Pham Duc Chinh B21DCCN181

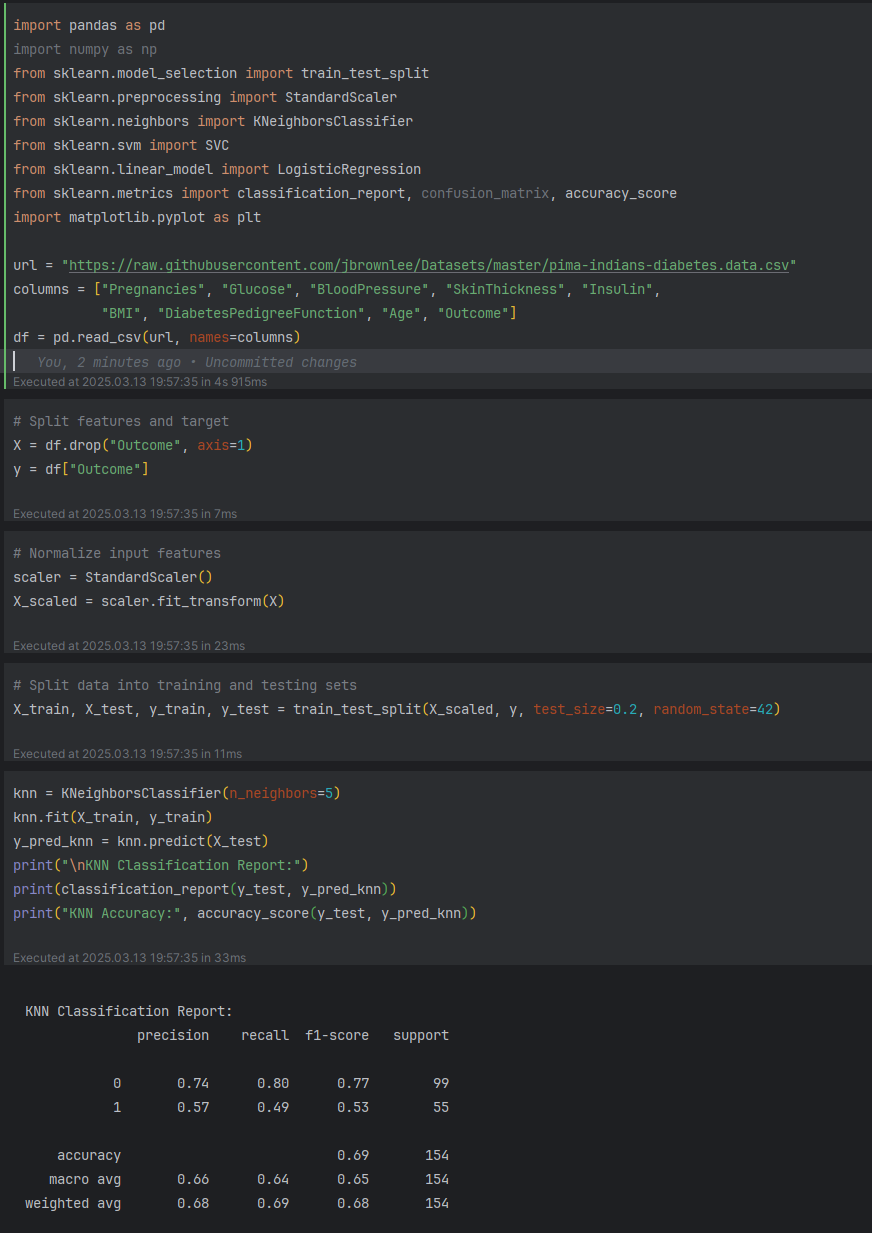
Developing techniques for predicting diabetes (refer to Chap 12, [1])

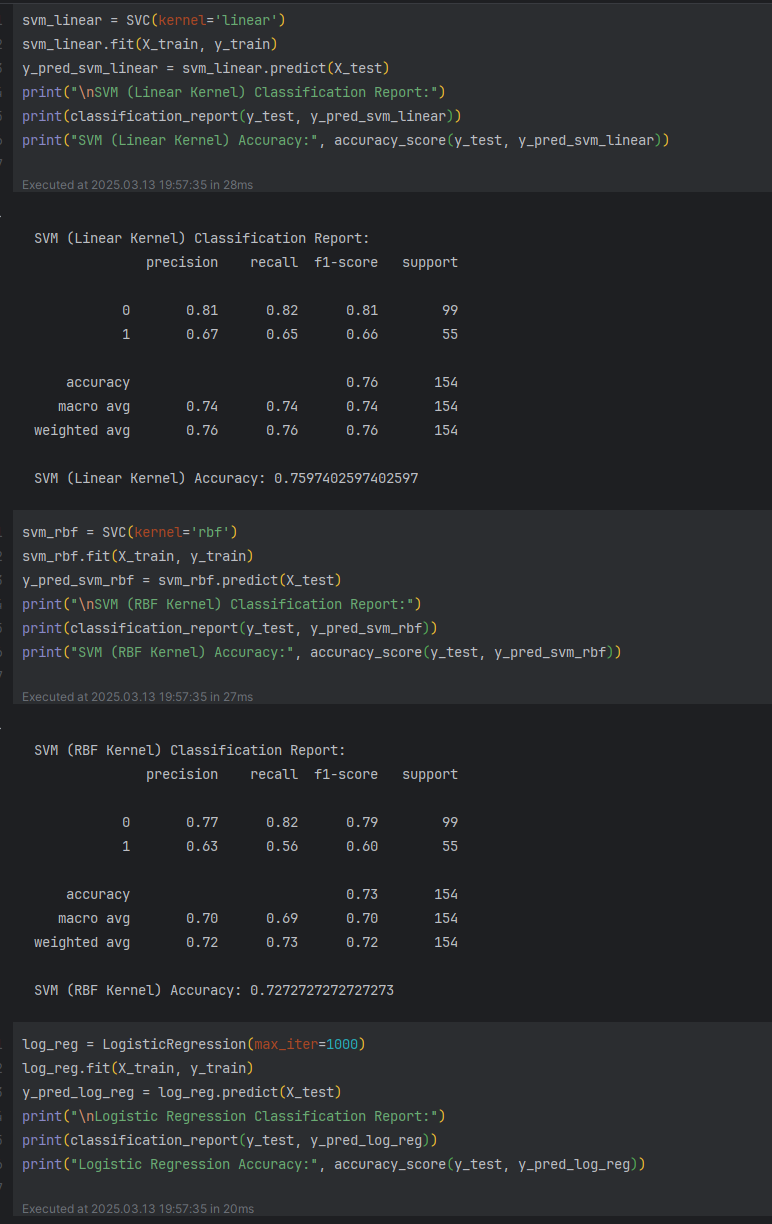
• kNN

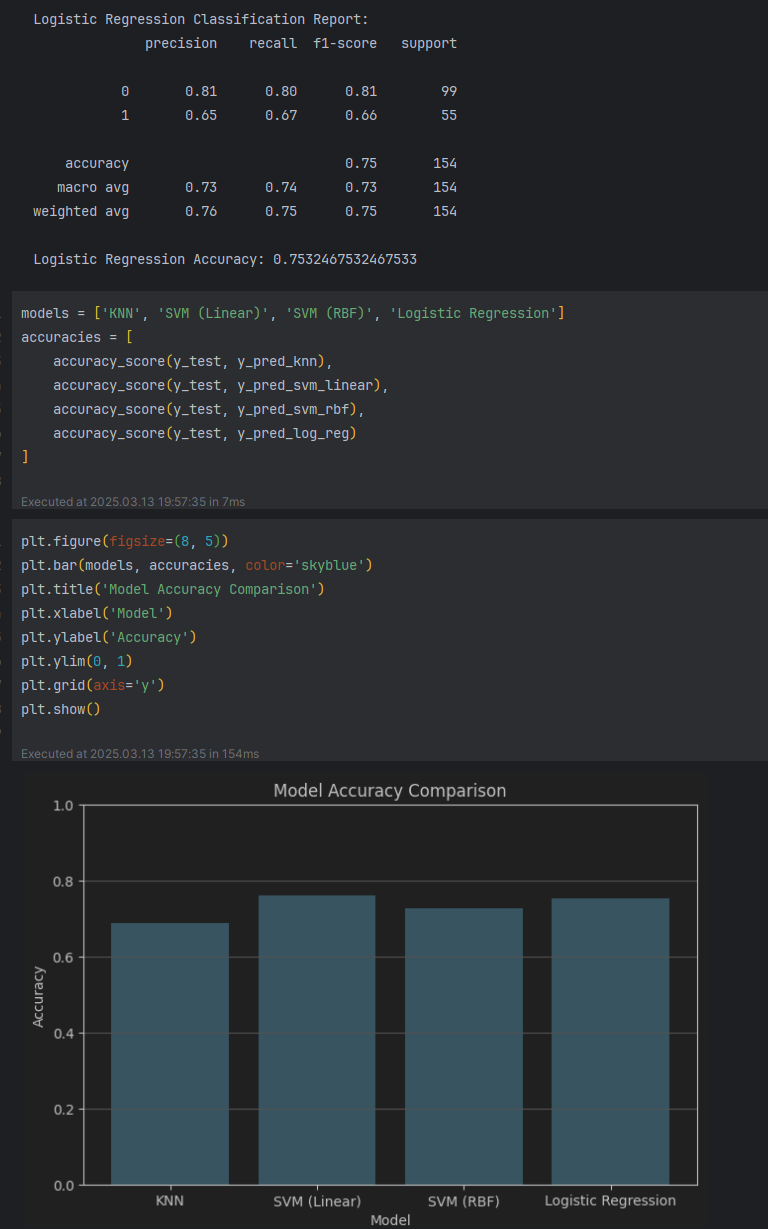
• SVM linear kernel

• SVM RBF kernel

• Logistic Regression

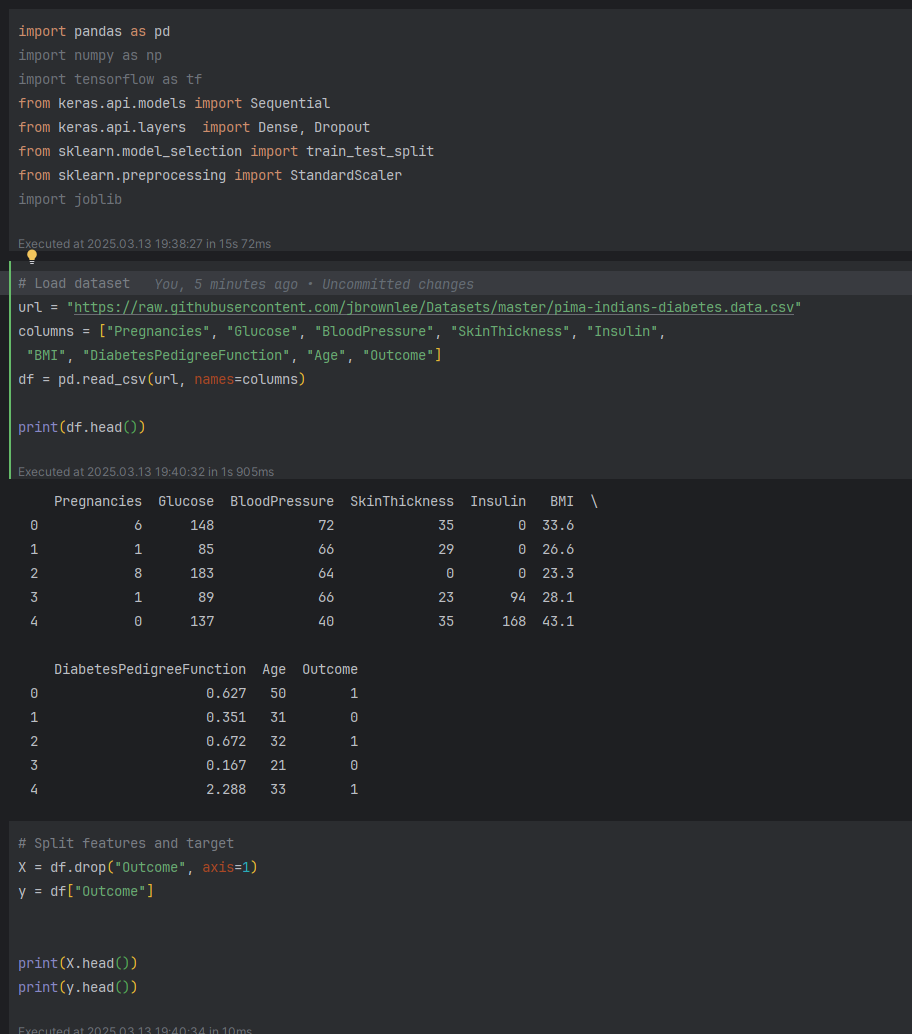


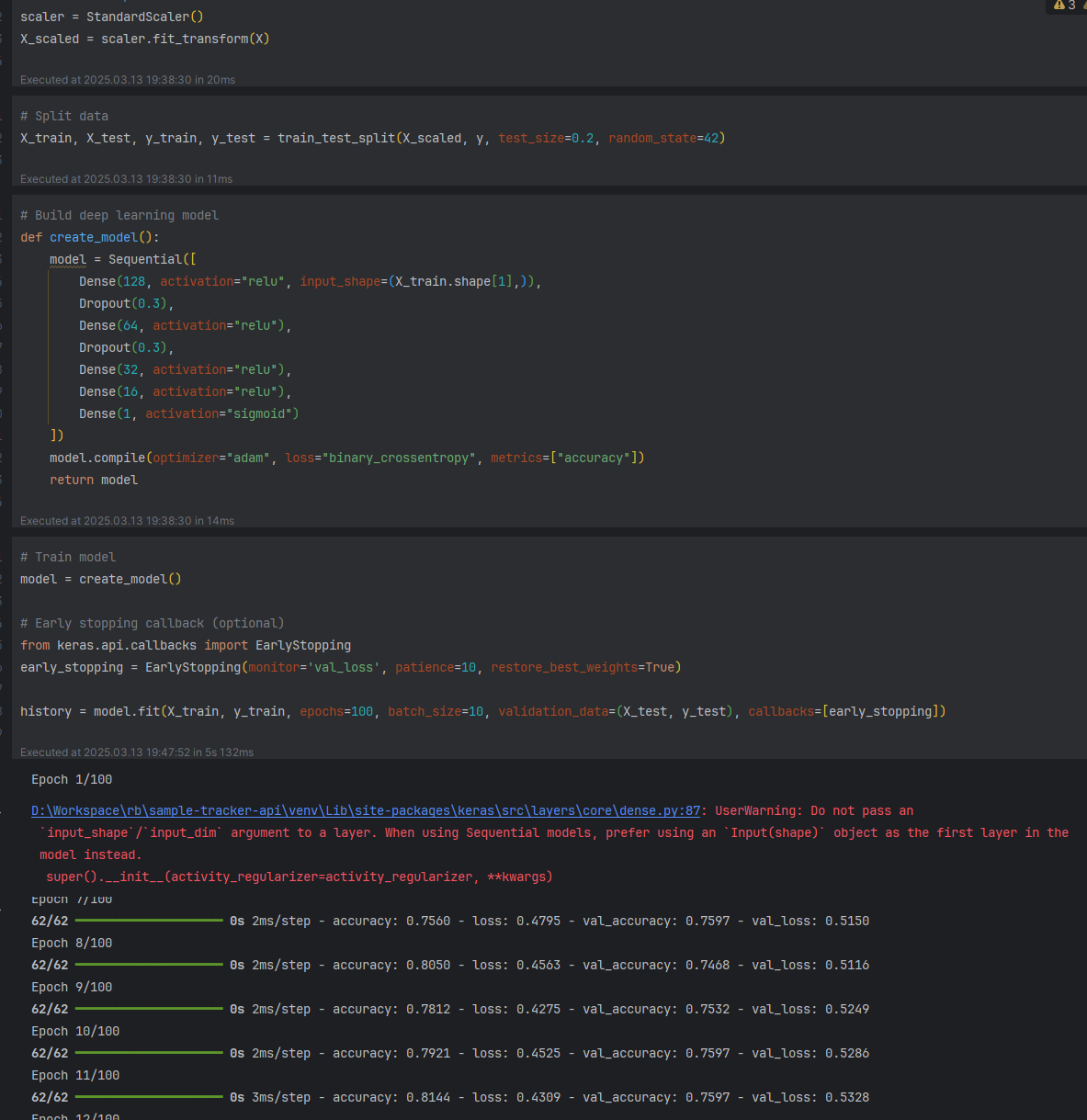


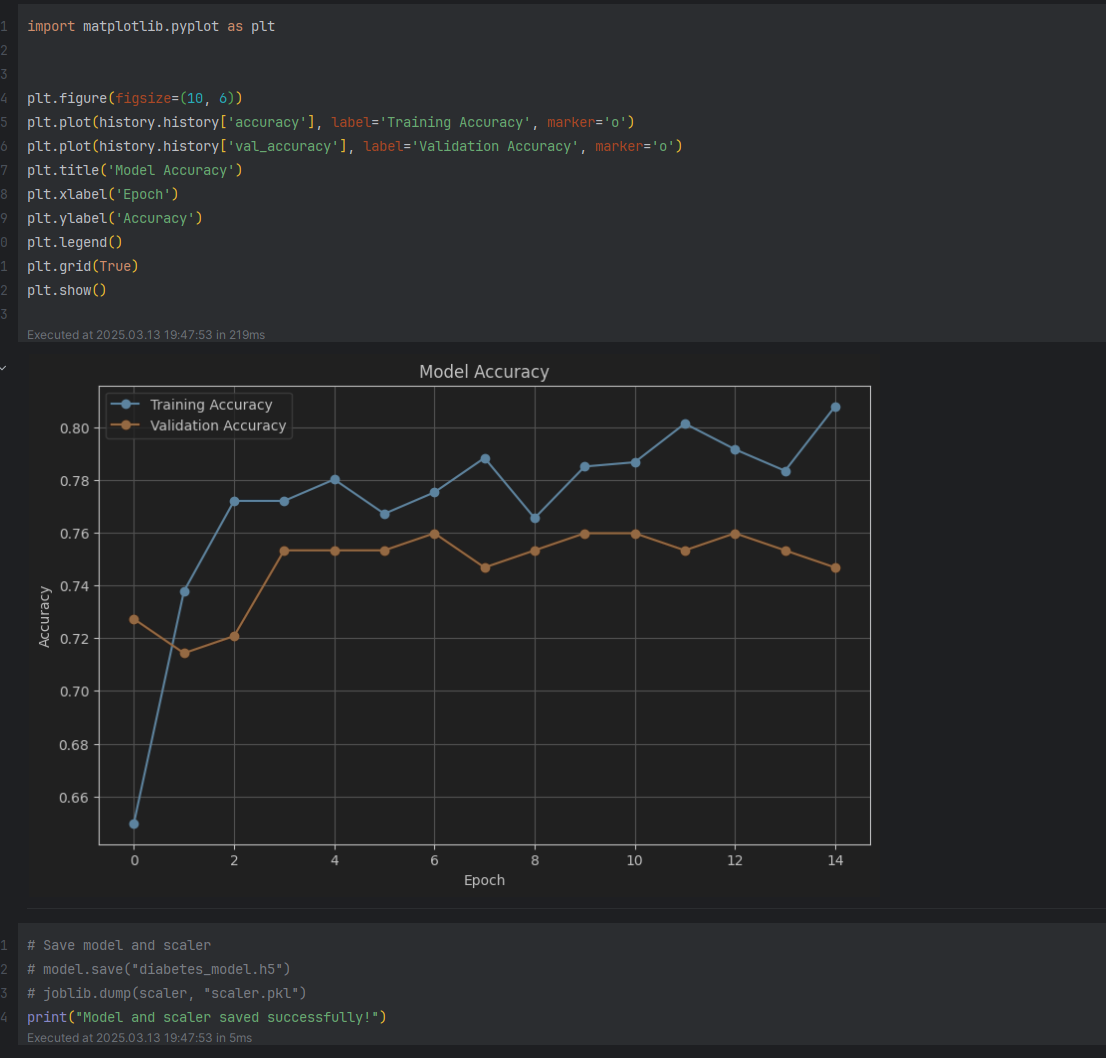


Deep Learning > 5 Layers

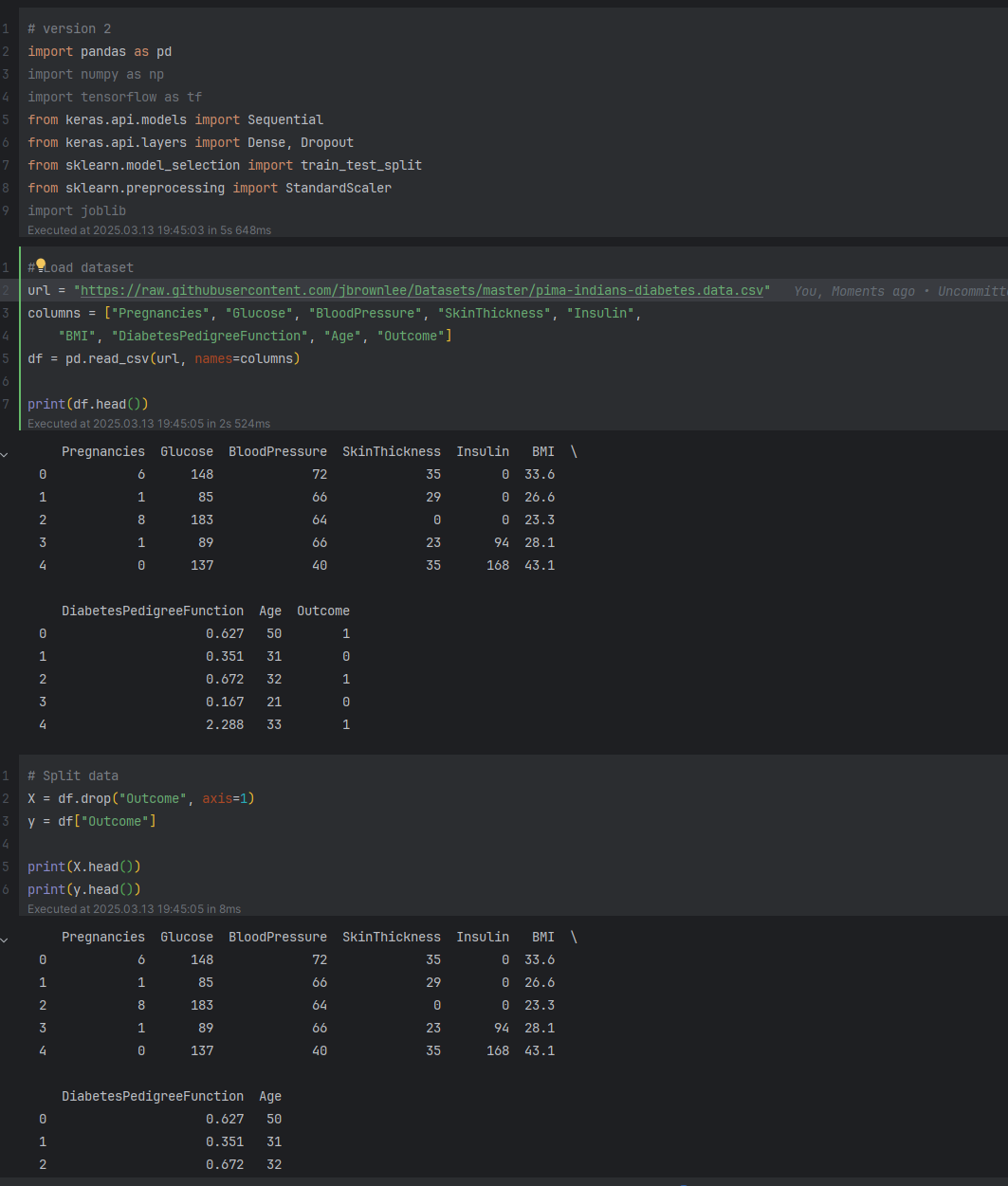
V1

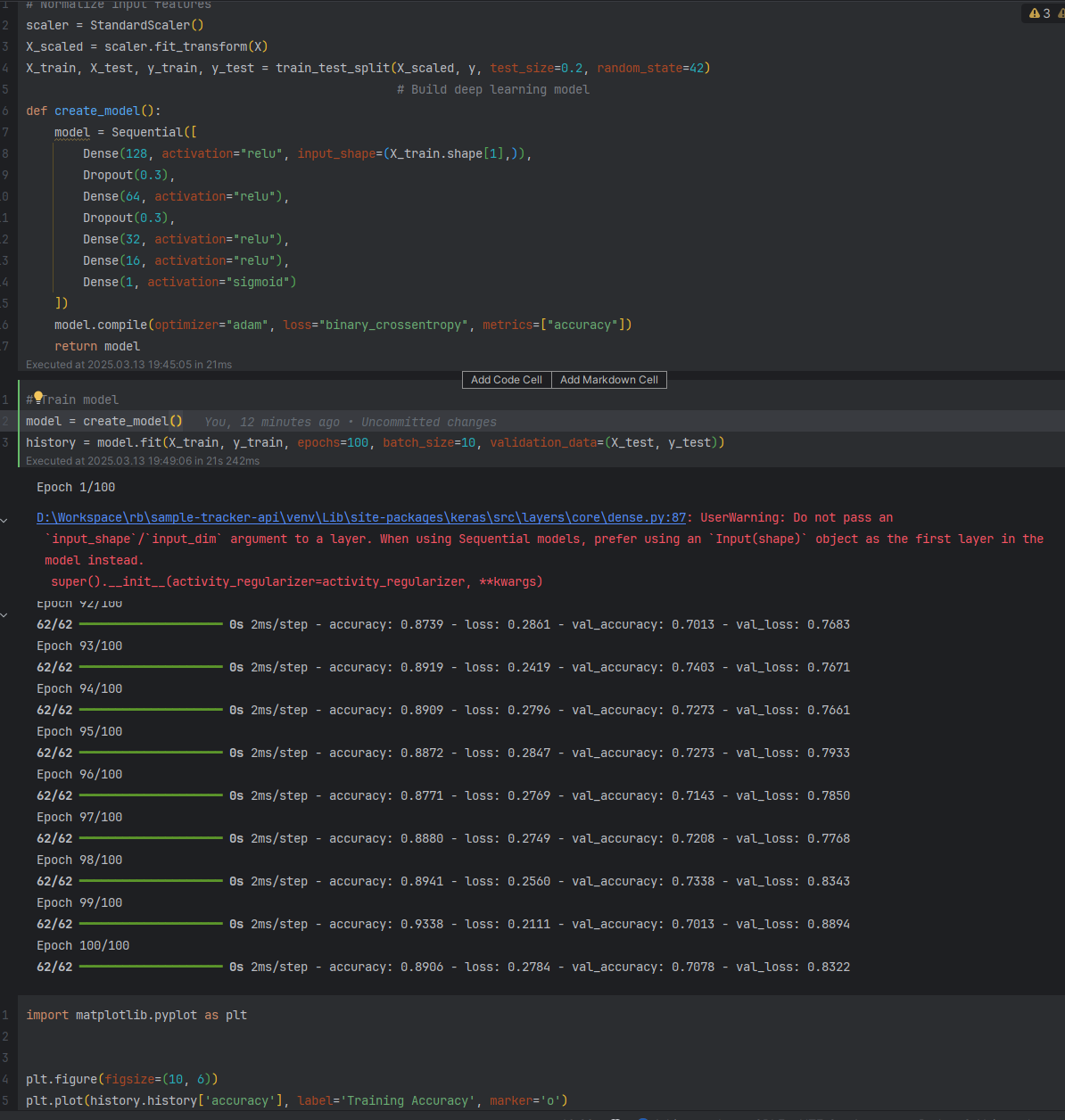






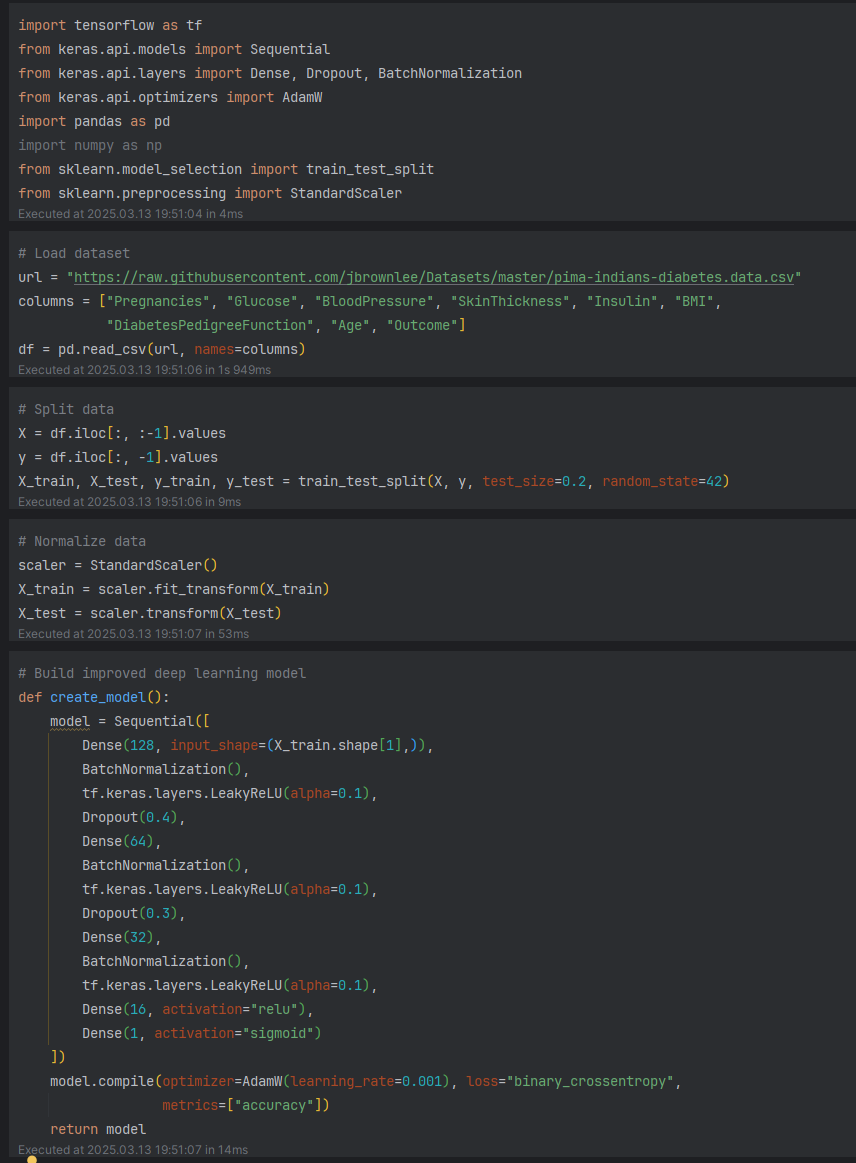
V2

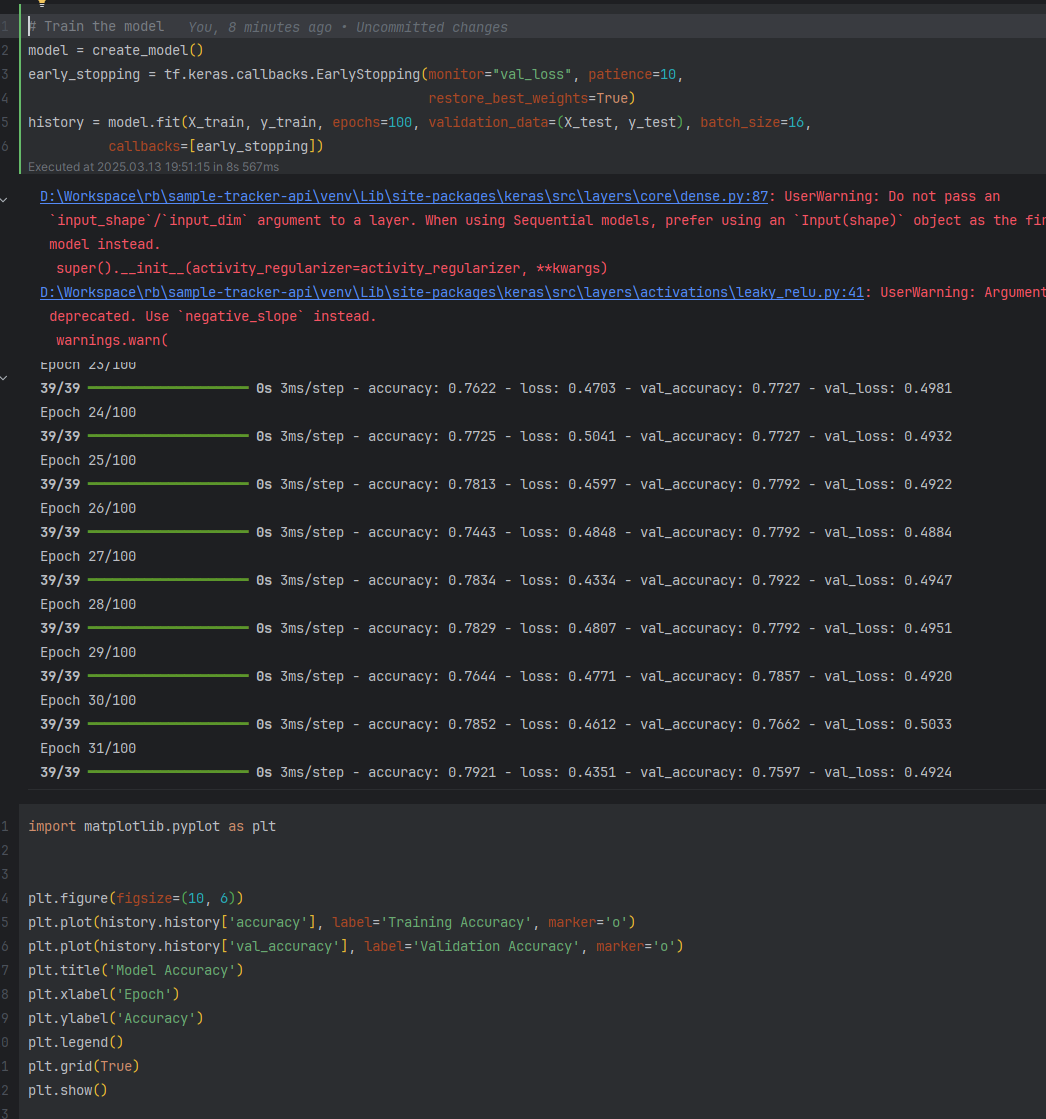


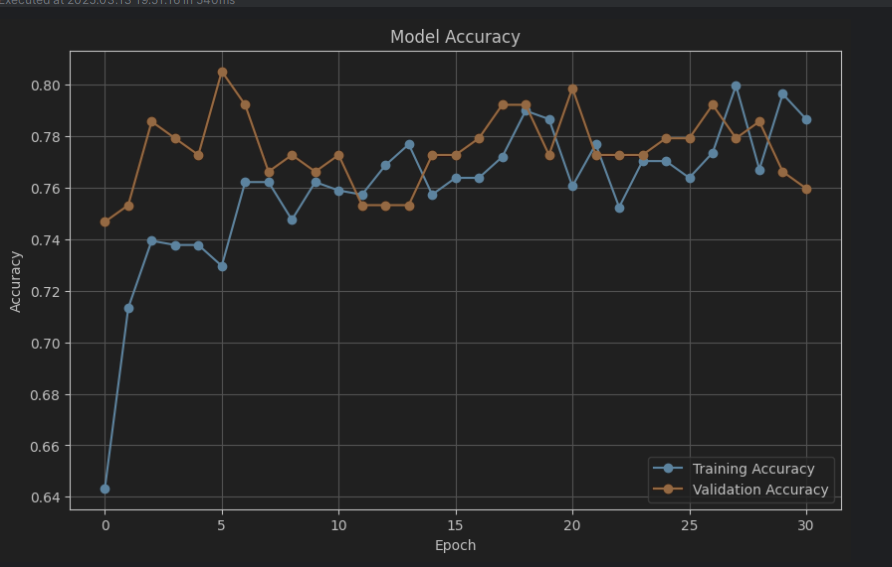




V3

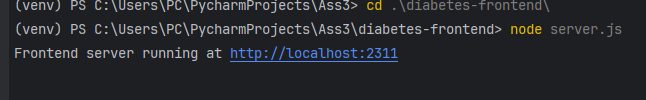




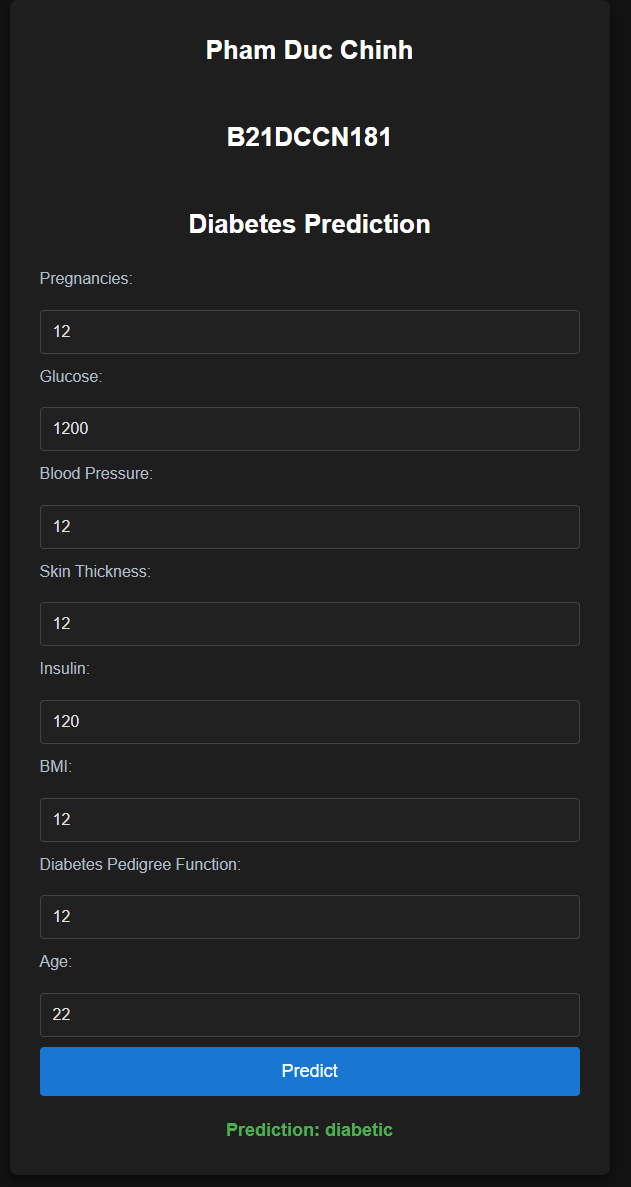


3. Deploy the models and use interface to allow users entering features, select model for showing the results  







Write 5 pages of deep learning

**Deep Learning**

**1. Introduction to Deep Learning**

**What is Deep Learning?**

Deep Learning is a subset of **Machine Learning (ML)** that focuses on using **artificial neural networks (ANN)** to model and solve complex problems. Unlike traditional ML methods that rely heavily on manual feature extraction, deep learning models can automatically discover and learn patterns from data through multi-layered neural networks.

Deep learning models are inspired by the structure and function of the human brain, where neurons process and transmit information. The key idea is that through multiple layers of computation, the model can extract high-level features from raw data, such as:

* Identifying objects in an image
* Understanding speech patterns
* Generating human-like text

**Why Deep Learning Matters**

Traditional ML methods (e.g., decision trees, SVMs) struggle with large-scale, high-dimensional data. Deep learning models, due to their ability to learn hierarchical representations, have shown significant improvements in:

* Image and video recognition
* Natural language processing (NLP)
* Speech recognition
* Autonomous systems (e.g., self-driving cars)

**Key Components of Deep Learning:**

1. **Neurons** – Inspired by biological neurons, these are the basic computational units.
2. **Weights and Biases** – Parameters adjusted during training to minimize error.
3. **Activation Functions** – Introduce non-linearity to the network (e.g., ReLU, Sigmoid).
4. **Layers** – Stacked layers of neurons to extract patterns:
   * **Input Layer** – Takes raw data as input.
   * **Hidden Layers** – Extracts hierarchical features.
   * **Output Layer** – Produces final predictions.

**Evolution of Deep Learning**

| **Year** | **Milestone** |
| --- | --- |
| 1943 | First mathematical model of neurons (McCulloch & Pitts) |
| 1958 | Perceptron model (Frank Rosenblatt) |
| 1980s | Backpropagation algorithm rediscovered |
| 2006 | Deep Belief Networks (Hinton) |
| 2012 | AlexNet wins ImageNet competition, marking DL's breakthrough |
| 2015+ | Explosion in AI applications (e.g., GPT, AlphaGo) |

**2. Neural Network Architectures**

**2.1. Artificial Neural Networks (ANN)**

An **Artificial Neural Network (ANN)** consists of interconnected layers of neurons. Each neuron processes weighted inputs and passes them through an activation function to the next layer.

**Structure of an ANN:**

* **Input Layer:** Takes input features (e.g., pixels in an image).
* **Hidden Layers:** Perform complex transformations using weighted sums and activation functions.
* **Output Layer:** Outputs the final prediction or classification.

**Mathematical Formulation:** For a single neuron:

y=f(∑i=1nwixi+b)y = f\left( \sum\_{i=1}^n w\_i x\_i + b \right)y=f(i=1∑n​wi​xi​+b)

where:

* yyy = output
* wiw\_iwi​ = weights
* xix\_ixi​ = input features
* bbb = bias
* fff = activation function

**2.2. Convolutional Neural Networks (CNN)**

CNNs are designed for **image and video processing**. They apply convolutional filters to detect patterns in local regions.

**Key Components:**

* **Convolutional Layer** – Applies filters to detect edges, textures, and patterns.
* **Pooling Layer** – Reduces dimensionality while preserving important features.
* **Fully Connected Layer** – Combines extracted features to produce the output.

✅ **Example:** CNNs are used in **image classification** (e.g., recognizing cats and dogs).

**2.3. Recurrent Neural Networks (RNN)**

RNNs are designed for **sequential data** (e.g., text, speech). They have loops that allow them to maintain information over time steps.

**Key Components:**

* **Hidden State** – Stores information from previous time steps.
* **Vanishing Gradient Problem** – RNNs struggle with long-term dependencies.

✅ **Solution:** Use **LSTM (Long Short-Term Memory)** or **GRU (Gated Recurrent Unit)** networks to capture long-term dependencies.

✅ **Example:** RNNs are used in **language translation** and **speech recognition**.

**2.4. Transformers**

Transformers are state-of-the-art models for handling sequential data without recurrence. They rely on **self-attention** mechanisms to understand context.

**Key Components:**

* **Self-Attention:** Weighs the importance of different parts of the input sequence.
* **Positional Encoding:** Provides order information to the model.

✅ **Example:** GPT (Generative Pre-trained Transformer) and BERT (Bidirectional Encoder Representations from Transformers).

**3. Training Deep Learning Models**

**3.1. Backpropagation**

Backpropagation is the key algorithm used to train neural networks by updating weights using the **gradient descent** method.

**Steps:**

1. **Forward Pass:** Compute the output using current weights.
2. **Loss Calculation:** Measure the error between predicted and actual output.
3. **Backward Pass:** Compute gradients using the chain rule.
4. **Weight Update:** Adjust weights using gradient descent.

**3.2. Loss Functions**

Loss functions measure the error between predicted and true values:

* **Binary Cross-Entropy:** For binary classification
* **Categorical Cross-Entropy:** For multi-class classification
* **Mean Squared Error (MSE):** For regression problems

**3.3. Optimizers**

Optimizers control how weights are updated:

* **SGD (Stochastic Gradient Descent):** Updates weights after each sample.
* **Adam (Adaptive Moment Estimation):** Combines momentum and adaptive learning rates.

**3.4. Regularization Techniques**

Regularization helps prevent overfitting:

* **Dropout:** Randomly disables neurons during training.
* **L2 Regularization:** Penalizes large weights to prevent complexity.

**4. Applications of Deep Learning**

**4.1. Computer Vision**

* Image classification (e.g., ResNet)
* Object detection (e.g., YOLO)
* Face recognition (e.g., FaceNet)

**4.2. Natural Language Processing (NLP)**

* Sentiment analysis
* Machine translation (e.g., Google Translate)
* Chatbots (e.g., GPT)

**4.3. Autonomous Systems**

* Self-driving cars (e.g., Tesla)
* Robotics (e.g., Boston Dynamics)

**4.4. Healthcare**

* Disease diagnosis (e.g., cancer detection)
* Drug discovery

**5. Challenges and Future of Deep Learning**

**5.1. Challenges**

1. **Data Dependence:** Requires large labeled datasets.
2. **Computational Cost:** Training deep models demands significant GPU/TPU power.
3. **Interpretability:** Difficult to understand how deep models make decisions.
4. **Overfitting:** High complexity increases the risk of overfitting.

**5.2. Solutions and Future Directions**

1. **Transfer Learning:** Use pre-trained models to reduce data dependence.
2. **Federated Learning:** Train models without sharing data (privacy-focused).
3. **Explainable AI (XAI):** Improve model transparency and interpretability.
4. **Energy Efficiency:** Develop more efficient architectures (e.g., pruning).

**6. Conclusion**

Deep Learning has revolutionized the field of artificial intelligence by enabling machines to perform complex tasks with human-like accuracy. From image recognition to natural language understanding, deep learning models have demonstrated exceptional performance, surpassing traditional methods. However, challenges like high data requirements and computational costs remain. Future research will focus on improving model efficiency, interpretability, and generalization, making deep learning more accessible and impactful across industries.

**✅ Deep Learning = Power + Complexity + Innovation 🚀**