Data Science Capstone Project Report: Tire Fault Detection

1. Project Title

Tire Fault Detection Using Machine Learning

2. Abstract

This project aims to develop a machine learning model to detect faults in tires by analyzing images of both good and defective tires. The project involves data collection, preprocessing, feature engineering, model training, evaluation, and deployment. The final model will assist in automating quality checks in tire manufacturing and maintenance processes.

3. Introduction

The quality of tires is crucial for vehicle safety and performance. This project addresses the need for an automated system to identify faulty tires using image classification techniques. The primary objectives include:

- Developing a robust image classification model.
- Evaluating various machine learning algorithms.
- Providing insights into the model's interpretability and performance.

4. Dataset Description

The dataset consists of images of good and faulty tires collected from various sources. It includes:

- **Good Tires**: Images representing tires in optimal condition.
- **Faulty Tires**: Images depicting various defects such as punctures, cracks, and uneven wear.

Dataset Attributes:

- **Image Files**: JPEG/PNG format.
- **Labels**: Binary classification (0 for good tires, 1 for faulty tires).

5. Data Exploration and Analysis

Conducted thorough exploratory data analysis (EDA) to understand the dataset's characteristics:

- **Descriptive Statistics**: Summary statistics of image counts per class.
- **Visualization**: Plots showing the distribution of good vs. faulty tire images.

Key Findings:

- The dataset is imbalanced, with more good tire images than faulty ones.
- Visual inspection reveals variability in tire conditions among faulty images.

6. Data Preprocessing

Data preprocessing steps included:

- **Data Cleaning**: Removed corrupted or unreadable images.
- **Normalization**: Scaled pixel values to the range [0, 1].
- **Augmentation**: Applied transformations such as rotation, width/height shifts, and horizontal flips to increase dataset diversity.

7. Feature Engineering

Feature engineering involved creating new features that could enhance model performance:

- No explicit feature engineering was needed since the model relies on raw pixel data from images.

8. Model Selection

Selected machine learning algorithms based on their suitability for image classification:

- **Convolutional Neural Networks (CNN)**: For deep learning-based image classification.
- **Logistic Regression**: As a baseline model for binary classification.
- **Decision Tree Classifier**: For its interpretability.
- **Random Forest Classifier**: To improve accuracy through ensemble learning.

9. Model Training and Evaluation

Training Process:

- Split the dataset into training (80%) and testing (20%) sets.
- Trained models using Keras Tuner to optimize hyperparameters for the CNN model.

Evaluation Metrics:

Used accuracy, precision, recall, F1-score, and confusion matrices to evaluate model performance.

Results:

- **CNN Model Validation Accuracy**: 0.70%
- **Logistic Regression Accuracy**: 0.62%
- **Decision Tree Accuracy**: 0.65%
- **Random Forest Accuracy**: 0.73%

10. Model Interpretability

Analyzed feature importance and used visualization techniques to interpret the CNN model's predictions:

- Employed SHAP (SHapley Additive exPlanations) values to explain individual predictions.

11. Documentation

Provided comprehensive documentation throughout the project:

- Detailed README.md file explaining project objectives, setup instructions, data sources, preprocessing steps, model selection rationale, evaluation metrics, and results.

12. Data Visualization

Created visualizations to communicate insights effectively:

- Bar charts comparing model performance metrics.
- Confusion matrices illustrating misclassifications for each model.

13. Model Deployment

Discussed deployment strategies for integrating the trained model into real-world applications:

- Considered using Flask or FastAPI to create a REST API for serving predictions on new tire images.

14. Communication

Prepared presentations summarizing key findings and methodologies tailored for both technical and non-technical audiences.

15. Bonus Points Considerations

1. **Packaging Solution**: Organized the project files into a well-structured zip file with clear setup instructions in README.md.
2. **Interactive Components**: Explored options for interactive visualizations using tools like Tableau or Dash to enhance user experience.