Hello, my name is Bhuvaneshwar V, and I am a Data Scientist with 3 years of experience, including 2 years in data science roles. I have a Bachelor's in Computer Science from Jerusalem College of Engineering, and I completed certifications in Data Analytics and Data Science.

Currently, at ELK Education Consultants Pvt Ltd, I developed a product classification system using CNNs and LSTMs, achieving a 91% accuracy. I also predicted gene disorders in children with a Random Forest model, reaching 93% accuracy. My skills include Python, TensorFlow, Keras, Scikit-Learn, OpenCV, and SpaCy, among others. I am proficient in data preprocessing, feature extraction, and model optimization.

Previously, I worked as an Application Support Specialist at PixStone Images and a Digital Associate at Amazon, focusing on data collection and annotation.

I am passionate about leveraging data to create impactful solutions and continuously improving my technical skills. I look forward to discussing how I can contribute to your team.

Thank you.

Genetic Disorder in Children

Project Overview:

Brief Description: "I worked on a project to predict gene disorders in children using machine learning techniques. The goal was to aid in early diagnosis and intervention, which can significantly improve treatment outcomes."

Problem Statement

Motivation:

Importance: "Early diagnosis of genetic disorders in children is crucial as it allows for timely intervention and treatment, which can prevent complications and improve quality of life."

Data Handling

Data Preprocessing and EDA:

Steps Taken: "We started with data preprocessing, where we cleaned and prepared the dataset for analysis. This included handling missing values, normalizing the data, and encoding categorical variables. We also performed exploratory data analysis (EDA) to understand the distribution and relationships within the data, which helped us in feature selection and understanding the underlying patterns."

Model Development

Model Selection and Training:

Choice of Model: "We chose a Random Forest model for its robustness and ability to handle complex, non-linear relationships within the data."

Hyperparameter Tuning: "We then optimized the model's performance through hyperparameter tuning. This involved adjusting parameters such as the number of trees, the depth of each tree, and the minimum number of samples required to split a node, to enhance the model's accuracy and reduce overfitting."

Tools and Technologies

Technologies Used:

Languages and Libraries: "We used Python for programming, Pandas for data manipulation, Matplotlib for visualization during EDA, and Scikit-Learn for building and evaluating the machine learning model."

Results and Achievements

Outcome:

Accuracy: "Our model achieved a test accuracy of 93%, which indicates a high level of precision in predicting gene disorders."

Impact: "This high accuracy suggests that our model can be a valuable tool for healthcare professionals, potentially improving early diagnosis and treatment planning for children with genetic disorders."

Product Classification

In this project, I developed an advanced product classification system that leverages both image and text data to categorize products accurately. This system combines the strengths of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to achieve a high classification accuracy.

Detailed Explanation:

Objective:

The goal was to create a system capable of accurately classifying products using both their images and textual descriptions.

Preprocessing:

Text Data: I performed data cleaning to remove noise, tokenization to break down text into manageable pieces, and normalization to standardize the data.

Image Data: I used image augmentation techniques such as rotation, scaling, and flipping to increase the diversity and robustness of the image dataset.

Model Architecture:

CNN for Images: CNNs were used to extract meaningful features from the product images. This involved multiple layers of convolutional operations to identify patterns and characteristics within the images.

LSTM for Text: LSTMs were employed to process and understand the sequential nature of product descriptions, capturing context and relationships between words.

Integration and Training:

The outputs from the CNN and LSTM models were combined to form a comprehensive feature set that represents both the visual and textual information of the products.

This combined feature set was then used to train a classifier, resulting in a model that can accurately categorize products based on their images and descriptions.

Achievements:

The final model achieved a classification accuracy of 91% on the test dataset, demonstrating its effectiveness in accurately categorizing products using multimodal data.

Integration of CNN and LSTM Models:

Feature Extraction:

CNN for Images:

The CNN processes the product images through multiple convolutional layers, pooling layers, and possibly some fully connected layers.

These layers extract high-level features from the images, capturing important visual patterns and characteristics.

The output from the CNN is a feature vector that encapsulates the visual information of the product.

LSTM for Text:

The LSTM processes the product descriptions, which are first tokenized and transformed into numerical representations (e.g., word embeddings).

The LSTM layers capture the sequential nature and context of the text, understanding relationships between words and phrases.

The output from the LSTM is a feature vector that encapsulates the textual information of the product.

Combining Features:

The feature vectors from the CNN and LSTM are concatenated into a single unified feature vector.

This combined feature vector represents both the visual and textual information of the product, providing a richer and more comprehensive understanding.

Fully Connected Layers:

The concatenated feature vector is then passed through one or more fully connected (dense) layers.

These layers learn to combine the features from the images and text effectively, identifying patterns that correlate with the product categories.

Output Layer:

The final fully connected layer outputs a probability distribution over the possible product categories.

Typically, a softmax activation function is used in the output layer for multi-class classification problems.

Training the Integrated Model:

Loss Function: A suitable loss function (e.g., categorical cross-entropy) is used to measure the difference between the predicted and actual categories.

Optimization: An optimizer (e.g., Adam, SGD) adjusts the weights of the network during training to minimize the loss.

Backpropagation: The gradients are computed and propagated back through both the CNN and LSTM parts of the network, updating the weights to improve accuracy.

Stock Price Prediction

Project Overview

Objective: The goal of this project is to predict stock prices by combining financial data and sentiment analysis of news headlines.

Approach: We use a two-part model:

BERT: To generate embeddings from news headlines.

LSTM: To predict stock prices using these embeddings along with traditional financial data.

2. Data Collection

Financial Data: Historical stock prices, trading volume, financial ratios, and other relevant financial indicators.

News Headlines: Daily news headlines related to the companies whose stock prices we aim to predict.

3. Data Preprocessing

Financial Data:

Normalization/standardization of numerical data.

Handling missing values.

Feature engineering to create additional relevant features (e.g., moving averages, volatility indicators).

News Headlines:

Text cleaning: Removing stopwords, punctuation, and other irrelevant text.

Tokenization and input preparation for BERT.

4. BERT for News Embeddings

BERT Model:

Use a pre-trained BERT model to convert news headlines into embeddings.

These embeddings capture the sentiment and semantic meaning of the news.

5. Combining Financial Data and News Embeddings

Integration:

Concatenate BERT embeddings with financial data features.

This combined feature set provides a richer context for the LSTM model.

6. LSTM for Price Prediction

LSTM Model:

Long Short-Term Memory (LSTM) networks are suitable for time series prediction.

Use the combined feature set (financial data + news embeddings) as input to the LSTM.

Train the LSTM model to predict future stock prices.

7. Model Training and Evaluation

Training:

Split the dataset into training and validation sets.

Train the model using appropriate loss functions and optimization algorithms.

Evaluation:

Evaluate model performance using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE).

Perform backtesting to assess model performance in a real-world scenario.

8. Results and Insights

Model Performance:

Discuss the accuracy and robustness of the model predictions.

Highlight any significant findings or patterns discovered during the analysis.