ERICK WAMBUGU
aguerowaza@gmail.com

Course: AI For Software Engineering
Week: 3 Assignment

PART 1

THEORETCAL UNDERSTANDING:

1. Explain the primary difference between TensorFlow and PyTorch. When would you choose one over the other?

ANS: *TensorFlow and PyTorch are both popular tools for building deep learning models.*

TensorFlow: It was created by ***google*** *and was originally designed using* ***static computation graphs****. It is more focused on production and deployment since it has powerful tools like TensorFlow Serving, TensorFlow Lite and TensorFlow.js for the web. It is a strong choice for large organizations that need reliable and scalable systems. However it can feel complex for beginners and debugging can take more effort*

PyTorch: It was built by ***Meta {FaceBook}*** *and is more flexible and it uses* ***dynamic computational graphs****. It feels more like normal* ***python programming*** *which is why many students, researchers and developers prefer it. You can build and change your models while the program runs.*

I personally prefer PyTorch because it is simple, clear and great for learning and experimenting. It allows me to focus on understanding machine learning and artificial intelligence concepts instead of struggling with complicated syntax – its also dynamic in nature and it integrates well and easily with numpy, matplotlib and pandas

2. Describe two use cases for Jupyter Notebooks in AI Development

a) ***Data Exploration and Pre-processing*** *- they are ideal for exploring and cleaning datasets before training AI models.*

b) ***Model Development and Experimentation*** *– it is used for building and testing machine learning or deep learning models. One can write codes in steps – importing libraries, defining models, training and evaluating performance all while visualizing the metrics like accuracy or loss in real time*

3. How does spaCy enhance NLP tasks compared to basic Python string operations?

spaCy enhances Natural Language Processing tasks since it's an NLP library that understands the structure and meaning of language unlike python strings. It can automatically tokenize text, identify parts of speech, recognize named entities and even understand word relationships through dependency parsing. It makes NLP tasks faster, more accurate and much easier to implement because it handles complex linguistic details automatically, allowing developers focus on building intelligent language applications instead of manually processing texts.

ERICK WAMBUGU
aguerowaza@gmail.com

Course: AI For Software Engineering
Week: 3 Assignment

COMPARATIVE ANALYSIS:
Compare Scikit-learn and TensorFlow in terms of:
1. Target applications (e.g classical ML vs Deep Learning)

Scikit-learn is best suited for classical ML tasks and simpler models such as linear regression, decision trees, random forest, support vector machines and clustering algorithms like K-Means while TensorFlow is designed for deep learning and large, complex neural networks architectures like convolutional neural networks CNN for image recognition, recurrent neural networks RNN for text and speech and transformers for advances AI applications.

2. Ease of use for beginners

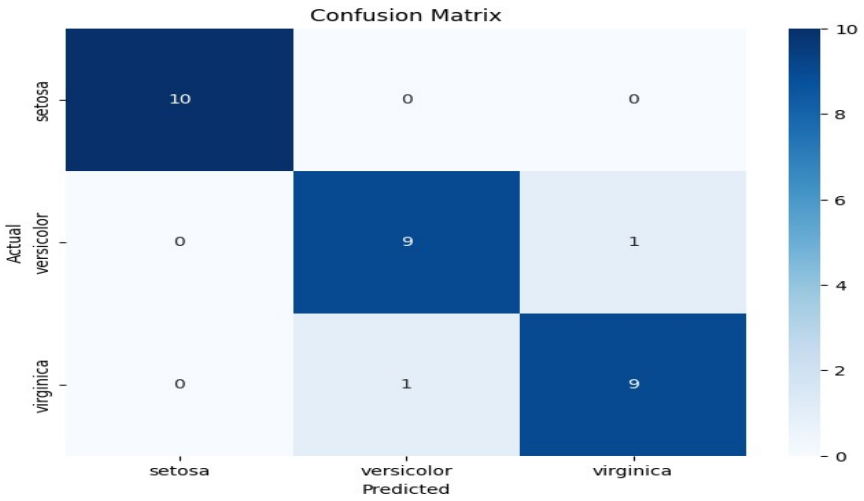
scikit-learn is simpler and more beginner-friendly for traditional ML tasks, while TensorFlow is more advanced and suited for learners who already have some background in machine learning or want to focus on deep learning

3. Community support

scikit-learn has excellent support for traditional ML learners and practitioners, while TensorFlow has a much broader and deeper community focused on deep learning, AI research and large-scale deployments

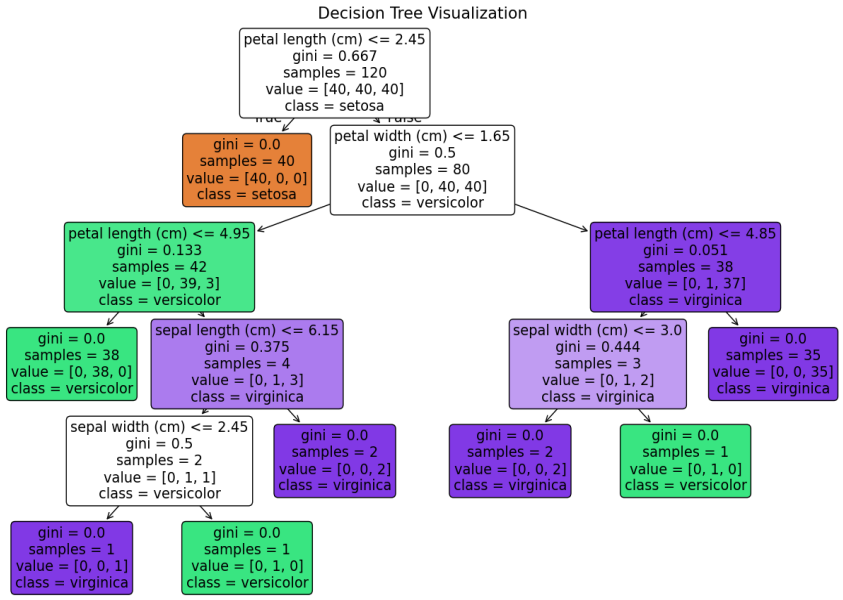
PART 2

Task 1: Classical ML with scikit-learn



ERICK WAMBUGU
aguerowaza@gmail.com

Course: AI For Software Engineering
Week: 3 Assignment



Task 2: Deep Learning with PyTorch

The CNN model had a 99.03% accuracy

Task 3: NLP with spaCy