## Project Documentation: Cancer Prediction

### Project Title: Cancer Prediction

### 1. Business Problem

In cancer diagnosis, there can often be ambiguity in determining whether a tumor is benign (non-cancerous) or malignant (cancerous). Even with skilled professionals like radiologists, pathologists, and oncologists, errors in diagnosis may occur due to the complexity of medical imaging and histopathology analysis. To enhance diagnostic accuracy, a team of physicians seeks to develop an AI application that will confidently predict the presence of cancer in patients, serving as a support tool to improve diagnostic precision.

#### Business Objective

- \*\*Maximize Cancer Detection\*\*: Detect the presence of cancer with high accuracy.

#### Business Constraints

- \*\*Minimize Treatment Cost\*\*: Ensure cost-effective diagnosis to support patient affordability.

- \*\*Maximize Patient Convenience\*\*: Offer a fast, reliable diagnostic tool that minimizes wait time and improves patient experience.

#### Success Criteria

1. \*\*Business Success\*\*: Achieve a minimum correct diagnosis rate of 96% for cancer cases.

2. \*\*Machine Learning Success\*\*: Attain at least 98% accuracy in cancer predictions.

3. \*\*Economic Success\*\*: A reduction in medical expenses leading to increased patient trust and a projected hospital revenue boost of at least 12%.

### 2. Data Collection

Data was sourced from the hospital, consisting of 569 records representing patient data. The dataset includes 30 feature columns, with each feature providing insights into cell nucleus characteristics, and a target label indicating tumor type (Benign or Malignant).

#### Feature Descriptions

Each cell nucleus is characterized by 10 primary attributes, calculated as real-valued metrics:

- \*\*Radius\*\*: Mean distance from center to perimeter points.

- \*\*Texture\*\*: Standard deviation of gray-scale values.

- \*\*Perimeter\*\*: Measurement of the nucleus boundary length.

- \*\*Area\*\*: Size of the nucleus.

- \*\*Smoothness\*\*: Local variation in radius lengths.

- \*\*Compactness\*\*: Defined as \((\text{perimeter}^2 / \text{area}) - 1.0\).

- \*\*Concavity\*\*: Severity of the concave sections of the contour.

- \*\*Concave Points\*\*: Count of concave sections in the contour.

- \*\*Symmetry\*\*: Symmetry of the nucleus shape.

- \*\*Fractal Dimension\*\*: An approximation of the contour complexity (often described as a “coastline” approximation).

### 3. Project Workflow

#### Step 1: Import Necessary Libraries

Key libraries for data manipulation, visualization, machine learning, and statistical analysis were imported:

- \*\*Pandas\*\* and \*\*NumPy\*\* for data handling and processing.

- \*\*Scikit-Learn\*\* for modeling, evaluation metrics, and preprocessing.

- \*\*Statsmodels\*\* for statistical insights.

- \*\*RandomForestClassifier\*\* for building the primary classification model.

#### Step 2: Load and Explore the Dataset

The dataset was loaded from a local directory, and the first few rows were examined to verify structure and completeness. Exploratory Data Analysis (EDA) was conducted to gain insights into the data:

1. \*\*Basic Data Information\*\*: Examined data types, null values, and unique counts per column.

2. \*\*Descriptive Statistics\*\*: Analyzed distribution, mean, standard deviation, and other descriptive statistics for numeric columns.

3. \*\*Feature Engineering\*\*:

- Transformed the target variable values (“B” for Benign and “M” for Malignant) to improve clarity.

- Removed unnecessary columns to streamline the dataset.

4. \*\*Missing Values\*\*: Assessed and addressed any missing values via imputation methods.

5. \*\*Duplicate Records\*\*: Identified and removed duplicate entries to ensure data accuracy.

6. \*\*Variable Partitioning\*\*: Divided columns into numeric and categorical features and separated input (features) and output (target) variables.

#### Step 3: Data Preprocessing

A preprocessing pipeline was designed to convert the data into a suitable format for model training:

1. \*\*Outlier Handling\*\*: Used Winsorization to cap extreme values, thus reducing the impact of outliers.

2. \*\*Imputation\*\*: Performed mean imputation for missing numeric values.

3. \*\*Mapping and Encoding\*\*: Utilized \*\*DataFrameMapper\*\* to map attributes within the pipeline, encoding categorical variables to numerical format using OneHotEncoding.

4. \*\*Scaling\*\*: Applied \*\*MinMaxScaler\*\* to normalize numeric values to a 0–1 range.

5. \*\*Data Concatenation\*\*: Combined numeric and OneHotEncoded features into a single DataFrame to facilitate model training.

#### Step 4: Data Partitioning

The dataset was split into training and testing sets:

- \*\*80%\*\* of the data was used for training the model, and \*\*20%\*\* for testing. This partitioning ensures that the model learns from a substantial portion of the data, leaving an adequate subset for evaluation.

#### Step 5: Model Initialization and Training

A \*\*RandomForestClassifier\*\* was initialized as the primary model due to its robustness in handling classification tasks and its ability to model non-linear relationships. The model was trained on the training dataset to capture patterns indicative of cancer presence based on input features.

#### Step 6: Model Prediction

Predictions were generated using the trained model on the test dataset to assess its generalization ability in classifying new data points.

#### Step 7: Model Evaluation

Evaluation metrics were calculated to measure model performance:

- \*\*Accuracy\*\*: Overall correctness of the model in classifying tumors.

- \*\*Classification Report\*\*: Provided detailed precision, recall, and F1 scores, enabling insights into the model’s ability to identify both benign and malignant cases effectively.

#### Step 8: Hyperparameter Tuning with RandomizedSearchCV

To optimize model performance, hyperparameter tuning was performed:

1. \*\*Parameter Setup\*\*: Defined a set of parameters to test various model configurations, such as the number of trees, maximum tree depth, and minimum samples required to split nodes.

2. \*\*RandomizedSearchCV Execution\*\*: RandomizedSearchCV was configured with 100 iterations, testing different random parameter combinations to find the most effective setup.

#### Step 9: Fit the Optimized Model

The best parameter configuration from RandomizedSearchCV was used to re-train the model. This optimization improved the model’s accuracy and generalization ability.

#### Step 10: Evaluate the Tuned Model

The optimized model was then evaluated on the testing dataset to validate its effectiveness. Performance metrics were recorded to verify improvements achieved through tuning.