Thyroid Cancer Recurrence Prediction Using Deep Learning

1. **Introduction**

This project focuses on developing a predictive system for thyroid cancer recurrence using a deep learning model. Cancer recurrence prediction is crucial for planning effective post-treatment monitoring and ensuring prompt interventions. Using a structured dataset containing various patient health records and cancer-related attributes, we aim to create a reliable model to forecast the chances of recurrence.

**Objectives:**

1. Predict thyroid cancer recurrence using a deep neural network.
2. Evaluate the model's performance using metrics like accuracy, precision, recall, and F1-score.
3. Visualize the data and model performance to gain insights and identify areas for improvement.
4. **Data Overview**

The dataset consists of several features related to patients' demographics, health conditions, and cancer characteristics, such as:

* **Age**: Age at the time of cancer diagnosis.
* **Gender**: Gender of the patient.
* **Smoking History**: Whether the patient has a history of smoking.
* **Hx Radiotherapy**: History of radiotherapy treatment.
* **Thyroid Function**: Status of the patient's thyroid function.
* **Physical Examination**: Findings from a clinical examination.
* **Adenopathy**: Presence of lymph node abnormalities.
* **Pathology**: Type of thyroid cancer.
* **Focality**: Whether the cancer is unifocal or multifocal.
* **Risk Assessment**: Cancer risk classification.
* **TNM Classification**: Tumor, Node, and Metastasis stages.
* **Stage**: Overall cancer stage.
* **Response to Treatment**: Indicates how well the cancer responded to treatment.
* **Recurred**: The target variable indicating if the cancer recurred.

1. **Data Preprocessing**

The following preprocessing steps were performed: -

* **Handling Missing Values**: Replaced missing values with the mean for numerical features.

*data.fillna(data.mean(), inplace=True)*

* **Encoding Categorical Variables**: Transformed categorical variables into numeric format using *LabelEncoder*.

*label\_encoders = {}*

*for col in categorical\_columns:*

*le = LabelEncoder()*

*data[col] = le.fit\_transform(data[col])*

*label\_encoders[col] = le*

* **Feature Scaling**: Used StandardScaler to normalize numerical features.

*scaler = StandardScaler()*

*X\_train = scaler.fit\_transform(X\_train)*

*X\_test = scaler.transform(X\_test)*

1. **Exploratory Data Analysis (EDA)**

The data was visualized to understand the distribution of features and relationships:

* **Age Distribution**: Analyzed the spread of patient ages using a histogram.

*plt.figure(figsize=(12, 6))*

*sns.histplot(data['Age'], kde=True, bins=30, color='blue')*

*plt.title('Age Distribution')*

*plt.xlabel('Age')*

*plt.ylabel('Frequency')*

*plt.show()*

A graph of age distribution

Description automatically generated

* **Feature Correlation**: Used a heatmap to understand the correlations between features, aiding in feature selection and interpretation.

*plt.figure(figsize=(14, 10))*

*sns.heatmap(data.corr(), annot=True, cmap='coolwarm', fmt=".2f")*

*plt.title('Feature Correlation Heatmap')*

*plt.show()*

A chart with red and blue squares

Description automatically generated

1. **Model Architecture**

A deep learning model was developed using TensorFlow and Keras:

* **Input Layer**: Specifies the input shape based on the number of features.
* **Hidden Layers**: Two dense layers with 128 and 64 neurons, respectively, using ReLU activation. Dropout layers (30%) are added to prevent overfitting.
* **Output Layer**: A single neuron with a sigmoid activation for binary classification.

*model = keras.Sequential([ keras.layers.Input(shape=(X\_train.shape[1],)), keras.layers.Dense(128, activation='relu'), keras.layers.Dropout(0.3), keras.layers.Dense(64, activation='relu'), keras.layers.Dropout(0.3), keras.layers.Dense(1, activation='sigmoid') ])*

1. **Model Training and Evaluation**

* **Compilation**: The model is compiled with the adam optimizer and binary\_crossentropy loss.

*model.compile(optimizer='adam',*

*loss='binary\_crossentropy',*

*metrics=['accuracy'])*

* **Training**: Trained for 100 epochs with a validation split of 20% to monitor performance.

*history = model.fit(X\_train, y\_train, epochs=100, batch\_size=32, validation\_split=0.2)*

* **Metrics**: Evaluated using accuracy, precision, recall, and F1-score. Also visualized the confusion matrix to assess model performance.

A graph of a graph

Description automatically generated

A graph showing a loss

Description automatically generated

1. **Results:**

The model achieved good performance on the test set:

* **Accuracy**: 0.9478260869565217
* **Precision**: 0.95
* **Recall**: 0.95
* **F1-Score**: 0.95

A blue squares with white text

Description automatically generatedA screenshot of a computer

Description automatically generated

1. **Discussion**

The deep learning model provided robust predictions for thyroid cancer recurrence. However, there are areas for potential improvement:

* **Model Tuning**: Adjusting the architecture or hyperparameters could yield better results.
* **Imbalanced Data**: If the dataset is imbalanced, techniques like oversampling or SMOTE could be applied to improve performance.

1. **Conclusion**

This project demonstrated the use of a deep learning approach to predict thyroid cancer recurrence. The model, supported by rigorous preprocessing and EDA, showed promising results. Further improvements and testing on real-world data are necessary for clinical application.