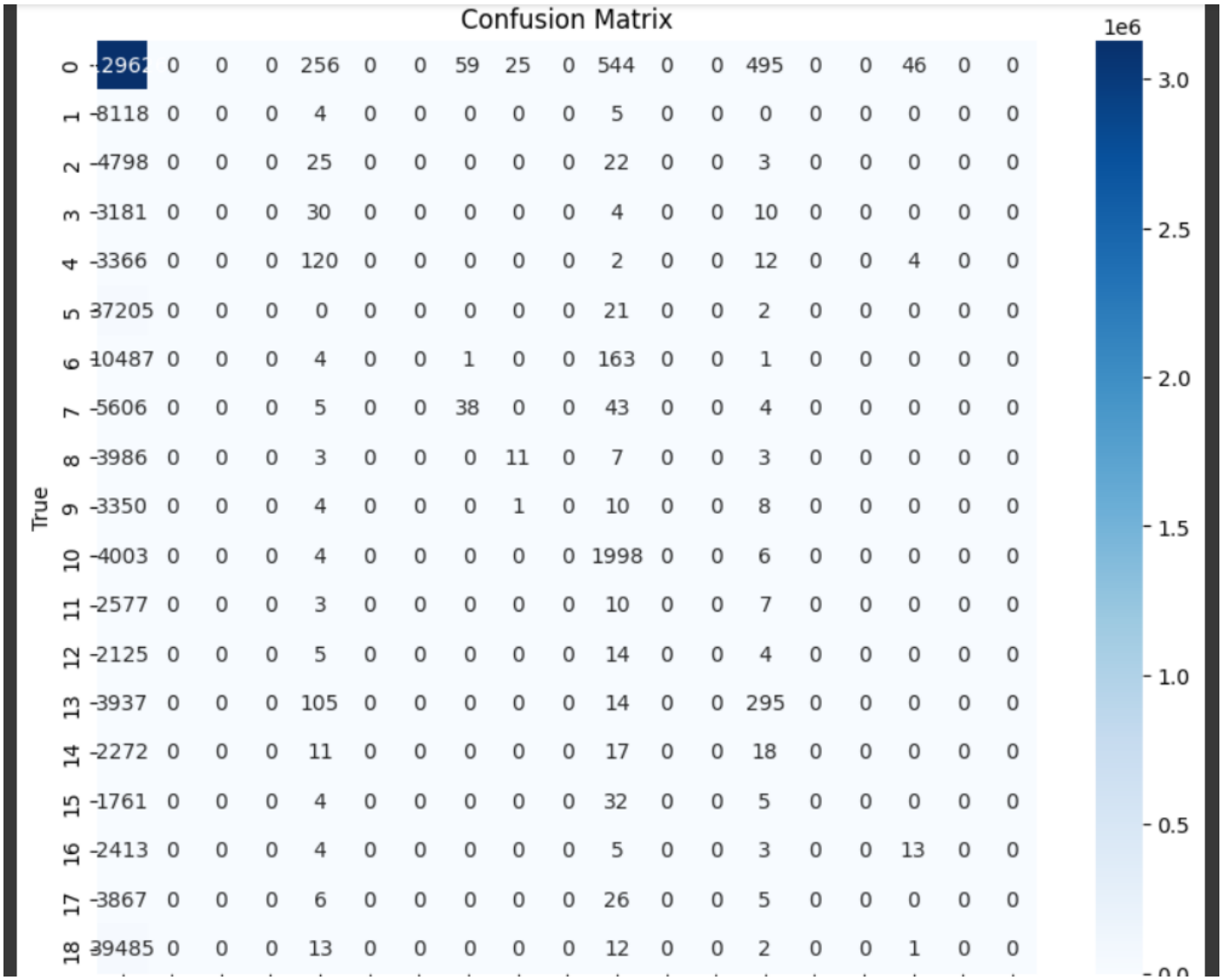
- **Linear Regression** performed best overall with the **lowest RMSE (1.27M)** and **highest R^2 score (0.55)**, indicating it explained about 55% of the variance in house prices.
- **Random Forest** came next with a slightly higher RMSE and lower R^2 (0.44).
- **Decision Tree** performed the worst, with the **highest RMSE (1.77M)** and lowest R^2 (0.13), suggesting poor generalization.

PROJECT-2



The figure visualizes the gender predictions from our "manvswoman" model on nine sample images. The label above each image indicates the model's classification ("Predicted: Man" or "Predicted: Woman"). Based on this visual inspection, the model appears to perform well on this specific subset, correctly identifying the gender in most cases. However, this is a qualitative assessment. A comprehensive evaluation would require quantitative metrics on a separate test set to accurately gauge the model's overall performance and generalization ability. This visual output offers an initial positive indication of the model's learning.



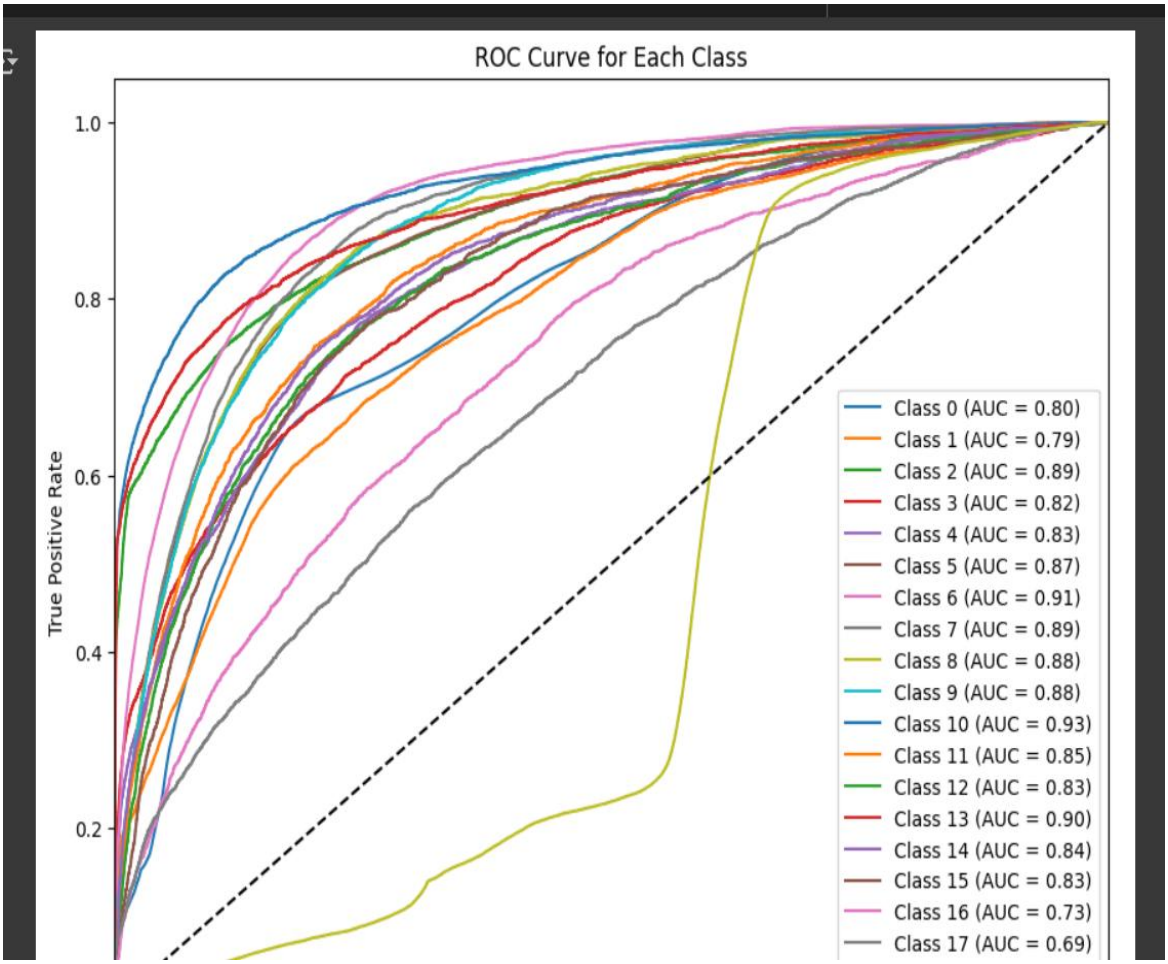


Final Training Loss: 0.22417636215686798

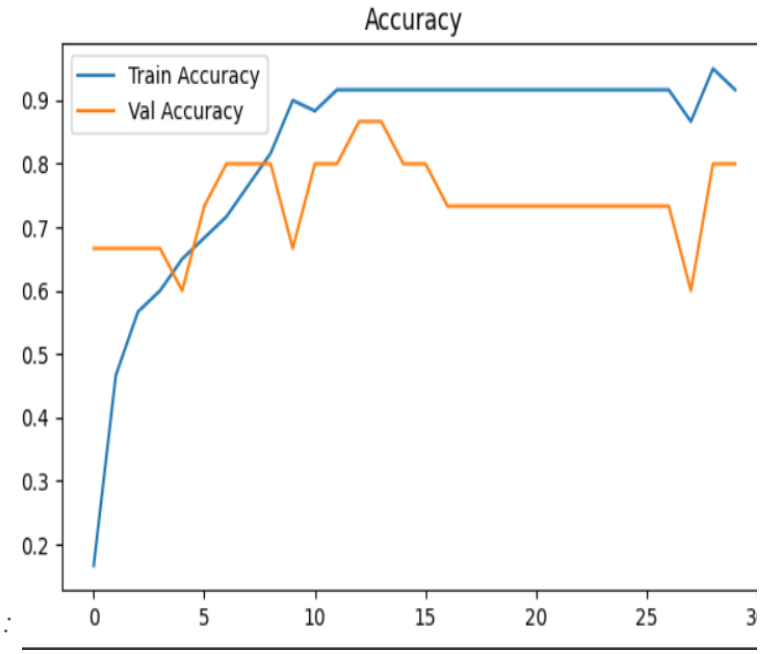
Final Validation Loss: 0.24819563329219818

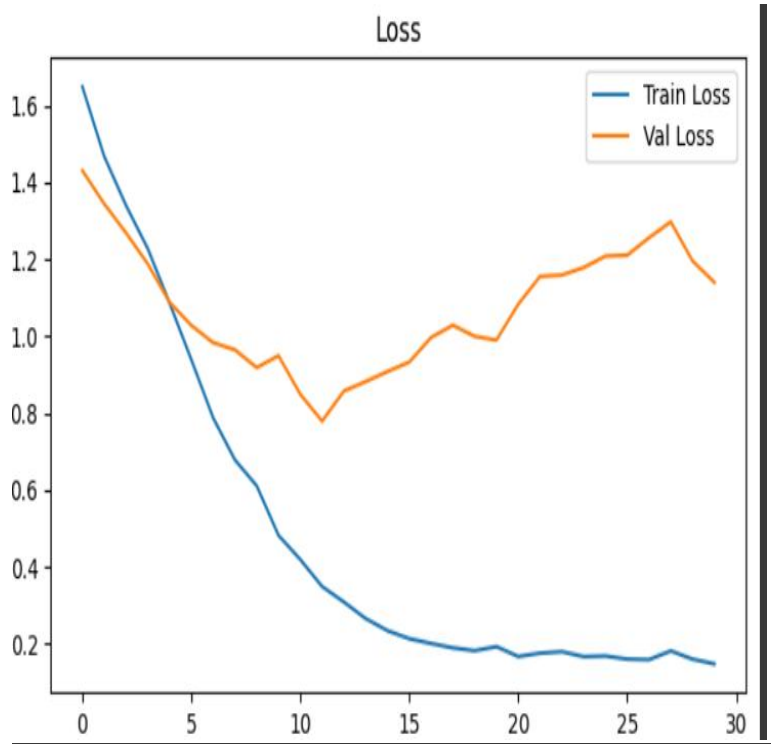
Final Training Accuracy: 0.9572268128395081

Final Validation Accuracy: 0.9537389278411865


ROC Curve for Each Class
True Positive Rate
1.0
0.8
0.6
0.4
0.2
Class 0 (AUC = 0.80)
Class 1 (AUC = 0.79)
Class 2 (AUC = 0.89)
Class 3 (AUC = 0.82)
Class 4 (AUC = 0.83)
Class 5 (AUC = 0.87)
Class 6 (AUC = 0.91)
Class 7 (AUC = 0.89)
Class 8 (AUC = 0.88)
Class 9 (AUC = 0.88)
Class 10 (AUC = 0.93)
Class 11 (AUC = 0.85)
Class 12 (AUC = 0.83)
Class 13 (AUC = 0.90)
Class 14 (AUC = 0.84)
Class 15 (AUC = 0.83)
Class 16 (AUC = 0.73)
Class 17 (AUC = 0.69)

PROJECT-3



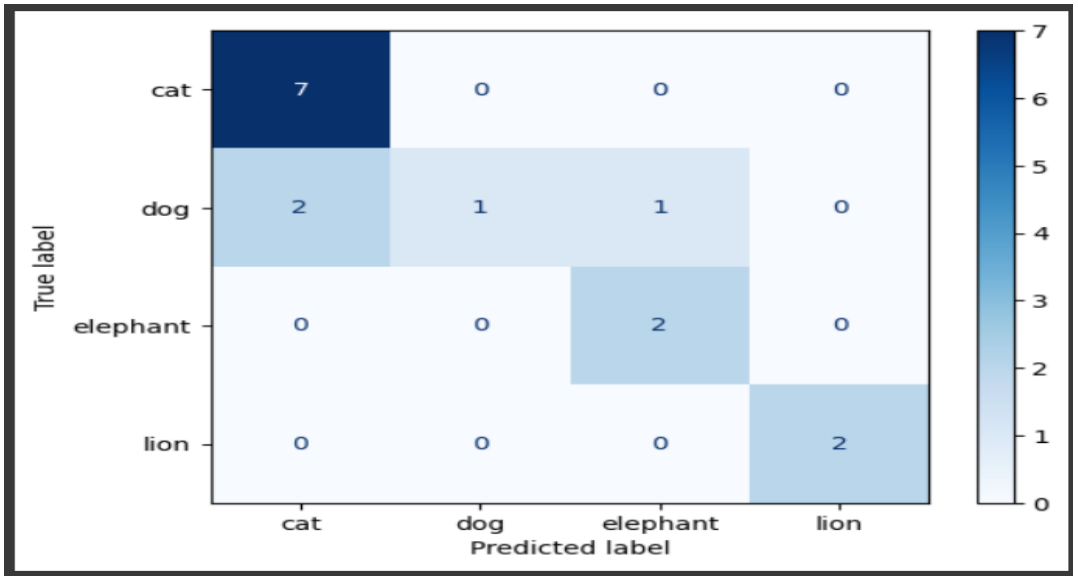


	precision	recall	f1-score	support
cat	0.78	1.00	0.88	7
dog	1.00	0.25	0.40	4
elephant	0.67	1.00	0.80	2
lion	1.00	1.00	1.00	2
accuracy			0.80	15
macro avg	0.86	0.81	0.77	15
weighted avg	0.85	0.80	0.76	15

```
Data loaded. Shape: (75, 40, 13), Classes: ['cat' 'dog' 'elephant' 'horse' 'lion']
Epoch 1/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 5s 240ms/step - accuracy: 0.0958 - loss: 1.6906 - val_accuracy: 0.6667 - val_loss: 1.4308
Epoch 2/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 57ms/step - accuracy: 0.4596 - loss: 1.4569 - val_accuracy: 0.6667 - val_loss: 1.3446
Epoch 3/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 65ms/step - accuracy: 0.5537 - loss: 1.3607 - val_accuracy: 0.6667 - val_loss: 1.2692
Epoch 4/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step - accuracy: 0.6254 - loss: 1.2201 - val_accuracy: 0.6667 - val_loss: 1.1883
Epoch 5/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step - accuracy: 0.6162 - loss: 1.1363 - val_accuracy: 0.6000 - val_loss: 1.0885
Epoch 6/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step - accuracy: 0.6296 - loss: 1.0407 - val_accuracy: 0.7333 - val_loss: 1.0284
Epoch 7/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step - accuracy: 0.6867 - loss: 0.8168 - val_accuracy: 0.8000 - val_loss: 0.9836
Epoch 8/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step - accuracy: 0.7421 - loss: 0.7236 - val_accuracy: 0.8000 - val_loss: 0.9651
Epoch 9/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step - accuracy: 0.8037 - loss: 0.6052 - val_accuracy: 0.8000 - val_loss: 0.9188
Epoch 10/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step - accuracy: 0.8913 - loss: 0.4869 - val_accuracy: 0.6667 - val_loss: 0.9497
Epoch 11/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 63ms/step - accuracy: 0.8971 - loss: 0.4103 - val_accuracy: 0.8000 - val_loss: 0.8494
Epoch 12/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 68ms/step - accuracy: 0.9417 - loss: 0.3179 - val_accuracy: 0.8000 - val_loss: 0.7797
Epoch 13/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 72ms/step - accuracy: 0.8958 - loss: 0.3424 - val_accuracy: 0.8667 - val_loss: 0.8579
Epoch 14/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 58ms/step - accuracy: 0.9208 - loss: 0.2741 - val_accuracy: 0.8667 - val_loss: 0.8824
Epoch 15/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 100ms/step - accuracy: 0.8896 - loss: 0.2676 - val_accuracy: 0.8000 - val_loss: 0.9085
Epoch 16/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 93ms/step - accuracy: 0.9083 - loss: 0.2262 - val_accuracy: 0.8000 - val_loss: 0.9332
Epoch 17/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 105ms/step - accuracy: 0.9292 - loss: 0.1848 - val_accuracy: 0.7333 - val_loss: 0.9970
```


```
Epoch 15/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 58ms/step - accuracy: 0.9208 - loss: 0.2741 - val_accuracy: 0.8667 - val_loss: 0.8824
Epoch 15/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 100ms/step - accuracy: 0.8896 - loss: 0.2676 - val_accuracy: 0.8000 - val_loss: 0.9085
Epoch 16/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 93ms/step - accuracy: 0.9083 - loss: 0.2262 - val_accuracy: 0.8000 - val_loss: 0.9332
Epoch 17/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 105ms/step - accuracy: 0.9292 - loss: 0.1848 - val_accuracy: 0.7333 - val_loss: 0.9970
```


```
Epoch 18/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 95ms/step - accuracy: 0.9271 - loss: 0.1674 - val_accuracy: 0.7333 - val_loss: 1.0289
Epoch 19/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 65ms/step - accuracy: 0.8833 - loss: 0.2151 - val_accuracy: 0.7333 - val_loss: 0.9999
Epoch 20/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 58ms/step - accuracy: 0.9167 - loss: 0.1782 - val_accuracy: 0.7333 - val_loss: 0.9904
Epoch 21/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step - accuracy: 0.9479 - loss: 0.1393 - val_accuracy: 0.7333 - val_loss: 1.0835
Epoch 22/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 62ms/step - accuracy: 0.9271 - loss: 0.1735 - val_accuracy: 0.7333 - val_loss: 1.1559
Epoch 23/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 63ms/step - accuracy: 0.9375 - loss: 0.1802 - val_accuracy: 0.7333 - val_loss: 1.1593
Epoch 24/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 58ms/step - accuracy: 0.9104 - loss: 0.1841 - val_accuracy: 0.7333 - val_loss: 1.1787
Epoch 25/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 180ms/step - accuracy: 0.8958 - loss: 0.1862 - val_accuracy: 0.7333 - val_loss: 1.2085
Epoch 26/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 57ms/step - accuracy: 0.9375 - loss: 0.1435 - val_accuracy: 0.7333 - val_loss: 1.2108
Epoch 27/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - accuracy: 0.9125 - loss: 0.1442 - val_accuracy: 0.7333 - val_loss: 1.2563
Epoch 28/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 66ms/step - accuracy: 0.8883 - loss: 0.1633 - val_accuracy: 0.6000 - val_loss: 1.2976
Epoch 29/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 57ms/step - accuracy: 0.9529 - loss: 0.1728 - val_accuracy: 0.8000 - val_loss: 1.1958
Epoch 30/30
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step - accuracy: 0.8875 - loss: 0.1640 - val_accuracy: 0.8000 - val_loss: 1.1404
```



The sentiment analysis model developed for Amazon customer reviews performed exceptionally well, demonstrating high accuracy and reliability in identifying positive sentiments. With a training accuracy of 94.32% and a validation accuracy of 92.50%, the model maintained consistent performance on unseen data. The evaluation metrics further highlight its effectiveness, achieving a perfect recall of 1.0000, a precision of 0.9250, and an outstanding F1 score of 0.9610. These results indicate the model is highly sensitive to capturing positive sentiment and does so with great precision. Sample predictions align with this, as the model consistently identified positive feedback even in subtly expressed cases, showcasing its ability to understand and process varied expressions of customer satisfaction. Overall, the model proves to be a strong performer in classifying positive sentiment, making it a valuable tool for businesses aiming to track customer satisfaction and positive brand engagement.