Project Report: Predicting Patient Survival for Colorectal Cancer

1. Introduction

Colorectal cancer stands as a significant health challenge globally, ranking among the most prevalent cancers. The ability to accurately predict a patient's survival prospects is of paramount importance. Such predictions empower clinicians to make more informed decisions regarding treatment strategies, facilitate more effective patient counseling, and aid in the development of personalized treatment plans. This project endeavors to construct a predictive model leveraging comprehensive patient data to estimate the likelihood of a patient surviving beyond a 12-month period following their diagnosis.

2. Dataset

The dataset employed for this project, `colorectal_cancer_prediction.csv`, encompasses a rich array of patient-related features. These include demographic information such as age, gender, and race, alongside critical clinical parameters like tumor aggressiveness and the stage of the cancer at diagnosis. Furthermore, the dataset incorporates lifestyle factors, including smoking habits, alcohol consumption, and dietary information, as well as details on treatment history, such as surgery, chemotherapy, and radiotherapy. Finally, data on follow-up care and screening practices are also included. The target variable, `Survival_Status`, clearly indicates whether a patient's outcome was "Deceased" or "Survived".

Key Features:

- Age, Gender, Race
- Tumor Aggressiveness, Stage at Diagnosis
- Lifestyle factors: Smoking, Alcohol, Diet
- Treatment history: Surgery, Chemotherapy, Radiotherapy
- Follow-up and screening behavior

3. Data Preprocessing

The initial phase of the project involved meticulous data preprocessing to ensure the dataset was suitable for model training. The steps undertaken include:

- Loading the `colorectal_cancer_prediction.csv` file into a Pandas DataFrame, a fundamental data structure for data manipulation in Python.
- Renaming the columns to establish a consistent and easily understandable naming convention throughout the analysis.
- Converting the categorical labels of the target variable, "Deceased" and "Survived", into a numerical representation (0 and 1, respectively), which is required by most machine learning algorithms.
- Applying one-hot encoding to the categorical features within the dataset. This technique transforms categorical variables into a numerical format without introducing ordinal relationships, which can be misinterpreted by the model.
- Dropping the `Patient_ID` column, as this identifier is unique to each patient and does not contribute predictive information to the survival outcome.

4. Model Training

A **Logistic Regression** model was selected as the predictive algorithm for this project. Logistic Regression is a widely used statistical model that, despite its simplicity, can perform effectively in binary classification tasks like predicting survival.

Steps:

- The preprocessed dataset was partitioned into two distinct subsets: a training set, comprising 80% of the data, used to train the model, and a testing set, comprising the remaining 20%, used to evaluate the model's performance on unseen data.
- The `LogisticRegression` class from the `sklearn.linear_model` module in Python's Scikit-learn library was instantiated and trained using the training dataset.
- The trained model's performance was rigorously evaluated using standard classification metrics, including accuracy and a comprehensive classification report, which provides insights into precision, recall, and F1-score for both survival classes.

Results:

- **Accuracy**: The Logistic Regression model achieved an overall accuracy of approximately 75.15% on the testing data, indicating that it correctly predicted the survival status for a significant proportion of the patients.
- **Precision & Recall**: The classification report revealed that the model exhibited a high recall for the “Survived” class, meaning it was effective at identifying patients who survived. However, the model demonstrated lower precision and recall for the

“Deceased” class. This discrepancy is likely attributable to the inherent class imbalance present in the dataset, where the number of surviving patients significantly outweighs the number of deceased patients.

5. Data Visualization

To gain a visual understanding of the distribution of the target variable, a count plot was generated using the Seaborn library:

Python
```
sns.countplot(data=df, x='Survival_Status', palette='Set2')
plt.title('Survival Status Distribution')
```

This visualization clearly illustrates the imbalance in the dataset, with a higher count of "Survived" cases compared to "Deceased" cases. This visual confirmation supports the observation made during model evaluation regarding the potential impact of class imbalance.

6. Conclusion

- The Logistic Regression model developed in this project demonstrated a promising overall accuracy of approximately 75% in predicting patient survival beyond 12 months. The model showed particular strength in correctly identifying patients who survived.
- A notable challenge identified during the analysis is the significant class imbalance within the dataset, where the number of survivors considerably exceeds the number of deceased patients. This imbalance likely contributed to the model's comparatively weaker performance in accurately predicting the "Deceased" class.
- To enhance the model's predictive capabilities in future iterations, several improvements can be explored:
    - Implementing resampling techniques such as SMOTE (Synthetic Minority Over-sampling Technique) to artificially balance the number of instances in each class, potentially improving the model's ability to learn from the minority class.
    - Investigating and employing more sophisticated machine learning models, such as gradient boosting algorithms like XGBoost or ensemble methods like Random Forests, which may be better equipped to handle complex relationships within the data and class imbalances.
    - Conducting hyperparameter tuning to optimize the parameters of the chosen model, potentially leading to improved performance. Additionally, exploring feature selection techniques could help identify the most relevant predictors of survival and potentially simplify the model while improving its generalization ability.

7. Technologies Used

- Python (Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn)
- Jupyter Notebook
