PORTFOLIO

1. B-bot (Personal Chatbot)

The personal chatbot I created serves as an efficient way of retrieving information about me that is listed in my resume. It provides prompt responses to user queries related to my skills, education, work experience, and other relevant information without the need to look about it in my resume. The chatbot operates by taking prompts from the user and classifying them based on predefined tags. To achieve this, I used a training file containing labeled sets of questions and answers. For this task, I utilized an MLP classifier with three hidden layers and ReLU activations. To represent text data, I used the Bag of Words (BoW) approach, which enables me to extract relevant keywords and provide appropriate responses to user queries. For instance, if anyone asks about my experiences, the chatbot returns a predefined response for my experiences. Furthermore, I have implemented text preprocessing techniques such as stemming and spell-checking to ensure the correctness of responses. The chatbot interface was developed using Streamlit and has been hosted on their platform.

Tools used: Streamlit-chat, PyTorch, PySpellchecker, and NLTK
Chatbot Link: https://bibekbot.streamlit.app/
Github Link: https://github.com/bibekyess/Personal-Chatbot

2. 6D Pose Estimation and Scene Graph Generation (SGG)

I used Cosypose framework for 6D pose and Neural-Motif framework for SGG and performed experiments on custom cup datasets. This was done for a single-camera setup and I estimated 6D pose and generated dynamic scene graph. For 6 cups, with 6D pose estimation, relationship and attributes predictions, the combined framework ran at a speed of 1.25frames/s.

Demo: https://tinyurl.com/pose-sgg-demo

3. Research on Probabilistic Multiview Object-Pose Estimation

We proposed a novel probabilistic multiview object-pose estimation framework that 1) associates multiple estimation results with scene graphs, 2) combines pose distributions from single-view based estimators, and 3) construct a unified scene graph by predicting a unified pose distribution per object. We evaluate our method with table-top object-pose estimation scenarios and show the proposed approach outperforms three baseline methods in terms of position and orientation accuracies. My task involved conducting neural architecture research for pose estimation and scene graph generation, followed by training and performing combined efficient inference. Meanwhile, my friend was responsible for environment setup, creating a simulation dataset, and visualization.

Draft paper link: https://tinyurl.com/kroc-draft

4. AI-empowered F1/10 Autonomous Racing Car

This is a team project for one course. Since I am more comfortable using Linux, I was in charge of setting up the environment and internet on the Jetson Xavier NX, including installing libraries and ROS packages for sensor interfaces. The other team members handled the hardware setup for the car, including the Arduino, LIDAR, IMU, camera, chassis, and wiring. My main responsibility was developing the perceptron, specifically object detection. I used Yolact_ros (it is instance segmentation-based framework and is more accurate) as we had plan to extend 2D detections to 3D detections for obstacle avoidance. However, it was computationally heavy, so we tried obstacle avoidance from LIDAR data. Later, due to computational issues, I used a simple rule-based classifier that involved cropping, masking, and counting the pixel values of red, green, and yellow. We only used the vision model for traffic light detection. For localization, we used AMCL and Hector Odom, while for control, we used PID. I also helped the team with algorithm testing and parameter tuning in the real-world environment.

Tools Used: Yolact_ros, OpenCV, Linux, ROS, NVIDIA SDK, Arduino IDE

Final Report: https://tinyurl.com/ee405-finalreport

5. Movie Recommendation (Similar to NetFlix Challenge, 2006)

My task was to fill in the <RATING> column of an input file that contained <USER ID>, <MOVIE ID>, and <TIMESTAMP> entries. To estimate the blank entries in the Utility matrix, which represented n users and m movies with rating values as entries, I used UV-decomposition for dimensionality reduction. This method approximates the matrix as R ≈ UV^T, where U is an n x k matrix and V is an m x k matrix, with k being smaller than n and m. Each user and movie is represented by a k-dimensional vector of latent factors. To train the UV decomposition, I used an iterative approach called Alternating Least Squares (ALS) to minimize the root-mean-square error (RMSE). Before training, I performed two-fold normalization by subtracting the average rating of the user and then the average rating of the movie. Finally, I denormalized the utility matrix and used the U and V matrices to predict the rating for each corresponding user and movie entry.

Tools Used: Numpy, Python

6. Paraphrase Generation Project:

We replicated the paper Quality Controlled Paraphrase Generation (Bandel et al., ACL 2022). My main part was to try out improvement approaches. First, I checked the empirical importance of QP (Quality Predictor) model by comparing the results of QCPG with and without QP. Secondly, I figured out a way to change the t5-base to multilingual mt5-base for training Korean dataset. Third, I wrote an end-end inference pipeline for paraphrasing from scratch.

Tools Used: Pytorch, Matplotlib, transformers, Huggingface models

Replicated Paper Link: https://tinyurl.com/qcpg-replication

GitHub Link: https://github.com/bibekyess/QCPG

7. Pintos Project

It was the course project for my Operating Systems course, where I implemented pintos-kaist, a simple operating system framework for the x86-64 architecture. This project was forked from the pintos project at Stanford University, and we ran the code under the QEMU simulation environment. In the first project, my task was to solve thread synchronization problems with interrupts using semaphores, locks, and condition variables. In project 2, I implemented a system call handler for I/O interactivity in running user programs. In project 3, I implemented virtual memory management, including page fault handling and swapping, to build an illusion of infinite memory. In the final project, I worked on implementing the file system and succeeded on making the basic file system part to support addition, deletion, and manipulation of files.
Overall, it was a great experience working in a large repository.
Project Reference: https://casys-kaist.github.io/pintos-kaist/

8. Crypto Price Predictor:

We used time-series forecasting and attention mechanisms to predict the future price of bitcoin. We developed seven forecasting models from deep neural networks- to Bayesian network-based model with two forecasting tasks: 1) one-day and 2) long-sequence (up to 100 days) forecasting. In my part, I experimented with two deep learning-based forecasting models: one is recurrent neural network based (bidirectional LSTM) and the other is convolutional neural network based (Conv1D) for one-day forecasting task. I also did multivariate forecasting with both of these models.
Tools used: TensorFlow, Pandas, Matplotlib
Project Paper Link: https://tinyurl.com/cryptopredictor
GitHub Link: https://github.com/bibekyess/AI_practice/blob/main/Killionaire.ipynb

9. Smart Home Project:

We did a security tech project where we detect any intruders and notify the house owner. In my part, I set up hardware (connecting the sensor to raspberry pi using breadboard) and my part was to manage the Node-Red workflow i.e connect each process to Telegram bot for automated response.
Tools Used: Node-RED
Demo Video: https://www.youtube.com/watch?v=YhccA0mm2YA
GitHub Link: https://github.com/bibekyess/HomeSec

10. Lecture Summarizer Project:

I contributed in making documents, specifically Project Planning & Management Document (PPMD), Software Architecture Document (SAD) and System Testing Document. Around 80% of the part in these documents were done by me. In development part, I was not explicitly involved. From our team of 4 members, 3 of them specifically did frontend, backend and NLP part. I helped them in two issues, first one is converting the text to pdf (there was issue in formatting because of the absence of EOL

character in the text format). Secondly, I managed the dependencies in project using conda and updated README to make our code work in Linux, Mac and Windows.
Tools Used: Django, React.js, NLP (Rule-based), Git, Notion, Figma
Presentation Link: https://tinyurl.com/lecsum-eosp
GitHub Link: https://github.com/NishantNepal1/LecSum

11. Data Cleanup Project:

I contributed to data visualization, and webpage design while my teammate was assigned the task of data pre-processing.
Tools Used: Python, Plotly, Pandas, Flask, Dash, Heroku
Link: https://data-anamoly-filter.onrender.com/
Demo Video: https://tinyurl.com/data-anamoly-filter
GitHub Link: https://github.com/bibekyess/data-viz-project

12. Data Visualization Task:

This was a visualization task given to me by one startup before hiring me as an intern in Winter 2022.
Tools Used: D3.js, Colab (for data preprocessing), Flask, Heroku
Link: https://visualization-app.onrender.com/
GitHub Repo: https://github.com/bibekyess/Bisonai-Assignment