DevelopersHub Corporation

AI/ML Internship Report

Muhammad Usman

Position: AI/ML Intern

Signature: _____

Executive Summary

During my internship at DevelopersHub Corporation, I successfully completed three tasks from a set of six assigned projects: Task 4 (General Health Query Chatbot), Task 5 (Mental Health Support Chatbot), and Task 6 (House Price Prediction). These tasks were strategically chosen to demonstrate my proficiency in diverse areas of artificial intelligence and machine learning, including natural language processing (NLP), model fine-tuning, regression modeling, and data visualization.

Each task presented unique challenges, such as designing effective prompts, managing dataset preprocessing, and optimizing model performance, which enhanced my technical and problem-solving skills. The projects collectively showcase my ability to apply AI/ML techniques to real-world problems, from conversational AI to predictive analytics. This report provides a detailed account of the objectives, methodologies, implementations, results, challenges, and learnings for each task, concluding with recommendations for future improvements and references to resources utilized. The internship experience has significantly strengthened my expertise, preparing me for advanced challenges in AI/ML development.

Table of Contents

Introduction

This report provides a detailed overview of the objectives, methodologies, implementations, results, challenges, and learnings for each task. It is structured to offer a clear and professional summary of my work during the internship, highlighting my growth as an AI/ML practitioner. The projects not only strengthened my technical proficiency but also enhanced my ability to approach complex problems systematically, preparing me for future opportunities in AI/ML development.

Task 4 involved developing a chatbot using the Mistral-8x7B-Instruct model, accessed via the OpenRouter API, to provide general health information with a friendly and safe tone. The chatbot incorporated safety filters to handle sensitive queries and was deployed through a user-friendly Streamlit web interface, ensuring accessibility and engagement. Task 5 focused on creating an empathetic mental health support chatbot by fine-tuning DistilGPT2 on the EmpatheticDialogues dataset, resulting in a command-line interface capable of generating supportive responses for users experiencing stress or anxiety. Task 6 entailed building a house price prediction model using the GradientBoostingRegressor on the Kaggle House Price Prediction Dataset, achieving reasonable predictive performance through careful feature engineering and data preprocessing.

Task 4: General Health Query Chatbot

Objective

The objective of Task 4 was to develop a chatbot powered by a large language model (LLM) to answer general health-related queries in a friendly, safe, and informative manner. The chatbot was designed to provide accurate general information while explicitly avoiding specific medical advice, ensuring user safety through the implementation of safety filters for sensitive queries. It was also required to guide users to consult healthcare professionals for personalized medical advice, maintaining ethical and responsible communication.

Methodology

The approach to building the chatbot was structured to ensure both functionality and user safety:

- **Model Selection**: I chose the Mistral-8x7B-Instruct model, accessed via the OpenRouter API, due to its efficiency, robust natural language understanding, and accessibility for rapid prototyping and deployment.
- **Prompt Engineering**: A carefully crafted system prompt was designed to enforce a friendly, empathetic, and clear tone in responses. The prompt instructed the model

to provide general health information, include disclaimers about consulting professionals, and avoid specific diagnoses or treatment recommendations.

- **Safety Filters**: A keyword-based filter was implemented to detect sensitive terms such as "emergency," "suicide," or "overdose." When triggered, the chatbot would respond with a warning and direct users to seek immediate professional help, ensuring responsible handling of critical queries.
- **Interface**: A Streamlit-based web application was developed to provide a user-friendly interface, featuring a conversational history display and Lottie animations to enhance visual engagement and improve the user experience.
- **Tools**: The project utilized Python for scripting, Streamlit for the web interface, the Requests library for API communication, Lottie animations for visual enhancements, and the OpenRouter API for model access.

Implementation

The chatbot was implemented in a Python script named `chat_app.py`, leveraging Streamlit for the front-end interface. The system prompt was designed to ensure responses adhered to safety and ethical guidelines, clearly stating the chatbot's role as a provider of general information. A predefined list of unsafe keywords was integrated into the logic, enabling the chatbot to identify sensitive queries and respond with appropriate warnings, such as directing users to emergency services or healthcare professionals.

The Streamlit application provided an intuitive interface where users could input queries and view the conversational history in real time. Lottie animations were incorporated to add a dynamic and engaging visual element, making the interaction more appealing. For example, a query like "What causes a sore throat?" would prompt the chatbot to explain common causes (e.g., viral infections, allergies, or environmental factors) while including a disclaimer to consult a doctor for persistent or severe symptoms.

Results

The implementation yielded the following outcomes:

- The chatbot successfully delivered general health information with a friendly and empathetic tone, aligning with the project's objectives.
- Safety filters effectively identified sensitive queries, preventing potentially harmful responses and redirecting users to professional resources when necessary.
- The Streamlit interface was intuitive and user-friendly, with clear conversational history and engaging Lottie animations that enhanced the user experience.
- Example response for "What causes a sore throat?": "A sore throat can be caused by viral infections like the common cold, allergies, dry air, or irritants like smoke. If symptoms persist or worsen, please consult a healthcare professional for proper diagnosis and treatment."

Challenges and Learnings

- **Challenge**: Balancing the provision of informative responses with the restriction on offering specific medical advice required meticulous prompt engineering to ensure clarity without crossing ethical boundaries.
- **Learning**: I gained significant expertise in prompt engineering, learning how to control LLM behavior to produce safe and contextually appropriate responses. This included crafting prompts that balanced informativeness with cautionary disclaimers.
- **Challenge**: API rate limits from the OpenRouter API occasionally disrupted the testing workflow, causing delays in iterative development.
- **Learning**: I learned to optimize API calls by caching responses where possible and implementing robust error handling using try-except blocks, improving the reliability of API-based applications.

Task 5: Mental Health Support Chatbot

Objective

The objective of Task 5 was to develop a chatbot that provides empathetic and supportive responses to users expressing mental health concerns, such as stress, anxiety, or low mood. The chatbot was built by fine-tuning a small language model on the EmpatheticDialogues dataset to ensure responses were gentle, supportive, and appropriate, fostering a safe conversational environment for users seeking emotional support.

Methodology

The approach to building the mental health support chatbot was designed to leverage lightweight NLP models and specialized datasets for empathy-driven responses:

- **Model Selection**: I selected DistilGPT2, a compact version of GPT-2, for its lightweight architecture, which is well-suited for fine-tuning on limited computational resources while maintaining reasonable performance in text generation.
- **Dataset**: The EmpatheticDialogues dataset from Facebook AI was used, which contains thousands of human-to-human dialogues crafted to exhibit empathy in response to various emotional contexts, making it ideal for training a supportive chatbot.
- **Fine-Tuning**: The Hugging Face Trainer API was employed to fine-tune DistilGPT2 on the EmpatheticDialogues dataset, focusing on generating responses that are empathetic, supportive, and contextually relevant to mental health queries.
- **Interface**: A command-line interface was developed to allow users to interact with the fine-tuned model, providing a simple yet effective platform for testing and validation.

- **Tools**: The project utilized Python for scripting, Hugging Face Transformers and Datasets libraries for model training and data handling, PyTorch for the underlying deep learning framework, and the DistilGPT2 model for text generation.

Implementation

The task was implemented through two Python scripts to separate the training and interaction phases:

- **back.py**: This script handled data preprocessing, tokenization, and model fine-tuning. The EmpatheticDialogues dataset was preprocessed by tokenizing dialogues with a maximum length of 128 tokens to manage computational efficiency. DistilGPT2 was fine-tuned for three epochs with a batch size of 4 using the Hugging Face Trainer API, optimizing for empathetic response generation. The fine-tuned model and tokenizer were saved to the `./empathetic_model` directory for later use.
- **main.py**: This script loaded the fine-tuned DistilGPT2 model and provided a command-line interface for user interaction. Users could input text (e.g., "I'm feeling stressed"), and the model generated responses with parameters such as `top_k=50`, `top_p=0.95`, and `temperature=0.8` to ensure diverse yet coherent and empathetic outputs. The interface included a clear exit mechanism by typing "quit."

Results

The implementation produced the following outcomes:

- The fine-tuned DistilGPT2 model successfully generated empathetic responses tailored to mental health concerns. For example, an input like "I'm feeling stressed" prompted a response such as, "I'm here for you. It sounds really tough, but you're not alone in feeling this way. Would you like to share more?"
- The command-line interface provided a seamless interaction experience, allowing users to engage with the chatbot and exit cleanly when desired.
- Training metrics indicated stable convergence, with the model achieving consistent performance by checkpoint-500, where it was saved for deployment.
- The chatbot effectively captured the emotional tone of user inputs, delivering responses that were supportive and encouraging, aligning with the project's goal of fostering emotional well-being.

Challenges and Learnings

- **Challenge**: Preprocessing the EmpatheticDialogues dataset required careful handling of tokenization and padding to avoid truncation errors, which could disrupt training and degrade response quality.
- **Learning**: I mastered the use of the Hugging Face Trainer API and tokenization processes, gaining a deeper understanding of preparing datasets for NLP tasks and ensuring compatibility with transformer models.

- **Challenge**: Balancing response diversity and coherence during text generation was critical to avoid repetitive or off-topic responses, which could undermine the chatbot's supportive role.
- **Learning**: Through experimentation with generation parameters like top_k, top_p, and temperature, I learned to optimize text generation for quality and relevance, enhancing my skills in fine-tuning language models for specific applications.

Task 6: House Price Prediction

Objective

The objective of Task 6 was to develop a predictive model to estimate house prices based on property features such as size, number of bedrooms, bathrooms, and location, using the House Price Prediction Dataset from Kaggle. The goal was to create an accurate regression model capable of capturing the relationships between these features and house prices, providing valuable insights for real estate applications.

Methodology

The approach to building the house price prediction model was structured to ensure robust data handling and model performance:

- **Model Selection**: I chose the GradientBoostingRegressor from Scikit-learn due to its robustness in handling regression tasks, ability to capture non-linear relationships, and effectiveness in reducing overfitting through ensemble techniques.
- **Dataset**: The House Price Prediction Dataset from Kaggle was utilized, containing features such as area (square footage), number of bedrooms, bathrooms, furnishing status, and binary indicators like mainroad, guestroom, and basement.
- **Preprocessing**: Binary features (e.g., mainroad, guestroom) were encoded as 1/0 for "yes"/"no" values, and the categorical feature furnishingstatus was one-hot encoded to create dummy variables. Numerical features like area and bedrooms were scaled using StandardScaler to normalize their distributions. The dataset was split into 80% training and 20% testing sets to evaluate model performance.
- **Evaluation**: Model performance was assessed using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), providing measures of prediction accuracy and error magnitude.
- **Visualization**: A scatter plot of predicted versus actual prices was generated to visually assess model performance, with a reference line indicating perfect predictions.
- **Tools**: The project utilized Python for scripting, Pandas for data manipulation, Scikit-learn for modeling and preprocessing, Matplotlib and Seaborn for visualization, and Jupyter Notebook for an interactive development environment.

Implementation

The task was implemented in a Jupyter Notebook named House_prediction_DevHub_Task06.ipynb, with the following steps:

- **Data Preprocessing**: The dataset was loaded using Pandas, and an initial exploration confirmed no missing values, simplifying preprocessing. Binary features like mainroad, guestroom, and basement were mapped to 1/0, and the categorical feature furnishingstatus (e.g., "furnished," "semi-furnished," "unfurnished") was one-hot encoded using Pandas' get_dummies function. Numerical features such as area, bedrooms, and bathrooms were scaled using Scikit-learn's StandardScaler to ensure consistent ranges. The dataset was then split into training (80%) and testing (20%) sets using train_test_split.
- **Model Training**: A GradientBoostingRegressor was instantiated with default hyperparameters and trained on the preprocessed training data, leveraging its ability to iteratively build decision trees to minimize prediction errors.
- **Evaluation**: The trained model was evaluated on the test set, yielding an MAE of 960,714.33 and an RMSE of 1,299,273.84, indicating the average prediction error in house prices.
- **Visualization**: A scatter plot was created using Matplotlib, plotting predicted prices against actual prices from the test set. A red dashed line representing perfect predictions (y = x) was added for reference, and the plot used green points with transparency (alpha=0.7) to highlight prediction density.

Results

The implementation produced the following outcomes:

- The GradientBoostingRegressor achieved an MAE of 960,714.33 and an RMSE of 1,299,273.64, indicating reasonable predictive performance, though the relatively high errors suggest potential areas for improvement.
- The scatter plot of actual versus predicted prices showed a positive correlation, with most points clustering near the perfect prediction line, indicating that the model effectively captured key relationships between features like area and price. However, some outliers highlighted instances where predictions deviated significantly.
- The model demonstrated the importance of feature engineering, as encoded features like furnishingstatus and scaled numerical features contributed to improved prediction accuracy.
- The results provided actionable insights into house price trends, useful for real estate stakeholders seeking to estimate property values based on structural and locational attributes.

Challenges and Learnings

- **Challenge**: Handling categorical features like furnishingstatus required careful one-hot encoding to manage dimensionality and avoid introducing errors, especially given the model's sensitivity to feature representation.
- **Learning**: I gained proficiency in feature engineering techniques, including one-hot encoding for categorical variables and scaling for numerical features, which are critical for optimizing regression model performance.
- **Challenge**: The model's high MAE and RMSE values indicated potential overfitting or limitations in the feature set, as some properties may have been influenced by unmodeled factors like neighborhood quality or market trends.
- **Learning**: I learned to interpret regression metrics like MAE and RMSE to assess model performance and use visualization techniques, such as scatter plots, to identify patterns and outliers, guiding iterative improvements in model development.

Conclusion

The successful completion of Tasks 4, 5, and 6 during my internship at DevelopersHub Corporation represents a significant milestone in my development as an AI/ML practitioner. Each task provided a unique opportunity to apply and refine my skills in distinct areas of artificial intelligence and machine learning, demonstrating my ability to tackle diverse and complex challenges.

Task 4, the General Health Query Chatbot, honed my expertise in prompt engineering and conversational AI safety, enabling me to create a user-friendly and ethically responsible system that balances informativeness with caution. Task 5, the Mental Health Support Chatbot, deepened my understanding of natural language processing through model fine-tuning, allowing me to build an empathetic conversational tool tailored to sensitive user needs. Task 6, the House Price Prediction model, strengthened my capabilities in regression modeling, feature engineering, and data preprocessing, providing practical experience in predictive analytics for real-world applications.

Collectively, these projects enhanced my technical proficiency in key AI/ML domains, including prompt design, transformer-based NLP, regression algorithms, and data visualization. They also developed my problem-solving skills, as I navigated challenges such as API rate limits, dataset preprocessing complexities, and model optimization. The internship experience has equipped me with the confidence and technical foundation to address real-world AI challenges effectively, preparing me for advanced roles in AI/ML development. The learnings from these tasks underscore the importance of iterative development, ethical considerations, and user-centric design in creating impactful AI solutions.

Future Improvements

Based on the experiences and challenges faced during the development of Tasks 4, 5, and 6 at DevelopersHub Corporation, the following improvements are proposed to enhance the performance, scalability, and user impact of each project:

- **Task 4: General Health Query Chatbot**
  Future enhancements could include integrating a verified medical knowledge base, such as data from trusted sources like the National Institutes of Health or WebMD, to improve the precision and credibility of responses while adhering to safety guidelines. Expanding the safety filter with advanced natural language processing techniques to detect nuanced risky queries, beyond keyword matching, would further ensure user safety. Additionally, deploying the chatbot across multiple platforms, such as mobile applications or social media messaging services, could broaden its reach and improve user accessibility.
- **Task 5: Mental Health Support Chatbot**
  To improve the quality and variety of empathetic responses, exploring larger language models like GPT-Neo or T5 could yield more nuanced and contextually appropriate outputs compared to DistilGPT2. Incorporating additional mental health-focused datasets or real-world user interactions could enhance the model's ability to handle diverse emotional scenarios. Upgrading the command-line interface to a web-based or mobile application with a more engaging design would make the chatbot more user-friendly and accessible to a larger audience.
- **Task 6: House Price Prediction**
  To boost prediction accuracy, implementing feature selection methods like recursive feature elimination or correlation analysis could help identify the most predictive features and eliminate irrelevant ones. Combining multiple regression models, such as Random Forest, XGBoost, and GradientBoostingRegressor, through ensemble techniques like stacking could reduce errors and improve robustness. Augmenting the dataset with external variables, such as local economic indicators or property age, could address outliers and enhance the model's applicability to real-world real estate scenarios.

References

[1] **Hugging Face Transformers Documentation**: A comprehensive guide for using transformer models, including fine-tuning and text generation, used extensively for Task 5. Available at: https://huggingface.co/docs/transformers

[2] **Scikit-learn Documentation**: Official documentation for Scikit-learn, utilized for regression modeling and preprocessing in Task 6. Available at: https://scikit-learn.org/stable/

[3] **EmpatheticDialogues Dataset**: A dataset of empathetic human dialogues from Facebook AI, used for fine-tuning the mental health chatbot in Task 5. Available at: https://huggingface.co/datasets/empathetic_dialogues

[4] **House Price Prediction Dataset**: A Kaggle dataset containing house features and prices, used for the regression model in Task 6. Available at: https://www.kaggle.com/datasets/harishkumardatalab/housing-price-prediction

[5] **OpenRouter API**: Documentation for accessing the Mistral-8x7B-Instruct model, used for the health chatbot in Task 4. Available at: https://openrouter.ai/docs
[6] **Streamlit Documentation**: Official guide for building web applications with Streamlit, used for the chatbot interface in Task 4. Available at: https://docs.streamlit.io