INTEGRATED ROBUST SOLUTIONS, ISLAMABAD

Week 1 Task: AI Internship

Introduction to Machine Learning and Practical Tasks

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

Version: 1.0

Date: July 04, 2025

Prepared for IR Solutions AI Interns

Contents

1 Introduction

This document outlines the Week 1 tasks for AI interns at IR Solutions. The tasks are designed to build a foundational understanding of machine learning (ML) through theoretical learning and practical implementation. Interns will study selected modules from Google's Machine Learning Foundational Courses and complete two programming tasks in Python using Google Colab: insurance cost prediction using linear regression and spam detection using logistic regression. Weekly deliverables include a PDF report and Colab .ipynb files, submitted by Friday, 05:30 PM PKT.

2 Objectives

The Week 1 tasks aim to:

- Develop a foundational understanding of ML concepts, including supervised learning, linear regression, logistic regression, and binary classification.
- Apply theoretical knowledge to practical ML problems using Python in Google Colab.
- 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 Foundational Courses (`https://developers.google.com/machine-learning/foundational-cours`

3.1 Introduction to Machine Learning

- **Course Material**: Introduction to Machine Learning
- **Learning Objectives**:
  - Understand the types of ML: supervised, unsupervised, and reinforcement learning.
  - Identify key components of ML workflows.

3.2 Linear Regression

- **Course Material**: Machine Learning Crash Course: Linear Regression
- **Learning Objectives**:
  - Understand linear regression models and their applications.
  - Learn about loss functions, gradient descent, and hyperparameter tuning.

3.3 Logistic Regression

- **Course Material**: Machine Learning Crash Course: Logistic Regression
- **Learning Objectives**:
  - Understand logistic regression for predicting probabilities.
  - Explore the sigmoid function and its role in classification.

3.4 Classification

- **Course Material**: Machine Learning Crash Course: Classification
- **Learning Objectives**:
  - Learn binary classification concepts, including thresholding and confusion matrices.
  - Understand evaluation metrics such as accuracy, precision, recall, and AUC.

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 `Week1_InternName_Task01.ipynb` and `Week1_InternName_Task02.ipynb`.

4.1 Task 01: Insurance Cost Prediction using Linear Regression

- **Description**: Use the insurance dataset to predict yearly medical costs based on features like age, sex, BMI, number of children, smoking status, and region. The dataset is available at the drive link already shared.
- **Steps**:
  - Download and explore the dataset.
  - Prepare the dataset for training (handle categorical variables, convert to tensors).
  - Create and train a linear regression model using PyTorch.
  - Evaluate the model and make predictions.
  - Make necessary plots.
  - Save the trained model.
- **Questions**:
  1. Load the dataset using `pd.read_csv`. How many rows and columns are in the dataset? List the column names.
  2. Implement the `customize_dataset` function using your name (at least 5 characters). Explain how the function modifies the dataset.
  3. Identify the input, categorical, and output columns. Why is it necessary to convert categorical columns to numerical codes?
  4. Convert the dataset to PyTorch tensors using `dataframe_to_arrays`. What are the shapes of the input and target arrays?

5. Split the dataset into training and validation sets with `val_percent = 0.15`. What are the sizes of the training and validation sets?
6. Create a `DataLoader` with `batch_size = 16`. Print the first batch of inputs and targets.
7. Define the `InsuranceModel` class. Why is `nn.Linear` used, and what are the input and output sizes?
8. Train the model for 500 epochs with `lr = 1e-2`. Plot the validation loss over epochs. What trend do you observe?
9. Make predictions for three validation samples using `predict_single`. Compare predictions with actual targets.
10. Save the model using `pickle`. Why is model saving important for deployment?

- **Deliverables**:
  - Colab notebook (`Week1_InternName_Task01.ipynb`) with answers to the questions in Markdown cells, complete code, and visualizations (e.g., loss curves).
  - Include comments following PEP 8 standards and ensure the notebook is executable.

4.2 Task 02: Spam Detection using Logistic Regression

- **Description**: Build a logistic regression model to classify SMS messages as spam or non-spam using the dataset `sms-data-labelled-spam-and-non-spam.csv`. Upload the dataset to your Colab environment.
- **Steps**:
  - Load and explore the dataset.
  - Preprocess the text data (e.g., convert to numerical features using TF-IDF).
  - Create and train a logistic regression model using PyTorch.
  - Evaluate the model using metrics Stuart metrics (accuracy, precision, recall).
  - Make necessary plots.
  - Make predictions on test data.
- **Questions**:
  1. Load the dataset using `pd.read_csv`. How many rows and columns are in the dataset? List the column names.
  2. Preprocess the text data using `TfidfVectorizer` from `scikit-learn`. What does TF-IDF represent, and why is it suitable for text data?
  3. Convert the TF-IDF features and labels to PyTorch tensors. What is the shape of the feature matrix?
  4. Split the dataset into training (80%) and validation (20%) sets. What are the sizes of these sets?
  5. Create a `DataLoader` with `batch_size = 32`. Print the first batch of inputs and targets.
  6. Define a `SpamModel` class using `nn.Linear` and `sigmoid`. Why is the sigmoid function used for logistic regression?

7. Train the model for 100 epochs with `lr = 1e-3` using `BCELoss` (Binary Cross-Entropy Loss). Plot the validation loss over epochs.
8. Evaluate the model on the validation set. Report accuracy, precision, recall, and AUC metrics.
9. Make predictions for three validation samples. Compare predictions with actual labels.
10. Discuss one challenge faced during preprocessing or training and how you addressed it.

- **Deliverables**:
  - Colab notebook (`Week1_InternName_Task02.ipynb`) with answers to the questions in Markdown cells, complete code, and visualizations (e.g., loss curves, confusion matrix).
  - 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.
    - **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., validation loss, accuracy, precision, recall) with plots (e.g., loss curves, confusion matrices).
    - **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**: `Week1_InternName_Report.pdf`

5.2 Colab Notebooks

- Submit two notebooks: `Week1_InternName_Task01.ipynb` and `Week1_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 hyperparameter analysis.

- Include visualizations (e.g., loss curves, confusion matrices, accuracy plots).
- 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, 05:30 PM PKT**.
- Ensure files are clearly named and organized in the Week 1 folder.

6 Evaluation Criteria

Interns will be evaluated based on:

- **Theoretical Understanding (30%)**: Depth and clarity of concepts learned from the course modules.
- **Project Quality (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 1 AI standup meeting and responsiveness to feedback.

7 Additional Notes

- **Dataset Access**: The dataset is available at the shared drive folder.
- **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