AI Basics

Omar Eltramsy

September 1, 2024

Abstract

This report focuses on a task called "AI Basics," which demonstrates the application of artificial intelligence and machine learning tools to solve real-life problems. The task is divided into three parts. The first part aims to understand weather patterns by analyzing historical data, extracting meaningful insights to aid in weather-related disaster prevention. The second part involves building a chess game that allows users to play with friends, implementing basic chess logic and game rules. The final part centers around predicting future weather conditions using machine learning models, with the goal of mitigating potential disasters through early forecasting.

1 Introduction

The rapid advancements in artificial intelligence **(AI)** and machine learning **(ML)** have unlocked vast potential for solving complex, real-world problems. This report focuses on a task titled "AI Basics," which leverages these technologies to address diverse challenges, from weather prediction to interactive game development.

The task is divided into three parts. **The first part** involves analyzing historical weather patterns to aid in disaster prevention. By cleaning and visualizing climate data, the goal is to extract actionable insights that can enhance forecasting accuracy and disaster preparedness.

The second part focuses on game development, specifically implementing a chess game using Python. This includes creating basic chess logic, enabling users to play against each other while adhering to official chess rules. This section highlights how AI can simulate strategic gameplay.

The final part involves predicting future weather conditions by applying machine learning models. The objective is to build a predictive model based on historical meteorological data, which can offer early warnings to help prevent disasters.

Collectively, these tasks demonstrate the practical applications of AI and machine learning in data analysis, interactive system development, and real-world problem-solving.

2 Understanding The Weather

This task focuses on understanding weather patterns by analyzing historical weather data. The primary tools used for data manipulation and visualization are Pandas, Matplotlib, and Seaborn. The goal is to clean the dataset, organize it, and create visualizations to gain insights into weather trends.

The process begins with cleaning the data. This includes removing duplicates and handling missing values by replacing them with appropriate defaults, such as zero for numerical fields, depending on their type. Once the data is cleaned and organized, the next step is visualizing the relationships between key variables.

The first visualization is a line plot using Matplotlib, which shows the temperature over time. This helps identify temperature trends and variations across the dataset. Following that, a histogram is created to visualize the distribution of temperature values, allowing us to observe how frequently different temperature ranges occur.

Next, a scatter plot is generated to show the relationship between temperature and humidity, revealing any potential correlations between these two variables. Finally, a heatmap is drawn using Seaborn, displaying the correlation matrix of all the numerical features in the dataset. This heatmap helps us understand the relationships between different weather variables and how they interact with each other.

By cleaning the data and creating these visualizations, we are able to gain a clearer understanding of historical weather patterns and relationships between different weather-related variables.

3 Chess Game

In this project, we used the Pygame library to create a two-player chess game. The game board is drawn on an 8x8 grid, consisting of 32 light and dark gray squares, each representing a tile where pieces can move. The chess pieces for both players are initialized with appropriate images and placed in their correct starting positions on the board.
We built several core features for the chess game:
Piece Definition: Each player (white and black) starts with their full set of pieces—rook, knight, bishop, queen, king, and pawns. These pieces are loaded with their corresponding images and scaled appropriately for display.
Game Board: The board consists of an alternating pattern of squares. In addition to the chessboard itself, there are areas dedicated to showing captured pieces and game status.
Piece Movements: Each piece has its unique movement rules implemented through various functions (check king, check queen, check rook, etc.). These functions validate legal moves according to chess rules and handle piece captures.
Turn-Based Gameplay: The game alternates between white and black turns, with each player able to select a piece and move it to a valid position on the board. If a piece is captured, it is stored in a separate array and displayed on the side of the board.
Check and Checkmate: The game detects when a king is in check and displays a flashing outline around the king. A win condition is triggered if a player's king is captured, displaying the winning player and resetting the game.
Captured Pieces: Captured pieces are stored in arrays for each player and drawn on the screen, allowing players to see which pieces they have lost.

4 Predicting The Weather

In this project, we employed machine learning techniques to predict future weather conditions using historical weather data. The primary goal was to forecast the temperature for the next day based on historical records of temperature, humidity, and wind speed.
Data Preparation: We began by loading the historical weather data from a CSV file, which included various features such as temperature, humidity, and wind speed. To ensure the quality of our dataset, we performed several preprocessing steps:
Duplicate Removal: We checked and removed any duplicate records from the dataset. Missing Values: We addressed missing values in the 'Precip Type' column by filling them with a placeholder value of 0. This step was crucial for maintaining the integrity of our predictive model.
Feature Engineering: To enhance the predictive capabilities of our model, we engineered several features:
Target Variable: We introduced a new column, 'target,' which represents the temperature for the following day. This column was created by shifting the 'Temperature (C)' column by one time step.
Rolling Average: We calculated a 30-day rolling average of the temperature, which helped in capturing the trend over time.
Model Training and Testing: We used a Ridge regression model, a type of linear regression that incorporates regularization to prevent overfitting. The dataset was divided into training and testing sets based on dates:
Training Set: Data up to December 31, 2014.
Testing Set: Data from January 1, 2015, onwards.
We trained the Ridge regression model on the training data and used it to predict the temperature in the testing set. The model's performance was evaluated using the Mean Absolute Error (MAE), which measures the average magnitude of the errors in the predictions.
Results and Evaluation: The model's predictions were compared against the actual temperatures in the testing set. We observed the following:
Error Measurement: The Mean Absolute Error was calculated to quantify the model's prediction ac-

curacy.
Results Visualization: The actual versus predicted temperatures were visualized to assess the model's performance.
Code Implementation: The implementation involved loading and preprocessing the data, training the model, making predictions, and evaluating the results. Key steps in the code included:
Handling missing values and feature engineering.
Training the Ridge regression model.
Evaluating prediction accuracy and visualizing the results.
In conclusion, this project demonstrated the effectiveness of using Ridge regression for weather prediction. The model was able to provide reasonably accurate forecasts, and the approach can be further refined by incorporating additional features or trying different algorithms.

5 Conclusion

In this project, we successfully developed a predictive model to forecast daily temperatures using historical weather data. By applying Ridge regression—a robust technique for linear modeling with regularization—we were able to construct a model that balances prediction accuracy with generalization capabilities.