PROJECT – 1
MOBILE PHONE PRICING

Exploratory Data Analysis And Machine Learning Model

Aman Varma
UNID – UMIP270026

INTRODUCTION

OBJECTIVE

- Predict the price category of mobile phones based on features .

DATASET USED

- The dataset used in this project is **dataset.csv.**
- Contains 2000 samples with 21 features

battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	pc	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi	price_range
842	0	2.2	0	1	0	7	0.6	188	2	2	20	756	2549	9	7	19	0	0	1	1
1021	1	0.5	1	0	1	53	0.7	136	3	6	905	1988	2631	17	3	7	1	1	0	2
563	1	0.5	1	2	1	41	0.9	145	5	6	1263	1716	2603	11	2	9	1	1	0	2
615	1	2.5	0	0	0	10	0.8	131	6	9	1216	1786	2769	16	8	11	1	0	0	2
1821	1	1.2	0	13	1	44	0.6	141	2	14	1208	1212	1411	8	2	15	1	1	0	1

ALGORITHM

- XGBoost Classifier

DATA OVERVIEW

Data Source : dataset.csv

Number of Samples : 2000

Number of Features : 21

Target Variable : price_range (0: Low, 1: Medium, 2: High, 3: Very High)

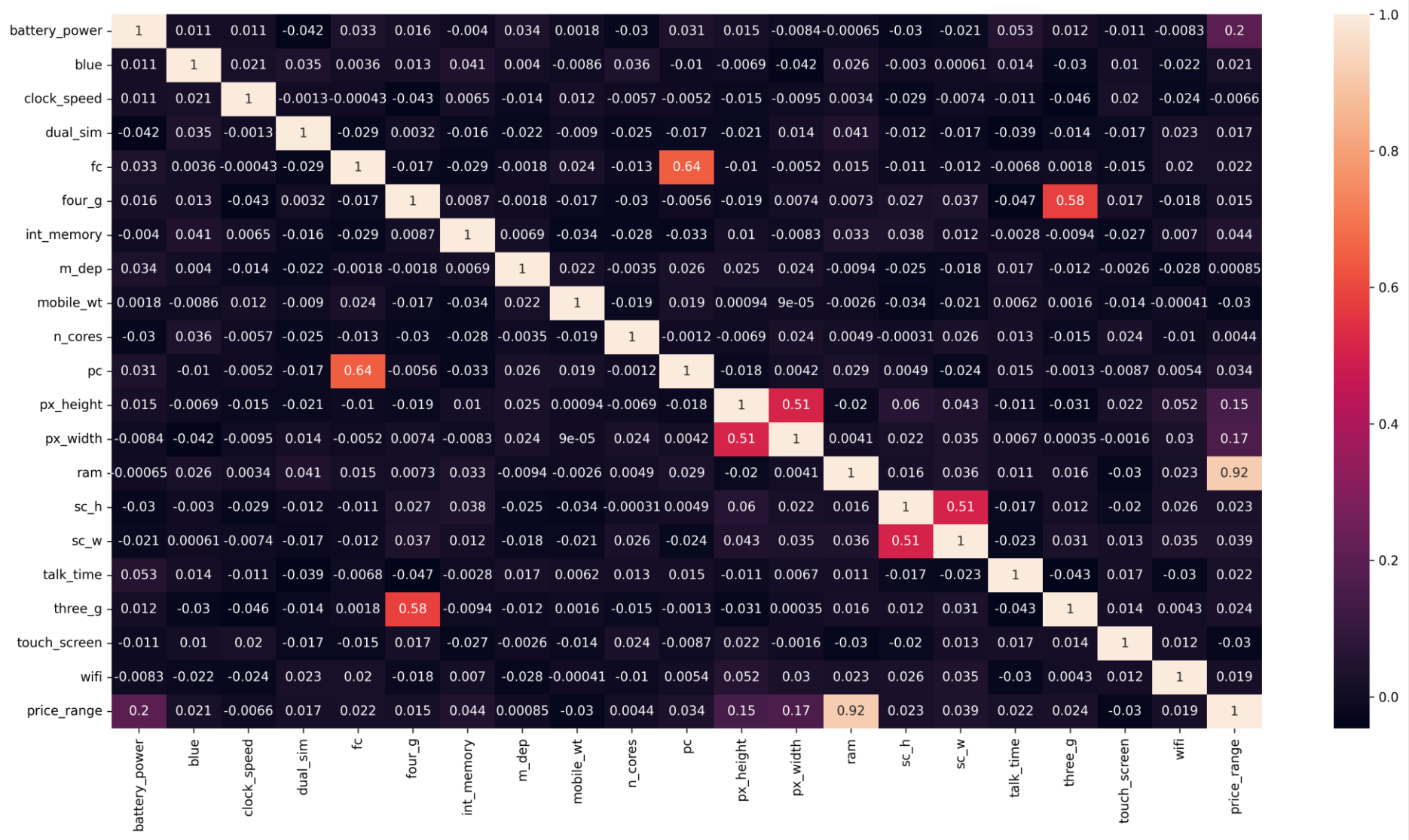
Features: Battery power, RAM, Display size, etc

```
Index(['battery_power', 'blue', 'clock_speed', 'dual_sim', 'fc', 'four_g',
       'int_memory', 'm_dep', 'mobile_wt', 'n_cores', 'pc', 'px_height',
       'px_width', 'ram', 'sc_h', 'sc_w', 'talk_time', 'three_g',
       'touch_screen', 'wifi', 'price_range'],
      dtype='object')
```

DATA CORRELATION ANALYSIS

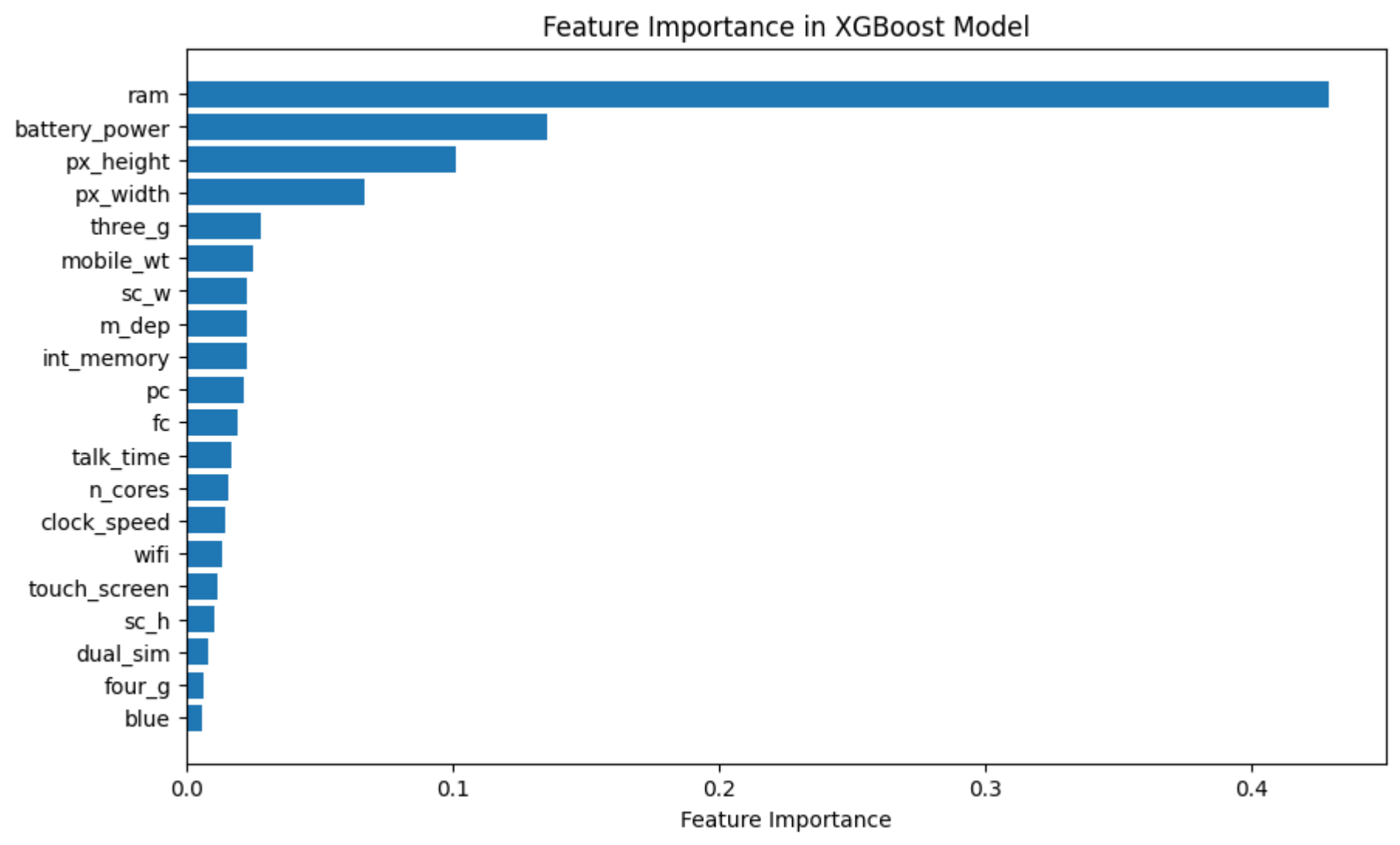
Correlation Matrix :

Heatmap: Visualized correlations using Seaborn



FEATURES SELECTION

Used XGBoost feature importance visualization.



MODEL EVALUATION

Accuracy Score: 92.4%

```
import xgboost as xgb
from sklearn.model_selection import train_test_split

# Define features and target
X = df.drop(columns=["price_range"])
y = df["price_range"]

# Split dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)

# Initialize and train the XGBoost model
model = xgb.XGBClassifier(objective="multi:softmax", num_class=4, eval_metric="mlogloss")
model.fit(X_train, y_train)
```

MAKING PREDICTION

```
new_mobile = {
    "battery_power": 1200, "blue": 1, "clock_speed": 1.5, "dual_sim": 1, "fc": 3, "four_g": 1,
    "int_memory": 16, "m_dep": 0.6, "mobile_wt": 150, "n_cores": 2, "pc": 8,
    "px_area": 720 * 1080, "ram": 1500, "sc_h": 11, "sc_w": 5, "talk_time": 10, "three_g": 1,
    "touch_screen": 1, "wifi": 1
}

predicted_category = predict_price(new_mobile)
print(f"Predicted Price Category: {predicted_category}")
```

```
Predicted Price Category: Medium Cost
```

CONCLUSION

Model Performance Summary :

- XGBoost achieved 92.4% accuracy.

	precision	recall	f1-score	support
0	0.97	0.98	0.98	125
1	0.93	0.90	0.91	125
2	0.85	0.90	0.88	125
3	0.95	0.91	0.93	125
accuracy			0.92	500
macro avg	0.92	0.92	0.92	500
weighted avg	0.92	0.92	0.92	500

Classification report saved as 'classification_report.txt'

PROJECT – 2
LUNG CANCER

Exploratory Data Analysis And Machine Learning Model

Aman Varma
UNID – UMIP270026

INTRODUCTION

OBJECTIVE

- The goal of this project is to analyze a given dataset using **Exploratory Data Analysis (EDA)** and build a **Machine Learning model** to make predictions.
- We aim to **understand the dataset, clean the data, and extract insights** before applying a machine learning algorithm.
- Finally, we train a **Logistic Regression model** to classify the data and evaluate its performance

DATASET USED

- The dataset used in this project is **dataset_med.csv.**
- It contains information related to **patient medical history, health conditions, treatments, and survival** outcomes.
- It consists of **890,000 rows** and **48 columns**, including features such as **age, gender, BMI, cholesterol level, hypertension, asthma, cancer stage, smoking status,** and **treatment type** etc.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
id	age	gender	country	diagnosis_d	cancer_stage	family_history	smoking_status	bmi	cholesterol_level	hypertension	asthma	cirrhosis	other_cancer	treatment_type	end_treatment_date	survived
1	64	Male	Sweden	4/5/2016	Stage I	Yes	Passive Smoker	29.4	199	0	0	1	0	Chemotherapy	9/10/2017	0
2	50	Female	Netherlands	4/20/2023	Stage III	Yes	Passive Smoker	41.2	280	1	1	0	0	Surgery	6/17/2024	1
3	65	Female	Hungary	4/5/2023	Stage III	Yes	Former Smoker	44	268	1	1	0	0	Combined	4/9/2024	0
4	51	Female	Belgium	2/5/2016	Stage I	No	Passive Smoker	43	241	1	1	0	0	Chemotherapy	4/23/2017	0
5	37	Male	Luxembourg	#########	Stage I	No	Passive Smoker	19.7	178	0	0	0	0	Combined	1/8/2025	0
6	50	Male	Italy	1/2/2023	Stage I	No	Never Smoked	37.6	274	1	0	0	0	Radiation	12/27/2024	0

DATA EXPLORATION

Dataset Loading & Overview

Processed Dataset after performing EDA & Feature Engineering :

- A preview of the dataset: (df.head())

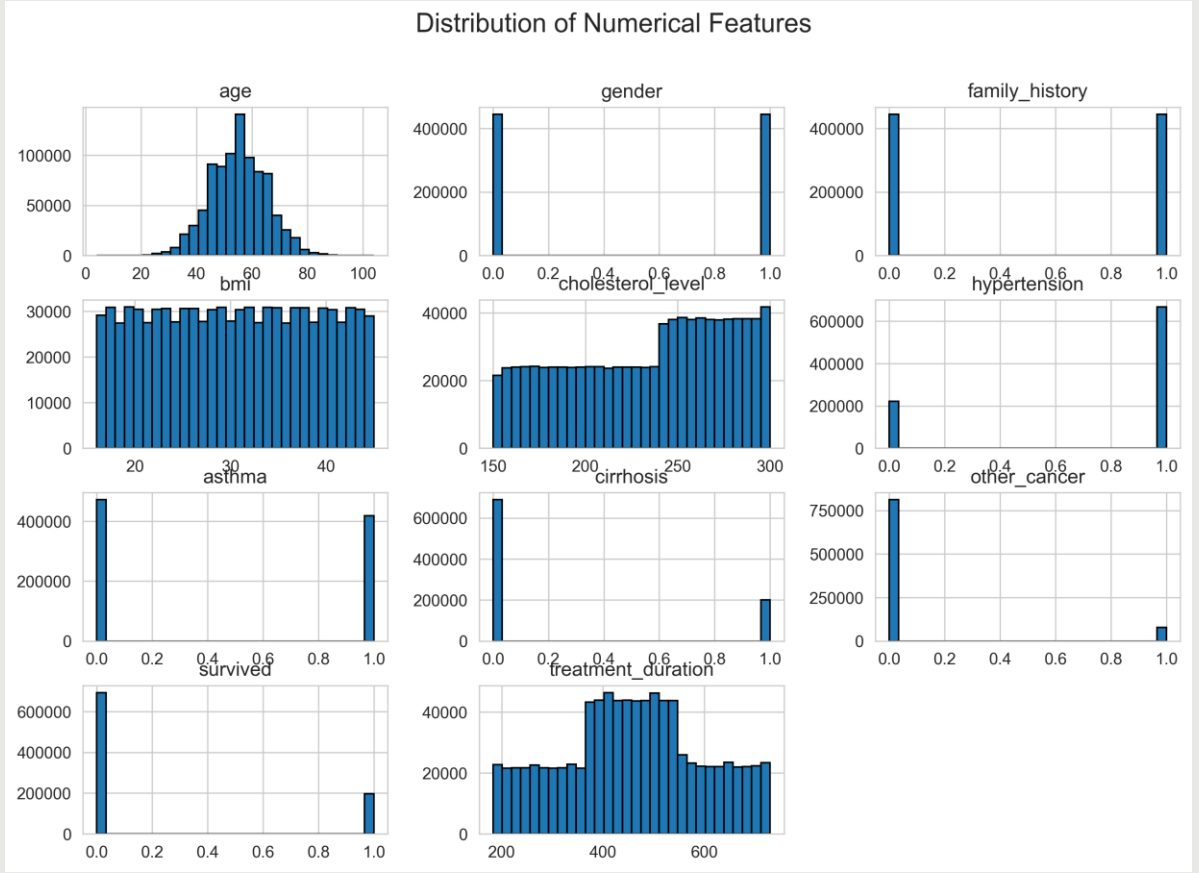
age	gender	country	diagnosis_date	cancer_stage	family_history	smoking_status	bmi	cholesterol_level	hypertension	asthma	cirrhosis
64.0	Male	Sweden	2016-04-05	Stage I	Yes	Passive Smoker	29.4	199	0	0	1
50.0	Female	Netherlands	2023-04-20	Stage III	Yes	Passive Smoker	41.2	280	1	1	0
65.0	Female	Hungary	2023-04-05	Stage III	Yes	Former Smoker	44.0	268	1	1	0
51.0	Female	Belgium	2016-02-05	Stage I	No	Passive Smoker	43.0	241	1	1	0
37.0	Male	Luxembourg	2023-11-29	Stage I	No	Passive Smoker	19.7	178	0	0	0

This shows the first few records and helps us understand the structure.

DATASET INFORMATION (df.info())

#	Column	Non-Null Count	Dtype
0	age	890000 non-null	float64
1	gender	890000 non-null	int64
2	diagnosis_date	890000 non-null	object
3	family_history	890000 non-null	int64
4	bmi	890000 non-null	float64
5	cholesterol_level	890000 non-null	int64
6	hypertension	890000 non-null	int64
7	asthma	890000 non-null	int64
8	cirrhosis	890000 non-null	int64
9	other_cancer	890000 non-null	int64
10	end_treatment_date	890000 non-null	object
11	survived	890000 non-null	int64
12	treatment_duration	890000 non-null	int64
13	cancer_stage_Stage II	890000 non-null	bool
14	cancer_stage_Stage III	890000 non-null	bool
15	cancer_stage_Stage IV	890000 non-null	bool
16	smoking_status_Former Smoker	890000 non-null	bool
17	smoking_status_Never Smoked	890000 non-null	bool
18	smoking_status_Passive Smoker	890000 non-null	bool
19	treatment_type_Combined	890000 non-null	bool
...			
46	country_Spain	890000 non-null	bool
47	country_Sweden	890000 non-null	bool

HISTOGRAM



MACHINE LEARNING MODEL

Data Preprocessing :

Train-Test Split (train_test_split).

- The dataset was divided into **80% training** and **20% testing**.
- This helps the model **learn patterns** from the training set and **generalize to unseen data**.
- **train_test_split()** from **sklearn.model_selection** was used for splitting.

Feature Scaling (StandardScaler)

- **StandardScaler** normalized numerical features to ensure a uniform scale, improving model stability and efficiency.

Model Selection :

- Chose **Logistic Regression** for binary classification due to its efficiency.
- Uses a **sigmoid function** to estimate class probabilities

MACHINE LEARNING MODEL

Model Training :

- Trained the model using **LogisticRegression()** from **sklearn**.
- Learned patterns from training data and made predictions on the test set.

Model Evaluation :

- **Accuracy Score** measured overall model performance.
- **Classification Report** provided key metrics:
  - **Precision** (True Positives vs. False Positives)
  - **Recall** (Sensitivity to positive cases)
  - **F1-score** (Balance of precision & recall)
  - **Support** (Instances per class)

CONCLUSION

Model Performance Summary :

- The **Logistic Regression model** was trained on the dataset.
- **Accuracy Score:** [0.78]
- **Classification Report:** Show key metrics (Precision, Recall, F1-score)
- Model was evaluated using a **train-test split (80-20%)**, ensuring fair evaluation.

```
Logistic Regression Model:
Accuracy: 0.78
Classification Report:
              precision    recall  f1-score   support

           0       0.78      1.00      0.88    138694
           1       0.00      0.00      0.00     39306

    accuracy                           0.78    178000
   macro avg       0.39      0.50      0.44    178000
weighted avg       0.61      0.78      0.68    178000
```

PROJECT – 3
ANIMAL CLASSIFICATION

Deep Learning-Based Image Classification

Aman Varma
UNID – UMIP270026

PROJECT OVERVIEW

- This project aims to classify animal images into **15 categories** using deep learning.
- Uses **EfficientNetB0**, a state-of-the-art convolutional neural network (CNN), for **transfer learning**.
- **Transfer Learning :**
  - *Pretrained on the **ImageNet** dataset.*
  - *Learns general image features and adapts to our dataset.*
- **Dataset Processing & Augmentation:**
  - **Normalization:** Pixel values scaled to **[0,1]**.
  - **Augmentations for better generalization:**
    - **Rotation:** Random rotations up to **±50°**.
    - **Shifts:** Horizontal & vertical shifts up to **30%**.
    - **Shearing & Zooming:** Up to **30%**.
    - **Horizontal Flipping:** For symmetry-based learning
- **Data Split:**
  - *80% Training Set*
  - *20% Validation Set*

DATA PREPARATION

- Data Organization:
  - Each animal category has its own **subdirectory**.
  - Example structure :
    - **dataset/Lion/** (contains lion images)
    - **dataset/Tiger/** (contains tiger images)
    - **dataset/elephant/** (contains elephant images)
- Data Splitting :
  - **Training Set (80%)**: Used to train the model.
  - **Validation Set (20%)**: Used to fine-tune model hyperparameters.
- Image Format & Size :
  - Images resized to **224x224 pixels**.
  - Supported formats: **JPG, PNG**.
- Challenges :
  - Class imbalance can affect model performance.
  - Ensuring diverse samples for better generalization.

DATA PREPROCESSING & AUGMENTATION

- Why Preprocessing?
  - Improves model performance and generalization..
  - Helps prevent overfitting by artificially expanding the dataset.
- Rescaling:
  - Pixel values normalized to **[0,1]** for consistent input representation.
- Augmentation Techniques:
  - **Rotation: Random rotations up to ±50° to make the model rotation-invarian**t.
  - **Width & Height Shift:** Random shifts up to **30%** to improve spatial robustness.
  - **Shearing & Zooming:**
    - Shearing by **±30%** distorts the image slightly to introduce variation
    - Zooming in/out by **±30%** to simulate different camera distances.
- Impact:
  - Improves model generalization to unseen images.
  - Enhances robustness against variations in image orientation and scale
  - Increases dataset size artificially.

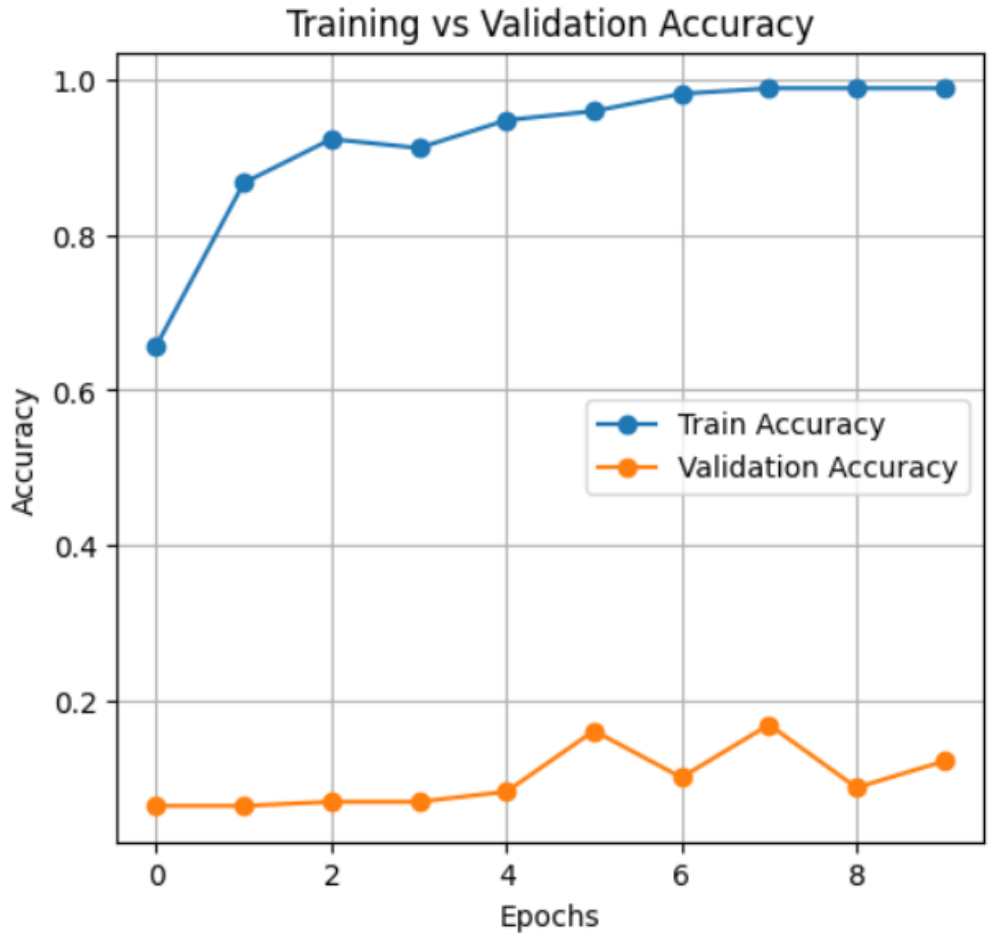
MODEL ARCHITECTURE

- Base Model: EfficientNetB0
  - **Pretrained on ImageNet** for feature extraction.
  - Provides **high accuracy** with fewer parameters.
- Fine-Tuning Strategy:
  - **Unfreezing the last 20 layers** allows the model to learn dataset-specific patterns.
  - The initial layers remain frozen to retain pre-trained knowledge.
- Output:
  - Produces **multi-class classification probabilities** for each input image.

TRAINING CONFIGURATION & MODEL TRAINING

- Training Configuration:
  - Optimizer: Adam (learning rate = 0.001)
  - Loss Function: Categorical Crossentropy
  - . Metrics: Accuracy
    - *Measures how well the model classifies images correctly.*
  - Learning Rate Scheduler :
    - *Automatically reduces the learning rate if validation loss does not improve after a few epochs.*
    - *Helps in stabilizing training and achieving better convergence.*
- Model Training:
  - Epochs: 50 .
    - *Number of times the entire dataset passes through the model.*
  - Batch Size: 32.
    - *Number of images processed before updating weights.*
  - Training & Validation Accuracy :
    - *Accuracy is tracked across epochs to identify overfitting or underfitting.*

RESULTS


Training vs Validation Accuracy
Accuracy
Epochs
Train Accuracy
Validation Accuracy
1.0
0.8
0.6
0.4
0.2
0
2
4
6
8


Training vs Validation Loss
Loss
Epochs
Train Loss
Validation Loss
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
2
4
6
8

THANK YOU

Aman Varma

955-506-3197 | amanvarma0486@gmail.com