INTEGRATED ROBUST SOLUTIONS, ISLAMABAD

Week 3 Task: AI Internship

Advanced Machine Learning Models

Prepared by:	Asad Ali
Designation:	Lead AI Developer
Department:	Artificial Intelligence

Version: 1.0

Date: July 18, 2025

Prepared for IR Solutions AI Interns

Contents

1 Introduction

This document outlines the Week 3 tasks for AI interns at IR Solutions. The tasks focus on advancing expertise in machine learning (ML) through theoretical learning of advanced ML model architectures and practical implementation. Interns will study selected modules from Google's Machine Learning Crash Course and complete two programming tasks in Python using Google Colab: breast cancer prediction using multiple ML models and Fashion MNIST clothing classification using a convolutional neural network (CNN). Weekly deliverables include a PDF report and Colab .ipynb files, submitted by **Friday, July 25, 2025, 5:30 PM PKT**.

2 Objectives

The Week 3 tasks aim to:

- Develop understanding of advanced ML model architectures, including neural networks, embeddings, and large language models.
- Apply data preprocessing and advanced ML models to real-world datasets.
- Compare performance of multiple ML models and implement a CNN for image classification.
- Practice professional documentation through structured PDF reports and well-commented Colab notebooks.
- Adhere to best programming practices (e.g., PEP 8, modular code, clear documentation).

3 Theoretical Learning

Interns are required to complete the following modules from Google's Machine Learning Crash Course (`https://developers.google.com/machine-learning/crash-course`):

3.1 Neural Networks

- **Course Material**: Neural Networks
- **Learning Objectives**:
  - Understand the fundamental principles of neural network architectures, including perceptrons, hidden layers, and activation functions.
  - Learn how neural networks process data and optimize model performance.

3.2 Embeddings

- **Course Material**: Embeddings
- **Learning Objectives**:
  - Explore how embeddings enable machine learning on large feature vectors.
  - Understand the role of embeddings in reducing dimensionality and capturing relationships.

3.3 Large Language Models

- **Course Material**: Large Language Models
- **Learning Objectives**:
  - Learn the basics of large language models, from tokens to Transformers.
  - Understand how LLMs are architected, trained, and used for text prediction.

4 Practical Tasks

Interns will complete two programming tasks in Python using Google Colab. Each task includes a dataset and specific questions to guide implementation and analysis. Submit the Colab notebooks as `Week3_InternName_Task01.ipynb` and `Week3_InternName_Task02.ipynb`.

4.1 Task 01: Breast Cancer Prediction using Multiple ML Models

- **Description**: Build and compare multiple ML models (Logistic Regression, Decision Tree, Random Forest, and Support Vector Classifier) to predict breast cancer diagnosis (malignant or benign) based on features like radius, texture, and perimeter. The dataset (`data.csv`) is available in the shared Google Drive folder.
- **Steps**:
  - Load and explore the dataset.
  - Preprocess numerical data and encode the target variable (`diagnosis`).
  - Train Logistic Regression, Decision Tree, Random Forest, and SVC models using scikit-learn.
  - Evaluate models using accuracy, precision, recall, F1-score, and AUC; generate visualizations.
  - Save the best model using `pickle`.
- **Questions**:
  1. Load the dataset using `pd.read_csv`. How many rows and columns are in the dataset? List the column names.
  2. Identify and handle missing values. How was the `Unnamed: 32` column handled?
  3. Apply `StandardScaler` to numerical features. Why is standardization important for these models?
  4. Split the dataset into training (70%) and test (30%) sets. What are the sizes of these sets?
  5. Train Logistic Regression, Decision Tree, Random Forest, and SVC models. Report their accuracy scores. Which model performed best, and why?
  6. Generate and interpret the confusion matrix for the Random Forest model. What are the implications of the recall score?
  7. Plot the ROC curve for the Logistic Regression model and report the AUC score. What does the AUC indicate?
  8. Analyze feature importance for the Random Forest model. Identify the top three features and explain their significance.

9. Perform 5-fold cross-validation for the SVC model. Report the mean accuracy and standard deviation.
10. Save the best model using `pickle`. Why is this step important for reproducibility?

- **Deliverables**:
  - Colab notebook (`Week3_InternName_Task01.ipynb`) with answers to the questions in Markdown cells, complete code, and visualizations (e.g., ROC curve, confusion matrix, feature importance).
  - Include comments following PEP 8 standards and ensure the notebook is executable.

4.2 Task 02: Fashion MNIST Clothing Classification using CNN

- **Description**: Build a convolutional neural network (CNN) to classify clothing images from the Fashion MNIST dataset, which contains 60,000 training and 10,000 test 28x28 grayscale images across 10 classes. The dataset is accessible via `keras.datasets.fashion_mnist`.
- **Steps**:
  - Load and preprocess the dataset (reshape, normalize, and encode labels).
  - Build a CNN with two convolutional layers, max-pooling layers, and dense layers using Keras.
  - Train the model and evaluate its performance using accuracy and loss.
  - Generate visualizations (e.g., sample images, loss/accuracy curves).
  - Save the model weights using `pickle`.
- **Questions**:
  1. Load the Fashion MNIST dataset using `keras.datasets.fashion_mnist`. What are the shapes of the training and test sets?
  2. Normalize the pixel values to the range [0, 1]. Why is normalization critical for CNNs?
  3. Apply one-hot encoding to the labels. Why is this necessary for multi-class classification?
  4. Build a CNN with two convolutional layers (128 filters, 3x3 kernel), ReLU activation, max-pooling, and dense layers. Explain the role of each layer.
  5. Train the model for 10 epochs with a batch size of 128. Report the final training and test accuracy.
  6. Plot the training and validation loss curves. Is there evidence of overfitting? Suggest one method to mitigate it.
  7. Generate and interpret the confusion matrix for the test set. Which classes are most commonly confused?
  8. Visualize the first test image and its predicted label. Was the prediction correct?
  9. Perform 5-fold cross-validation on a smaller subset (e.g., 10,000 samples) of the training data. Report the mean accuracy.
  10. Save the model weights using `pickle`. Why is this step important for reproducibility?
- **Deliverables**:

  - Colab notebook (`Week3_InternName_Task02.ipynb`) with answers to the questions in Markdown cells, complete code, and visualizations (e.g., loss curves, confusion matrix, sample predictions).
  - Include comments following PEP 8 standards and ensure the notebook is executable.

5 Submission Guidelines

5.1 PDF Report

- **Format**:
  - **Title Page**: Include IR Solutions logo, intern's name, designation (AI Intern), department (Artificial Intelligence), and date.
  - **Sections**:
    - **Theoretical Understanding**: Summarize key concepts from the assigned course modules (Neural Networks, Embeddings, Large Language Models).
    - **Project Work**: Describe the two tasks, including methodology, code implementation, and results.
    - **Challenges and Approaches**: List challenges faced in the tasks and solutions applied.
    - **Results and Analysis**: Present model performance (e.g., accuracy, AUC, recall) with plots (e.g., ROC curves, confusion matrices, loss curves).
    - **Spare Time Activities**: Note any additional learning or exploration during spare time.
    - **Conclusion**: Summarize key takeaways and plans for improvement.
    - **References**: Cite all resources used.
  - Use professional formatting with headings, subheadings, bullet points, and tables/plots for results.
- **File Naming**: `Week3_InternName_Report.pdf`

5.2 Colab Notebooks

- Submit two notebooks: `Week3_InternName_Task01.ipynb` and `Week3_InternName_Task02.i`
- Follow PEP 8 standards, include clear comments, and use print statements for debugging.
- Use Markdown cells for task overview, methodology, question answers, and analysis.
- Include visualizations (e.g., ROC curves, confusion matrices, loss curves, feature importance).
- Ensure notebooks are fully executable with no errors.

5.3 Submission Process

- Upload the PDF report and both .ipynb files to the designated Google Drive folder by **Friday, July 25, 2025, 5:30 PM PKT**.
- Ensure files are clearly named and organized in the Week 3 folder.

6 Evaluation Criteria

Interns will be evaluated based on:

- **Theoretical Understanding (30%)**: Depth and clarity of concepts learned from the course modules.
- **Project Output & Code Implementation (30%)**: Correctness, efficiency, and documentation of code; quality of analysis and visualizations.
- **Report Quality (20%)**: Clarity, completeness, and adherence to the defined format.
- **Best Practices (10%)**: Use of PEP 8, modular code, and clear documentation.
- **Participation (10%)**: Engagement in the Week 3 AI standup meeting and responsiveness to feedback.

7 Additional Notes

- **Dataset Access**: The breast cancer dataset (`data.csv`) is available in the shared Google Drive folder. The Fashion MNIST dataset is accessible via `keras.datasets.fashion_mnist`.
- **Support**: Reach out to AI team members or Lead AI via Slack or during the standup meeting (Friday, 3:00 PM PKT) for guidance.
- **Expectations**: Maintain professionalism, meet deadlines, and actively engage in learning and collaboration.

8 References

- Google Machine Learning Courses: `https://developers.google.com/machine-learning`
- GeeksforGeeks:
  - `https://www.geeksforgeeks.org/machine-learning`
  - `https://www.geeksforgeeks.org/machine-learning-projects`
  - `https://www.geeksforgeeks.org/deep-learning-tutorial`
  - `https://www.geeksforgeeks.org/computer-vision`
  - `https://www.geeksforgeeks.org/natural-language-processing-nlp-tutor`
- YouTube Tutorials:
  - `https://www.youtube.com/watch?v=_3y7C7_do-k`
  - `https://www.youtube.com/watch?v=7Bg3iBq-SIY`
  - `https://www.youtube.com/watch?v=0YdpwSYMY6I`
  - `https://www.youtube.com/watch?v=yLlgHKaMyIw`
  - `https://www.youtube.com/watch?v=NOJOYcmyDhM`