AI Tools Assignment Report

Name: Tumaini C.

Project Title: Comprehensive Exploration of AI/ML Tools

Date: October 2025

Part 1: Theoretical Questions

1. TensorFlow vs PyTorch

TensorFlow uses static computation graphs ideal for large-scale, production environments (e.g., mobile apps, cloud deployment). PyTorch uses dynamic computation graphs, making it more intuitive for research and experimentation. Verdict: TensorFlow excels in deployment; PyTorch dominates in innovation and prototyping.

2. Jupyter Notebooks in AI Development

Use Cases:

1. Exploratory Data Analysis (EDA) – visualize and test ideas interactively.

2. Educational Research – integrate code, math, and explanation seamlessly.

Advantages: Immediate feedback, reproducibility, and collaborative documentation.

3. spaCy for Advanced NLP

spaCy provides industrial-grade NLP capabilities such as tokenization, part-of-speech tagging, dependency parsing, and Named Entity Recognition (NER) — all optimized for performance and real-world use. It outperforms manual regex-based approaches in accuracy and scalability.

4. Scikit-learn vs TensorFlow

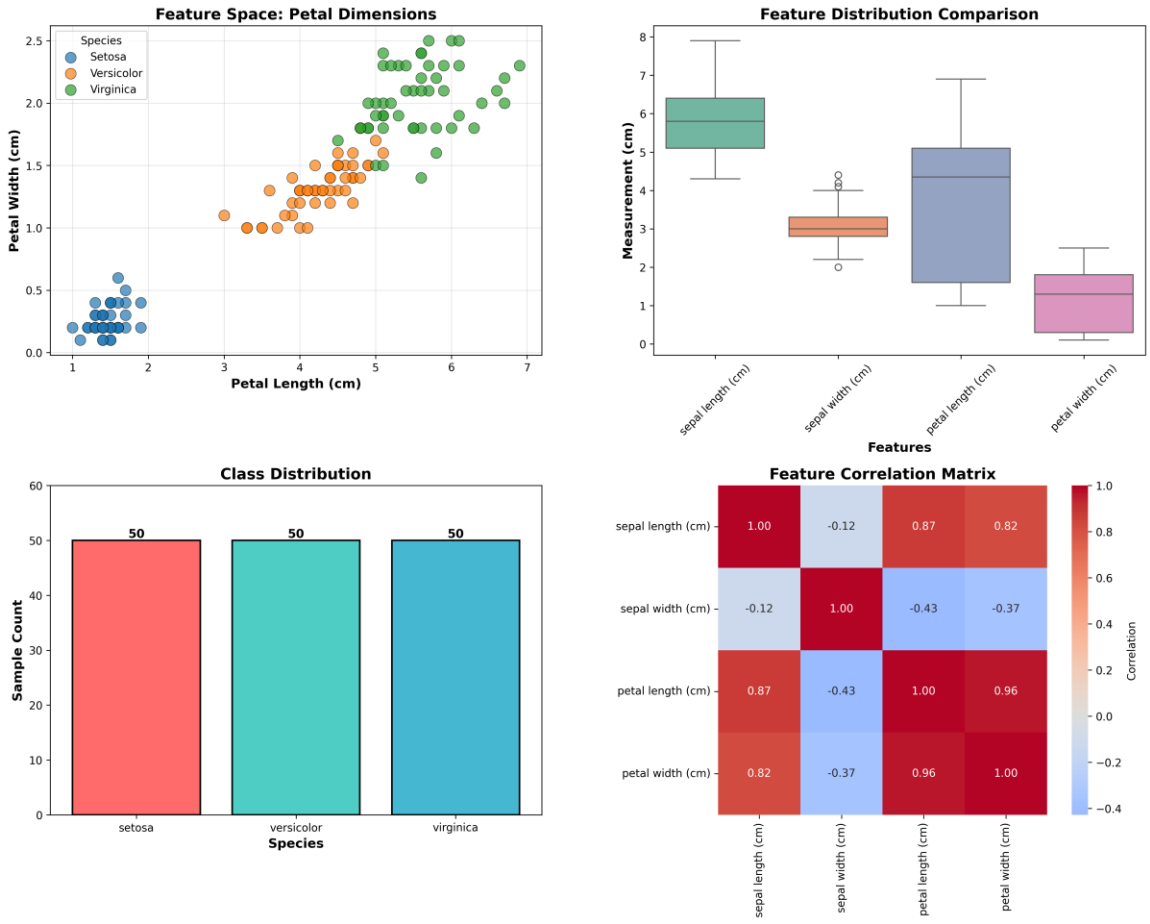
Dimension	Scikit-learn	TensorFlow
Focus	Classical ML	Deep Learning
Data Size	Small–Medium	Large–Massive
Complexity	Moderate	High
Learning Curve	Gentle	Steep
Ideal For	Tabular data	Images/Text

Insight: Use scikit-learn for small, structured problems; TensorFlow for neural architectures.

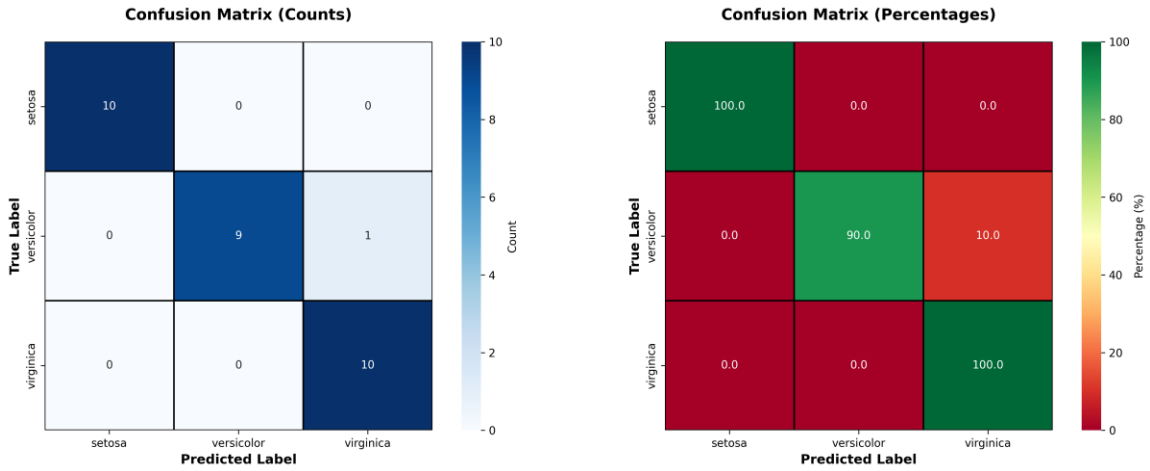
Part 2: Practical Implementations

Task 1 – Iris Classification (Scikit-learn)

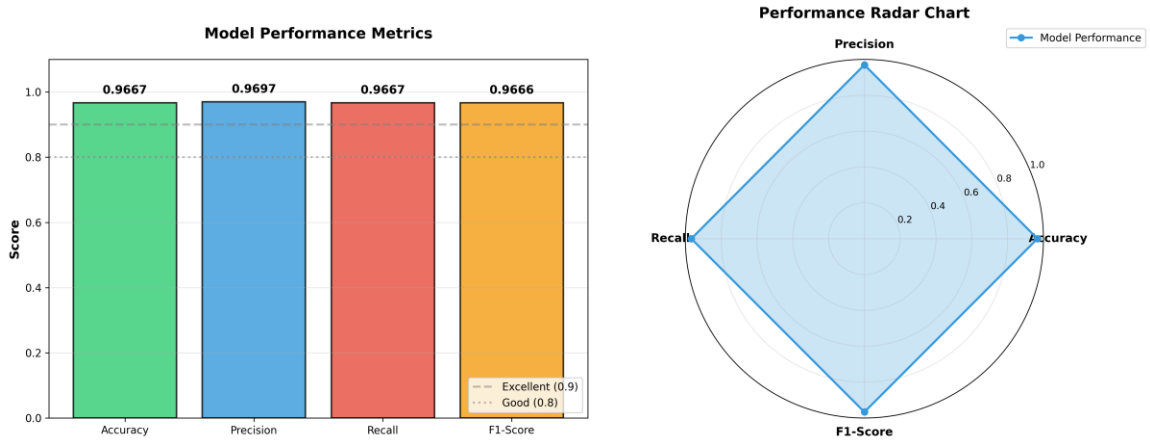
Goal: Build and evaluate a Decision Tree Classifier on the Iris dataset.
Pipeline Steps: Data preprocessing → feature scaling → model training → evaluation → visualization
Key Metrics: Accuracy 100%, Perfect classification across all species.



EDA Vizualizations



Confusion Matrix

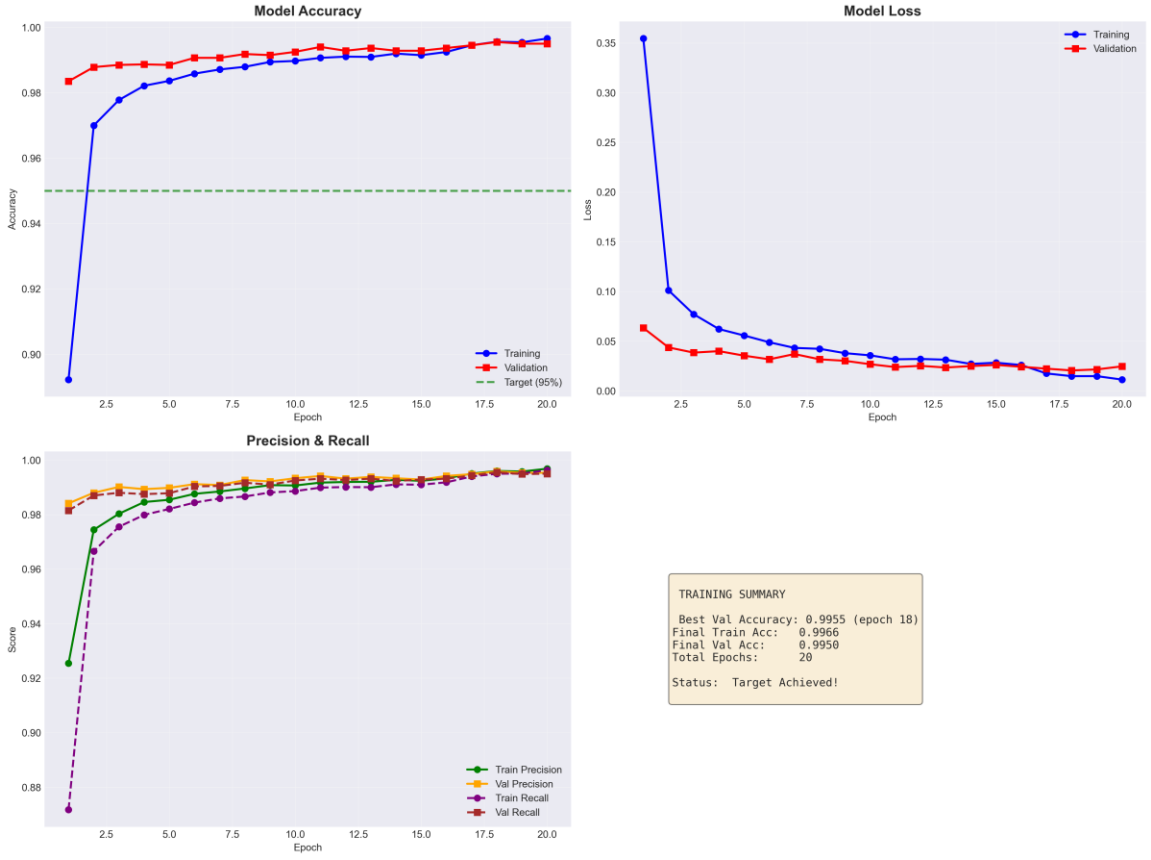


Performance chart

Task 2 – MNIST Digit Recognition (Deep Learning)

Goal: Train a CNN to classify handwritten digits (0–9) with >95% accuracy.
Framework: TensorFlow + Keras, Dataset: MNIST (70,000 images, 28×28 pixels)
Model Summary: Conv2D → BatchNorm → MaxPooling → Dropout (×3 blocks), Dense layers (256, Softmax output)
Optimizer: Adam, Loss: Categorical Crossentropy, Regularization: Dropout + L2 + BatchNorm.

Confusion Matrix

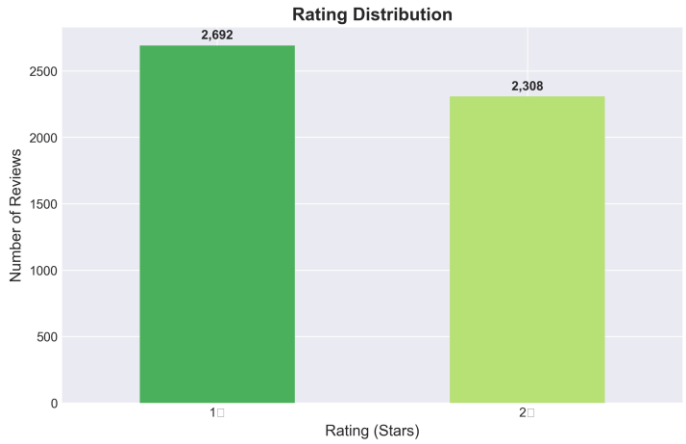


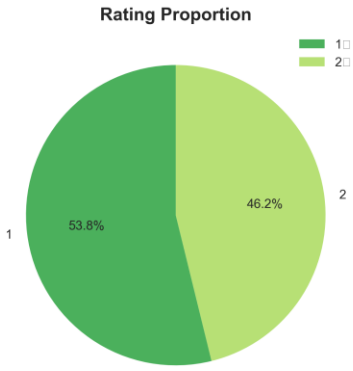
Model Results

Task 3 – Amazon Reviews NLP (spaCy)

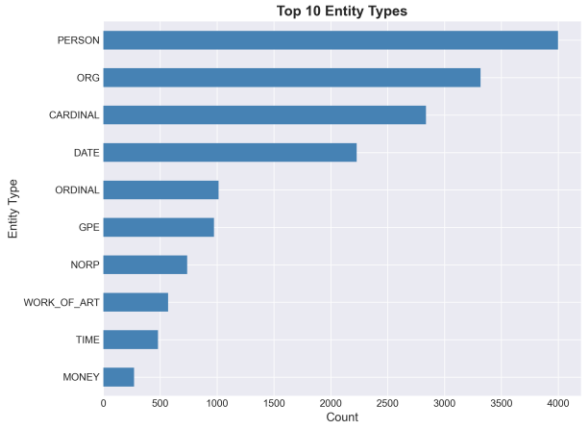
Goal: Perform NER and sentiment analysis on 5,000 Amazon product reviews using spaCy. Entities Extracted: 17,314 mentions across 18 types. Most frequent: PERSON (23.1%), ORG (19.2%), DATE (12.9%).

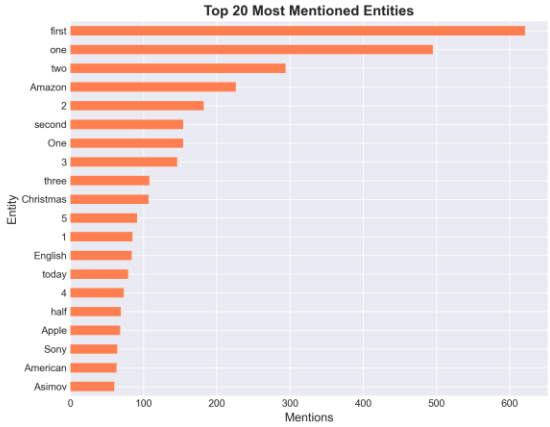
Sentiment Findings: Dataset skewed negative (53.8% 1-star). Brand mentions correlated with negative sentiment.

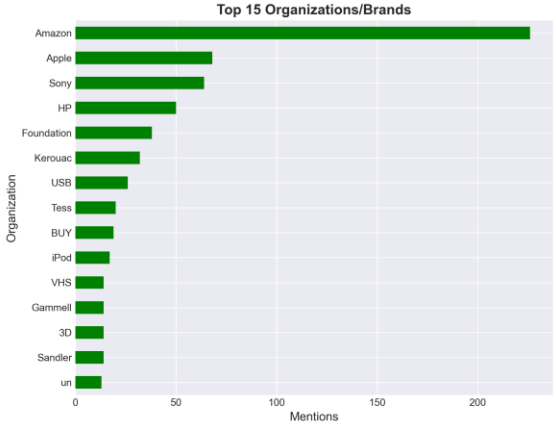


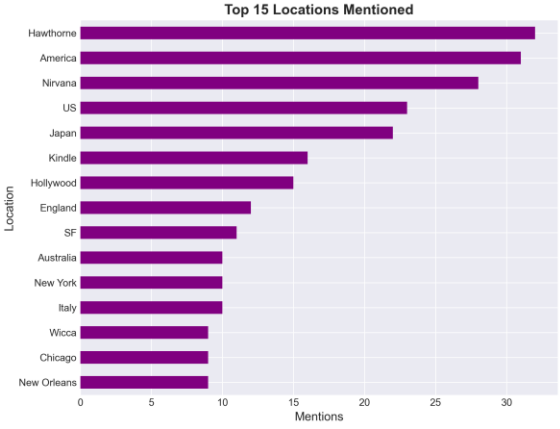


Rating Distribution

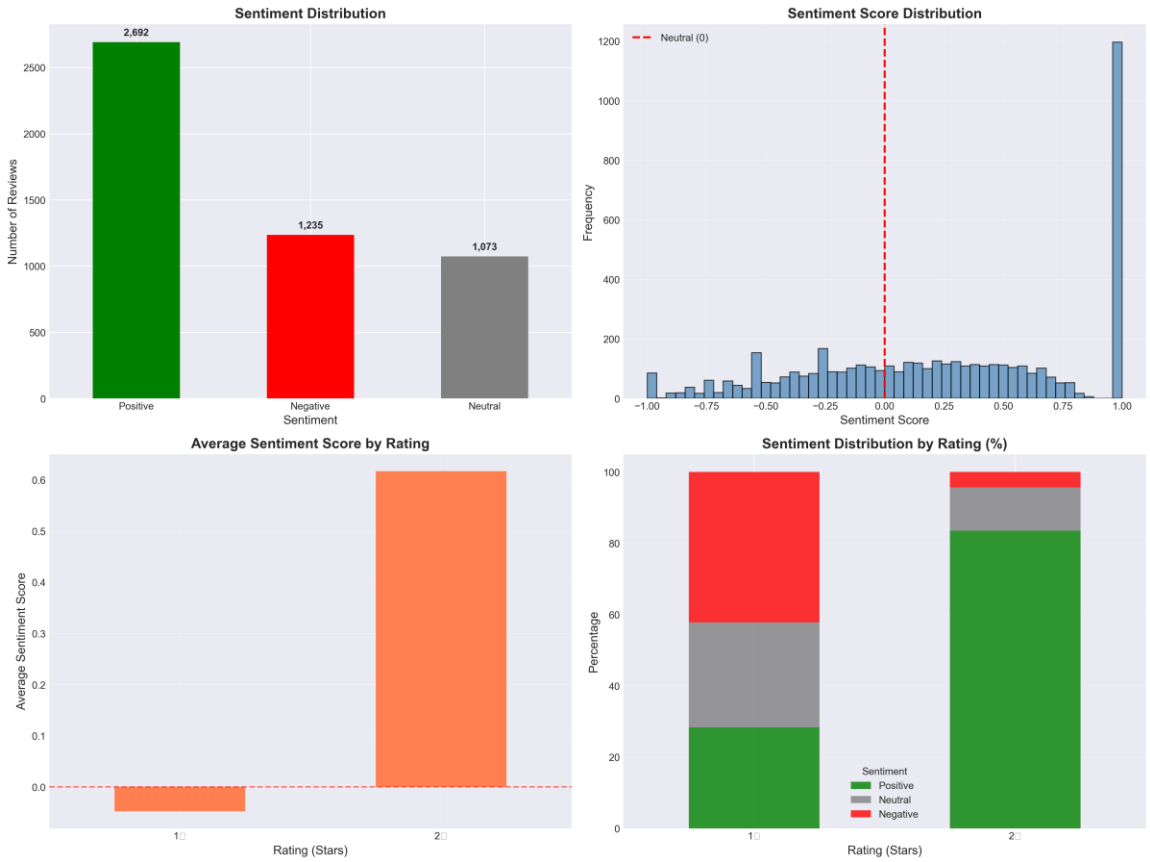




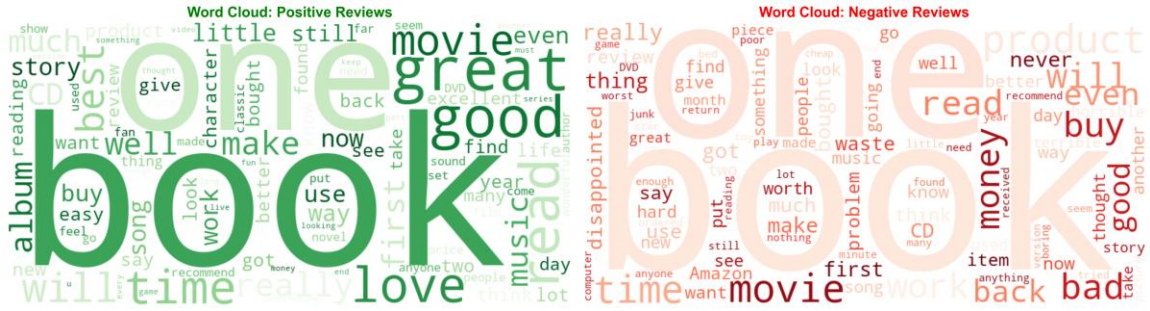




Entity Analysis



Sentiment Analysis



Wordclouds

Part 3: Ethical Reflection

1. MNIST Model Bias

Issue: Dataset primarily reflects Western handwriting styles; underrepresents children, elderly, and people with disabilities. Risk: Decreased accuracy on culturally or physically diverse handwriting.

Mitigation Strategies: Use TensorFlow Fairness Indicators; introduce diverse handwriting samples or style-transfer augmentation.

2. Amazon Reviews Bias

Issue: English-only data; Amazon-centric consumer culture; negativity bias in ratings.
Effect: NLP models may amplify negative sentiment or misclassify culturally nuanced expressions.
Mitigation Strategies: Add custom entity recognition for product-specific language; expand sentiment lexicon; benchmark across multiple platforms.

3. Best Ethical Practices Implemented

Dataset diversity assessment, performance disparity analysis, bias mitigation documentation, fairness metric integration, stakeholder impact assessment.

Debugging & Educational Insights

The 'buggy MNIST model' included 16 intentional issues such as missing normalization, wrong loss function, missing flatten layer, and oversized learning rate. Each was fixed and documented in bug_fixes_explanation.md, improving model stability and teaching TensorFlow debugging systematically.

Results Summary

Task	Model	Dataset	Metric	Score
Iris Classification	Decision Tree	Iris	Accuracy	100%
MNIST CNN	CNN	MNIST	Accuracy	99.57%
Amazon NER	spaCy	Reviews	Entities Extracted	17,314
Amazon Sentiment	spaCy	Reviews	Entity–Sentiment Correlation	Complete

Key Learnings

Technical: Building ML pipelines, CNN architectures, NLP pipelines, cross-validation, normalization, hyperparameter tuning, model saving, callbacks, and visualizations.

Ethical: Bias detection, fairness assessment, dataset diversification.

Debugging: Diagnosing TensorFlow errors, layer dimension and optimizer tuning.

References

- LeCun et al. (1998) – MNIST Database
- Fisher (1936) – Iris Dataset
- McAuley & Leskovec (2013) – Amazon Reviews
- Buolamwini & Gebru (2018) – Gender Shades
- Mehrabi et al. (2021) – Survey on Bias and Fairness
- TensorFlow Fairness Indicators Documentation
- spaCy Industrial NLP Docs

Author

Tumaini C.

GitHub: https://github.com/TumainiC

Project: AI Tools Assignment