SUMMER TRAINING/INTERNSHIP PROJECT REPORT

(Term June - July 2025)

SKIN CANCER DETECTION SYSTEM USING ML MODELS

Submitted by

Himanshu Kumar
Registration Number: 12318169
&
Vansh
Registration Number: 12317837

Course Code : PETV79

Under the Guidance of
Mahipal Singh Papola

School of Computer Science and Engineering
Lovely Professional University, Punjab

Lovely Professional University, Punjab

BONAFIDE CERTIFICATE

Certified that this project report "**SKIN CANCER DETECTION SYSTEM**" is the Bonafide work of "VANSH and HIMANSHU KUMAR" who carried out the project work under my supervision

SIGNATURE

<<Name of Supervisor>>

Himanshu Kumar, Reg. No. 12318169

Vansh, Reg. No. 12317837

SIGNATURE

<<Signature of the Head of the Department>>

SIGNATURE

<<Name>>
HEAD OF THE DEPARTMENT

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who have supported and guided me in the successful completion of this project titled “Skin Cancer Detection System”.

First and foremost, I extend my heartfelt thanks to my project guide Mahipal Singh Papola, for their valuable guidance, constant encouragement, and timely suggestions throughout the course of this project.

I would also like to thank Head of the Department for providing me with the necessary facilities and support.

My special thanks to all the faculty members and staff of the School of Computer Science and Engineering for their cooperation and support.

Lastly, I am deeply grateful to my family and friends for their moral support, patience, and encouragement throughout the project.

Himanshu Kumar & Vansh
12318169 & 12317837

TABLE OF CONTENTS

INTRODUCTION

Skin cancer represents one of the most prevalent malignancies globally, with incidence rates continuing to rise due to factors such as increased UV exposure, aging populations, and improved diagnostic awareness. Early detection remains the cornerstone of successful treatment, significantly improving patient outcomes and survival rates. However, traditional diagnostic approaches rely heavily on clinical expertise and visual examination, which can be subjective, time-intensive, and prone to human error, particularly in resource-limited settings.

The emergence of machine learning and artificial intelligence technologies has opened new avenues for medical diagnosis, offering the potential to augment clinical decision-making with objective, data-driven insights. These computational approaches can analyze complex patterns in patient data that may not be immediately apparent to human observers, potentially identifying subtle indicators of malignancy.

This project presents a comprehensive machine learning-based system for skin cancer detection, leveraging patient demographic information and clinical features to predict cancer presence. By implementing and comparing seven different classification algorithms – including Logistic Regression, Random Forest, Support Vector Machine, Gradient Boosting, K-Nearest Neighbors, Decision Tree, and Naive Bayes – we aim to identify the most effective approach for binary skin cancer prediction. The system incorporates robust data preprocessing, cross-validation techniques, and multiple evaluation metrics to ensure reliable performance assessment, ultimately providing a foundation for computer-aided diagnostic tools that could support healthcare professionals in clinical practice.

Furthermore, the comparative analysis framework enables systematic evaluation of algorithm strengths and weaknesses, providing valuable insights for model selection in medical applications. The automated model selection process identifies the optimal algorithm based on F1-score optimization, balancing both sensitivity and specificity – crucial factors in medical diagnostic applications where both false positives and false negatives carry significant clinical implications.

This research contributes to the growing field of medical AI by demonstrating the feasibility and effectiveness of machine learning approaches in skin cancer detection, ultimately providing a foundation for computer-aided diagnostic tools that could support healthcare professionals in clinical practice while potentially improving diagnostic accuracy and reducing healthcare disparities.

DATASET DESCRIPTION

This dataset contains 10,000 synthetic records generated to simulate various risk factors associated with skin cancer. It is designed for educational use and to help learners and researchers apply machine learning and data science techniques, especially in binary classification tasks.

The dataset includes commonly studied features such as age, gender, skin type, sun exposure, and lesion characteristics (like diameter, color variation, border irregularity). The target variable, Skin Cancer, indicates whether a person is at risk (1) or not at risk (0) of developing skin cancer.

Features:

Age: Age of the individual (18–90 years)

Gender: Biological sex (Male/Female)

Skin Type: Fitzpatrick skin types (I to VI)

Sun Exposure: Self-reported sun exposure level (Low, Moderate, High)

Family History: Family history of skin cancer (Yes/No)

Mole Count: Number of moles on the body

Itchiness: Itching in the lesion (Yes/No)

Bleeding: Bleeding from the lesion (Yes/No)

Asymmetry: Lesion shape asymmetry (Yes/No)

Border Irregularity: Irregular lesion borders (Yes/No)

Color Variation: Color variation in the lesion (Yes/No)

Diameter mm: Diameter of the lesion in millimeters

Evolution: Recent change in the lesion (Yes/No)

Skin Cancer: Target label (1 = At risk of skin cancer, 0 = Not at risk)

METHODS/TECHNIQUES APPLIED AND THEIR BRIEF DESCRIPTION

We use the following classification models for this project, these are mentioned below

- Logistic Regression
- Random Forest
- Support Vector Machine (SVM)
- Gradient Boosting
- K-Nearest Neighbors
- Decision Tree
- Naive Bayes

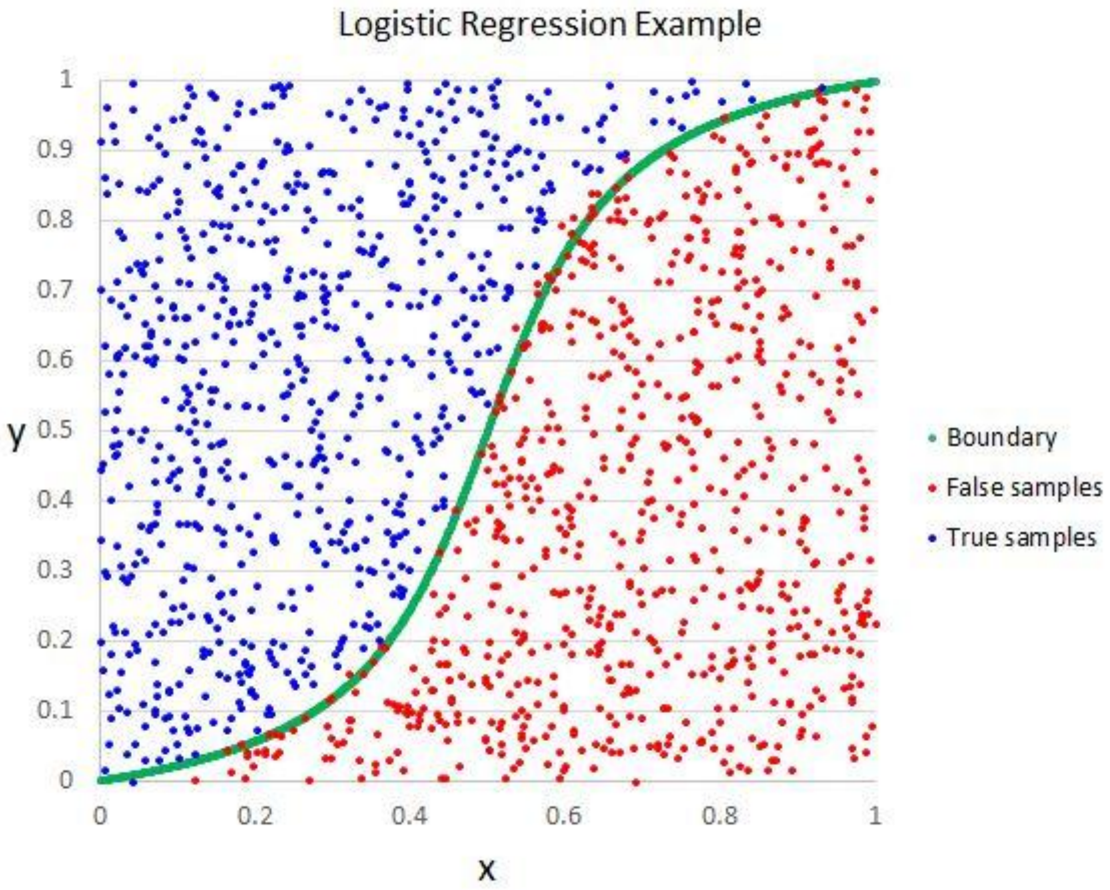
Description about the different models:-

1. Logistic Regression

Logistic regression is a statistical method used for binary and multiclass classification problems. Unlike linear regression, it uses the logistic function (sigmoid) to map any real-valued input to a value between 0 and 1, making it suitable for probability estimation. The algorithm finds the best-fitting relationship between the dependent variable and independent variables by maximizing the likelihood function.

The core principle involves transforming the linear combination of features using the sigmoid function: p = 1/(1 + e^(-z)), where z is the linear combination of features. This creates an S-shaped curve that naturally bounds predictions between 0 and 1. The algorithm uses gradient descent or other optimization techniques to find optimal coefficients.

Logistic regression is interpretable, computationally efficient, and doesn't require feature scaling. It works well when the relationship between features and log-odds is linear, and it provides probability estimates for predictions. However, it assumes linear relationships and can struggle with complex patterns or interactions between features. It's widely used in medical diagnosis, marketing, and social sciences due to its interpretability and statistical foundation.

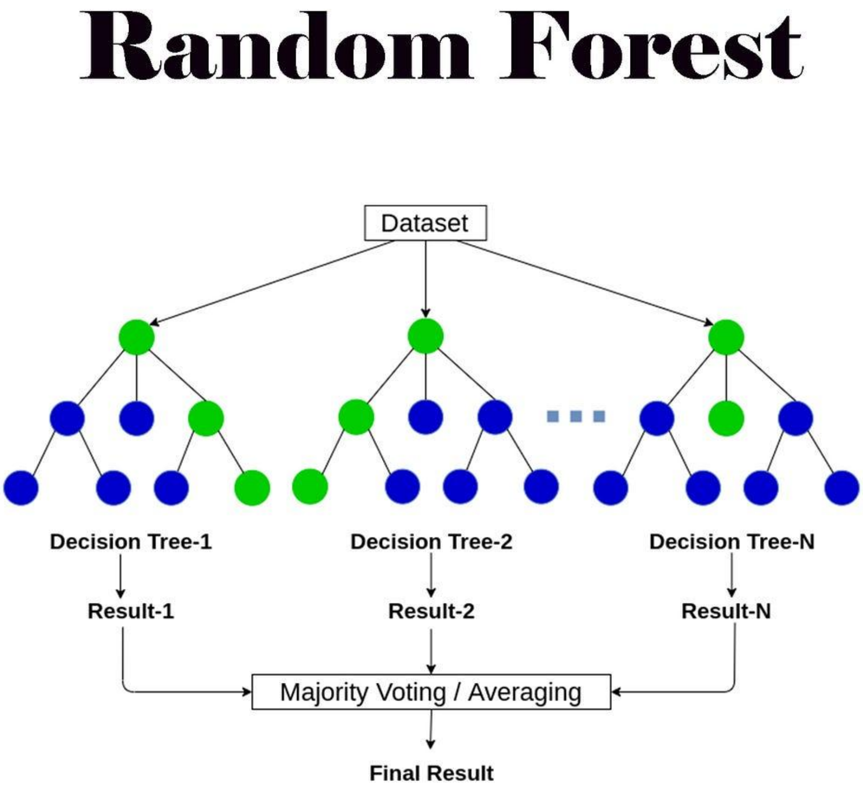


2. Random Forest Classifier

Random Forest is an ensemble learning method that combines multiple decision trees to create a more robust and accurate model. It operates on the principle of "wisdom of crowds" by training numerous decision trees on different subsets of the training data and features, then aggregating their predictions through voting (classification) or averaging (regression).

The algorithm introduces randomness at two levels: bootstrap sampling (bagging) where each tree is trained on a random sample of data with replacement, and feature randomness where each split considers only a random subset of features. This dual randomness reduces overfitting and improves generalization. Each tree in the forest is grown deep with minimal pruning, but the ensemble effect compensates for individual tree overfitting.

Random Forest provides feature importance rankings, handles missing values naturally, and works well with default parameters. It's robust to outliers and noise, can handle both numerical and categorical features, and provides out-of-bag error estimates for validation. The algorithm is parallelizable and scales well with large datasets. However, it can be memory-intensive, less interpretable than single trees, and may overfit with very noisy data. It's widely used in bioinformatics, finance, and e-commerce for its reliability and performance.

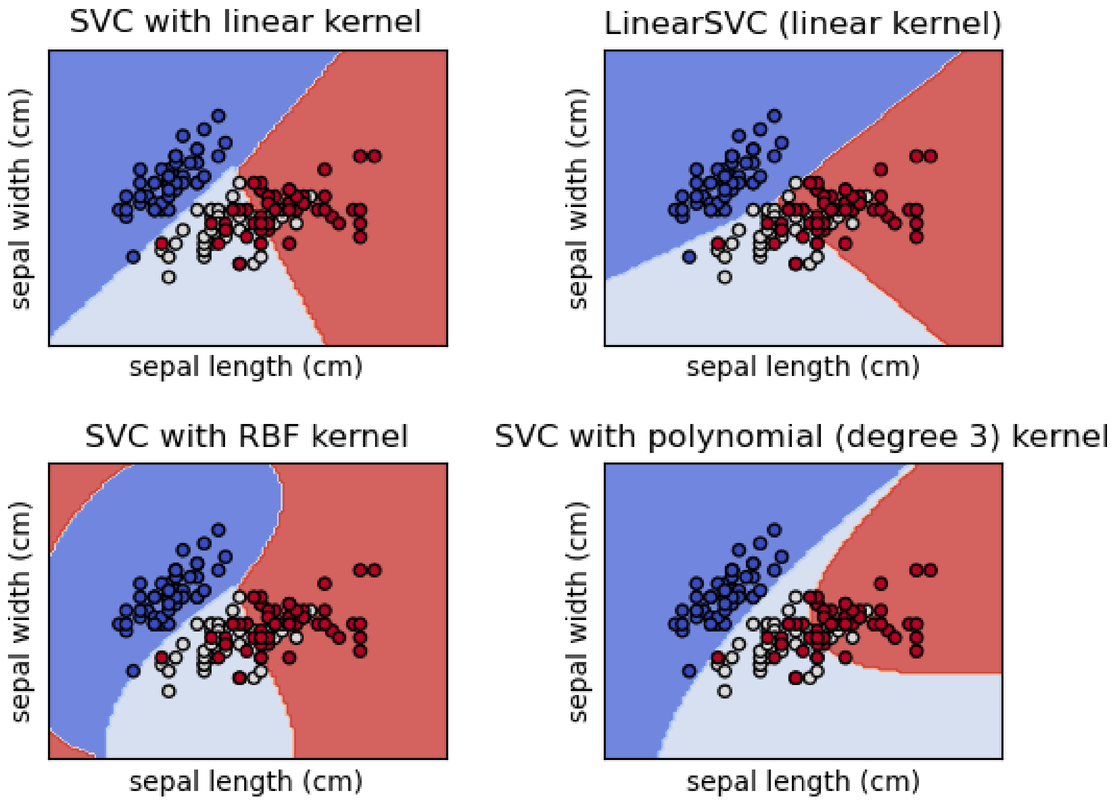


3. Support Vector Machines (SVM)

Support Vector Machine is a powerful supervised learning algorithm used for classification and regression tasks. SVM finds the optimal hyperplane that separates different classes by maximizing the margin between them. The key insight is that only the data points closest to the decision boundary (support vectors) are crucial for defining the classifier.

For linearly separable data, SVM finds the hyperplane with maximum margin. For non-linearly separable data, it uses the kernel trick to map data into higher-dimensional spaces where linear separation becomes possible. Common kernels include polynomial, radial basis function (RBF), and sigmoid. The algorithm solves a quadratic optimization problem to find the optimal hyperplane parameters.

SVM is effective in high-dimensional spaces and memory-efficient since it only uses support vectors for predictions. It's versatile through different kernel functions and works well with small datasets. The algorithm provides a unique solution and is less prone to overfitting in high-dimensional spaces. However, it doesn't provide probability estimates directly, can be sensitive to feature scaling, and training time increases significantly with large datasets. Parameter tuning (C, gamma, kernel choice) is crucial for optimal performance. SVM is widely used in text classification, image recognition, and bioinformatics.

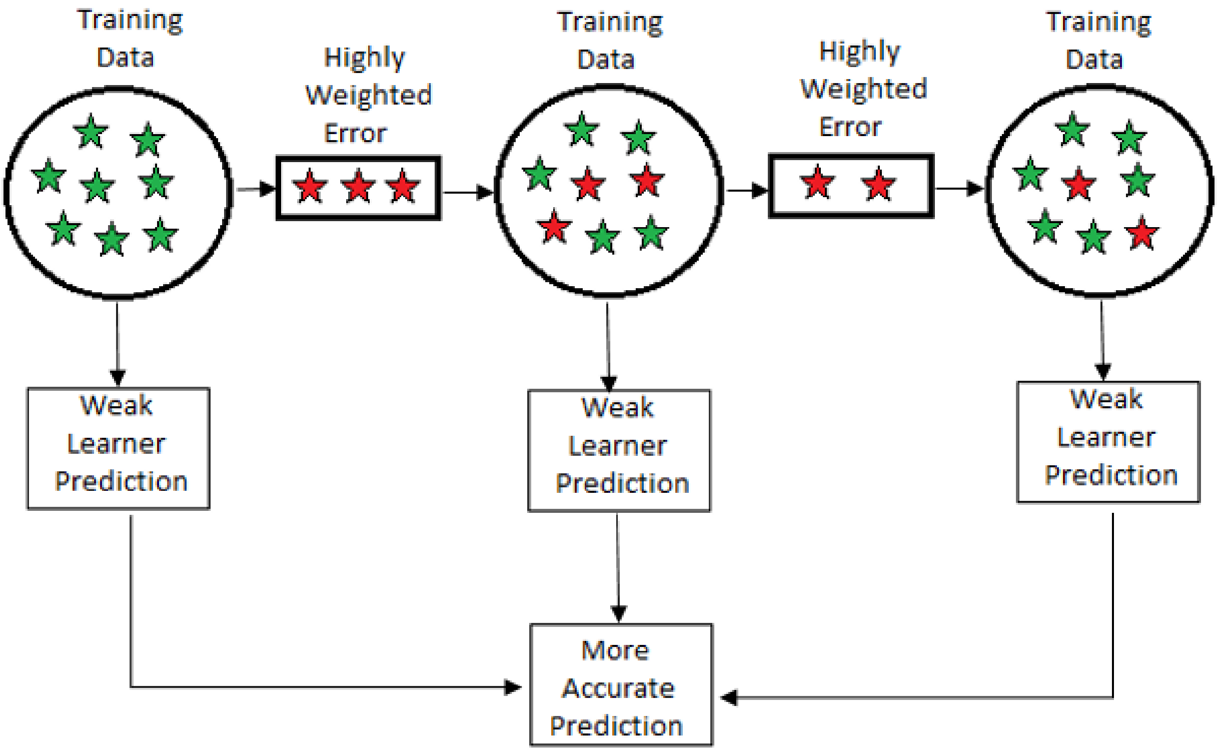


4. Gradient Boosting

Gradient Boosting is an ensemble method that builds models sequentially, where each new model corrects the errors of the previous ones. Unlike bagging methods that train models independently, boosting creates a strong learner by combining many weak learners in a sequential manner. The algorithm focuses on difficult-to-predict instances by giving them higher weights in subsequent iterations.

The process starts with a simple model (often a decision tree stump), calculates residuals (errors), then trains the next model to predict these residuals. Each subsequent model is added to the ensemble with a learning rate that controls the contribution of each model. The final prediction is the weighted sum of all models. Popular implementations include XGBoost, LightGBM, and CatBoost, which include optimizations for speed and accuracy.

Gradient Boosting often achieves state-of-the-art performance on structured data and handles mixed data types well. It provides feature importance and can capture complex patterns and interactions. The algorithm is robust to outliers and doesn't require extensive feature preprocessing. However, it's prone to overfitting, especially with noisy data, and requires careful hyperparameter tuning. Training is sequential and can be time-consuming. It's widely used in machine learning competitions and real-world applications like ranking systems and recommendation engines.

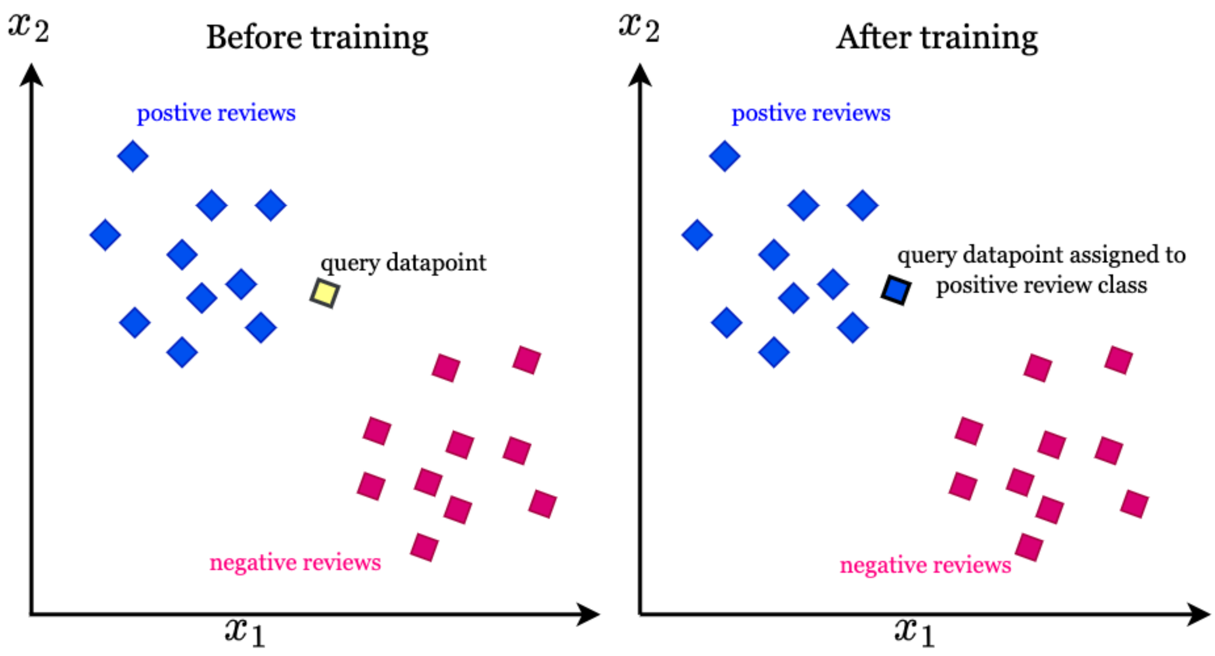


5. KNN (K-Nearest- Neighbors)

K-Nearest Neighbors is a simple, instance-based learning algorithm that makes predictions based on the K closest training examples in the feature space. It's called "lazy learning" because it doesn't build an explicit model during training but stores all training data and makes predictions at query time by finding the most similar instances.

The algorithm calculates distances (typically Euclidean, Manhattan, or Minkowski) between the query point and all training points, selects the K nearest neighbors, and makes predictions through majority voting (classification) or averaging (regression). The choice of K is crucial: small K values can lead to overfitting and noise sensitivity, while large K values can over smooth and lose local patterns.

KNN is simple to understand and implement, works well with small datasets, and can capture complex decision boundaries. It's naturally suitable for multi-class problems and can be used for both classification and regression. The algorithm makes no assumptions about data distribution and adapts well to new data. However, it's computationally expensive at prediction time, sensitive to irrelevant features and feature scaling, and suffers from the curse of dimensionality. Performance degrades with high-dimensional data and large datasets. It's commonly used in recommendation systems, pattern recognition, and as a baseline for comparison.

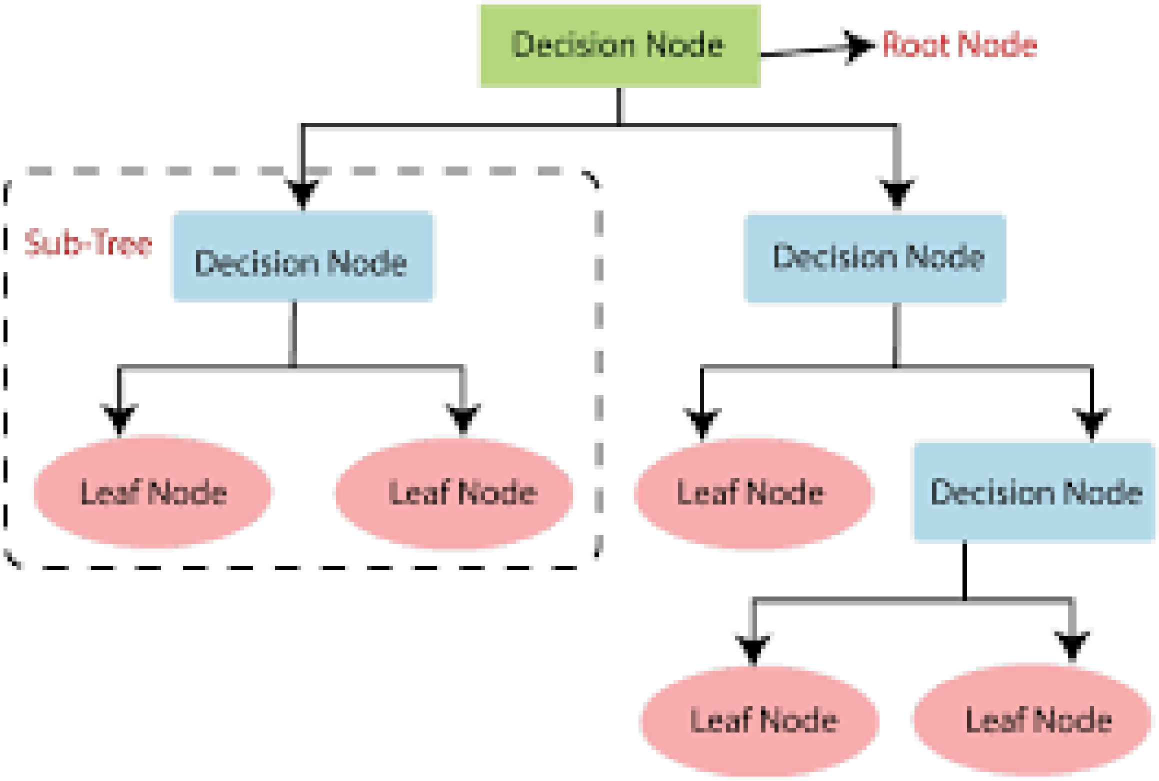


6. Decision Tree

Decision Tree is a hierarchical model that makes predictions by learning simple decision rules inferred from data features. It creates a tree-like structure where internal nodes represent feature tests, branches represent outcomes of tests, and leaf nodes represent class labels or predicted values. The algorithm recursively splits the data based on feature values that provide the best separation.

The splitting process uses criteria like Gini impurity, information gain, or mean squared error to determine the best splits. Popular algorithms include ID3, C4.5, and CART. The tree grows by selecting features and thresholds that minimize impurity in resulting subsets. To prevent overfitting, techniques like pruning, maximum depth limits, and minimum samples per leaf are employed.

Decision Trees are highly interpretable and provide clear decision paths that humans can easily understand. They handle both numerical and categorical features naturally, require minimal data preprocessing, and can capture non-linear relationships. The algorithm automatically performs feature selection and can identify important features. However, trees are prone to overfitting, can be unstable (small data changes can create very different trees), and may struggle with linear relationships. They tend to create biased trees when classes are imbalanced. Despite limitations, they're widely used in medical diagnosis, credit scoring, and as building blocks for ensemble methods.

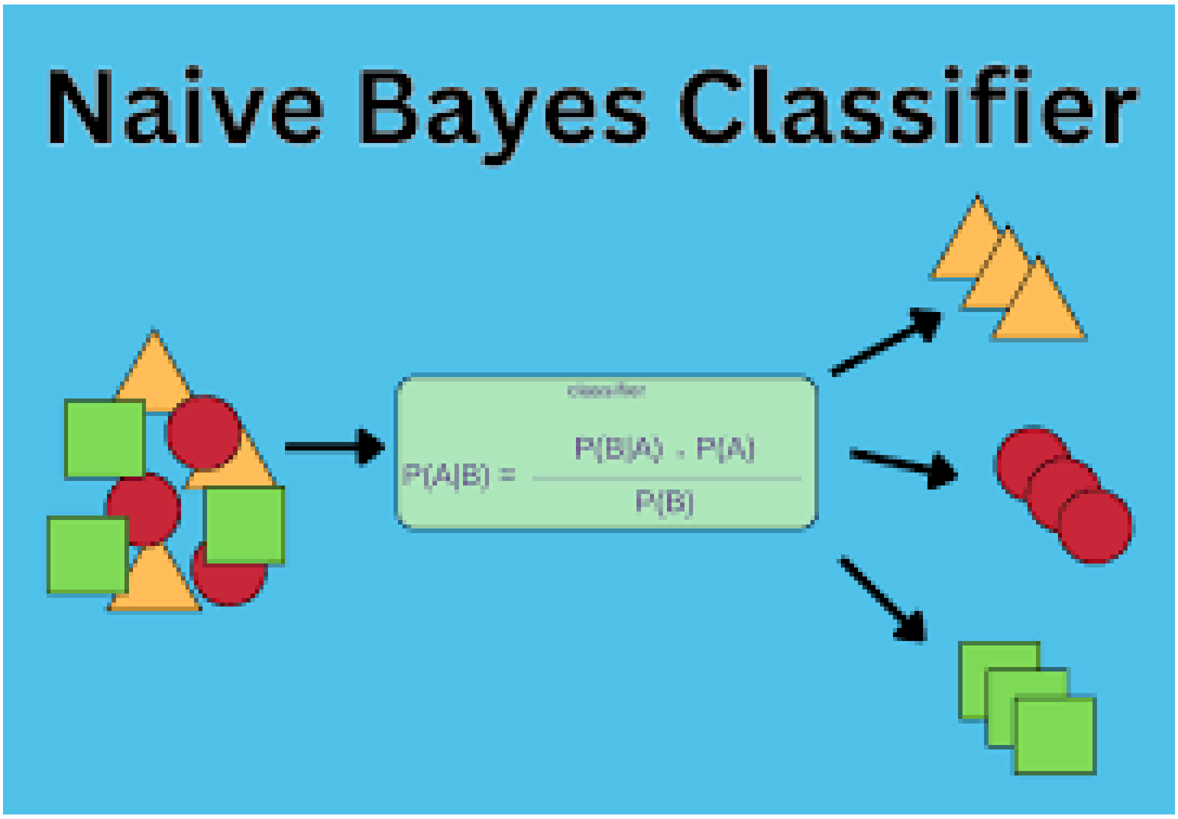


7. Naïve Bayes Classifier

Naive Bayes is a probabilistic classifier based on Bayes' theorem with a "naive" assumption of feature independence. Despite this strong assumption rarely holding in practice, the algorithm often performs surprisingly well. It calculates the probability of each class given the features and predicts the class with the highest probability.

The algorithm applies Bayes' theorem: P(class|features) = P(features|class) × P(class) / P(features). The "naive" assumption treats all features as conditionally independent given the class, simplifying calculations to the product of individual feature probabilities. Different variants exist: Gaussian Naive Bayes for continuous features, Multinomial for discrete counts, and Bernoulli for binary features.

Naive Bayes is computationally efficient, requiring only one pass through the data for training. It handles multi-class classification naturally, works well with small datasets, and is relatively immune to irrelevant features. The algorithm provides probability estimates and can be updated incrementally with new data. It's particularly effective for text classification and spam filtering. However, the independence assumption can hurt performance when features are highly correlated, and it can be sensitive to skewed data. The algorithm may assign zero probability to unseen feature combinations. Despite its simplicity, it's widely used in text mining, sentiment analysis, and real-time predictions due to its speed and effectiveness.

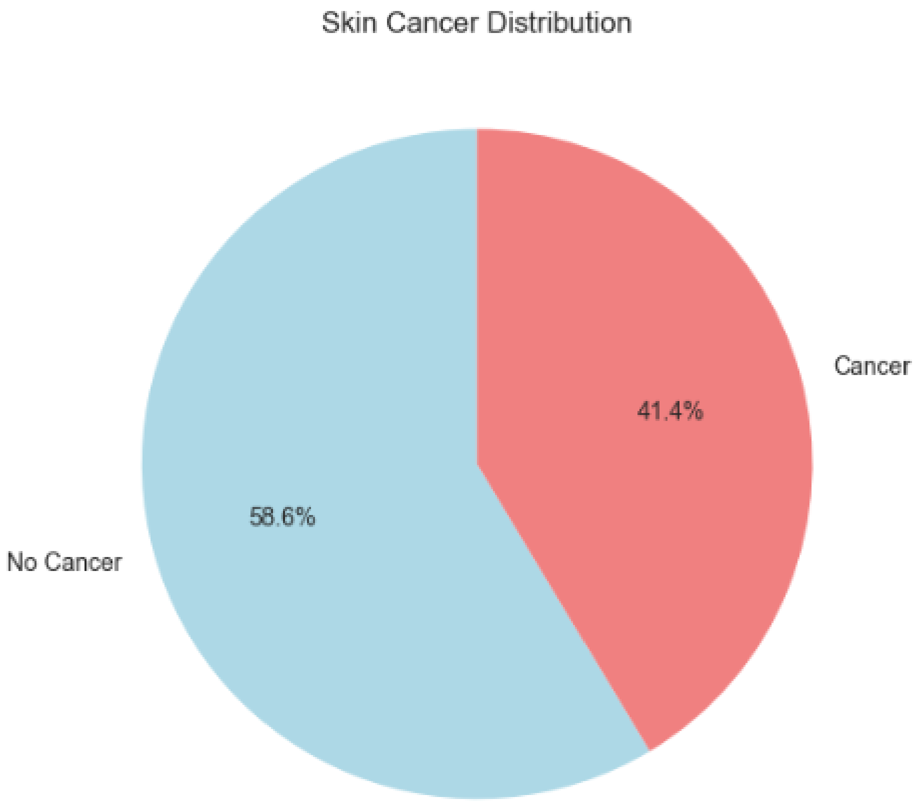


DATA VISUALIZATION AND FEATURE ANALYSIS

1. Data Visualization

Target Variable Distribution Analysis

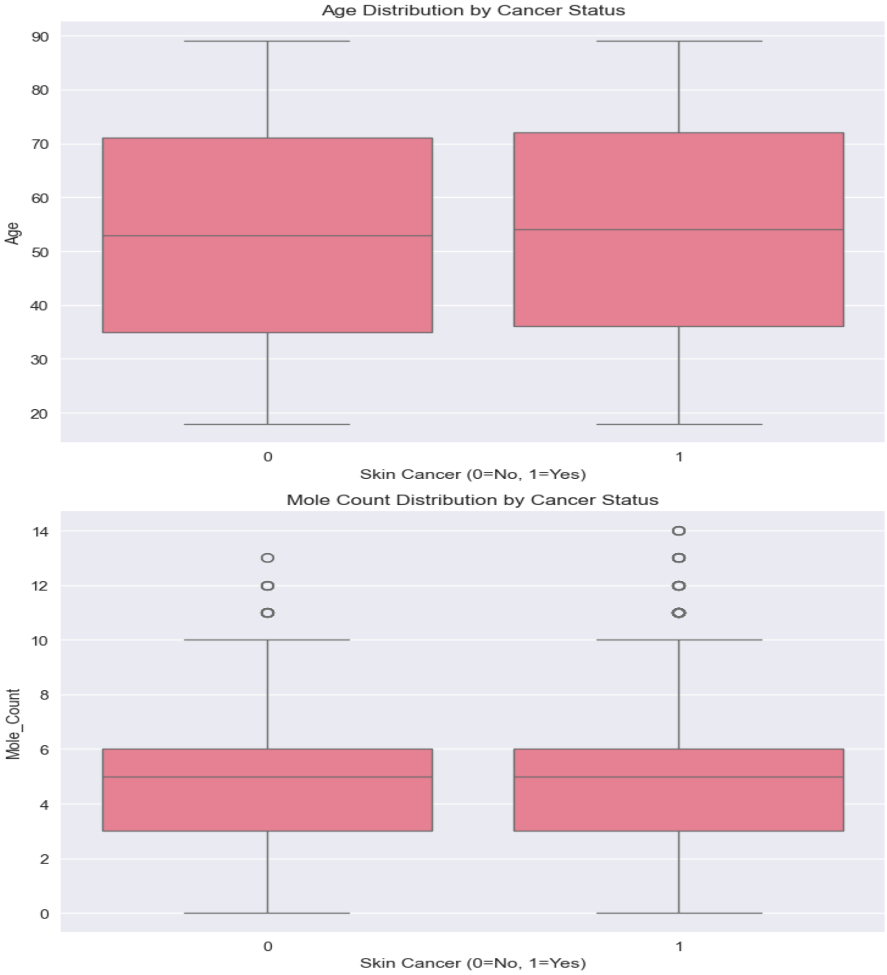
The target variable distribution pie chart provides crucial insights into the dataset's class balance, which is fundamental for understanding the scope of our prediction task. In medical datasets, particularly those dealing with cancer detection, class imbalance is common as the prevalence of positive cases (cancer) is typically lower than negative cases (benign).



Age and Mole Count Distributions by Cancer Status

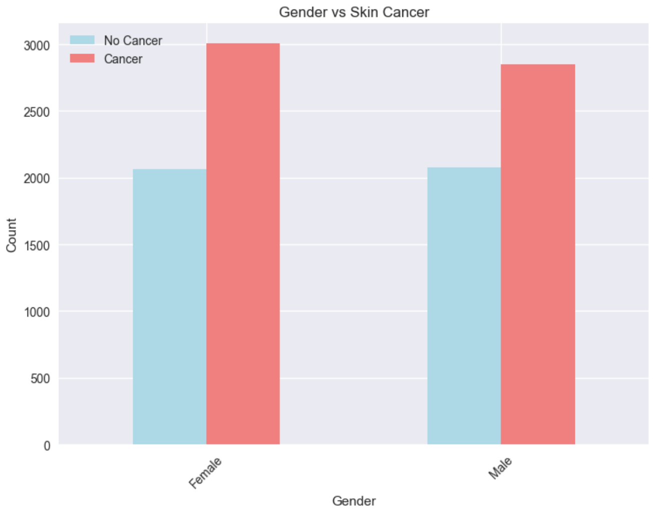
The distribution analysis of age and mole count across cancer status reveals critical demographic and clinical patterns. Age distribution comparisons can uncover whether certain age groups are more susceptible to malignant conditions, providing insights into risk factors and screening recommendations. Similarly, mole count distributions help understand the relationship between the number of moles and cancer likelihood. These visualizations often reveal whether there are clear separating thresholds or if the distributions overlap significantly, indicating the complexity of the classification task. Box plots or histograms segmented by cancer status can highlight differences in central tendencies, spread, and potential outliers that might influence model performance.

- Data Visualization



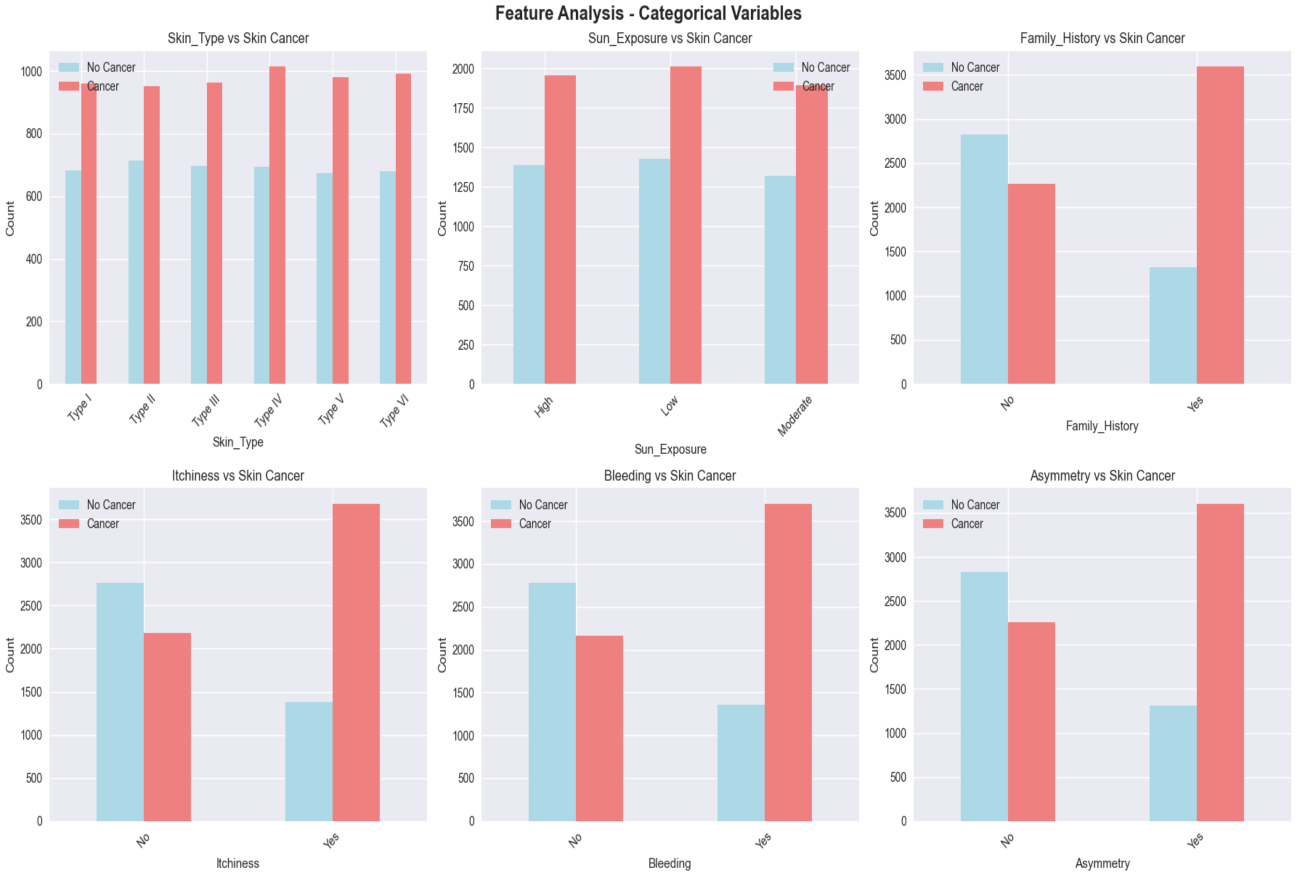
Gender Analysis with Cancer Correlation

Gender-based analysis provides valuable epidemiological insights into cancer patterns and helps identify potential gender-specific risk factors. This analysis examines whether there are significant differences in cancer incidence rates between males and females in the dataset. Such insights are crucial for understanding population-level health trends and can inform targeted screening programs or preventive measures. The visualization might reveal that certain demographic groups require more focused attention or different diagnostic approaches, which could influence feature engineering strategies and model interpretation.



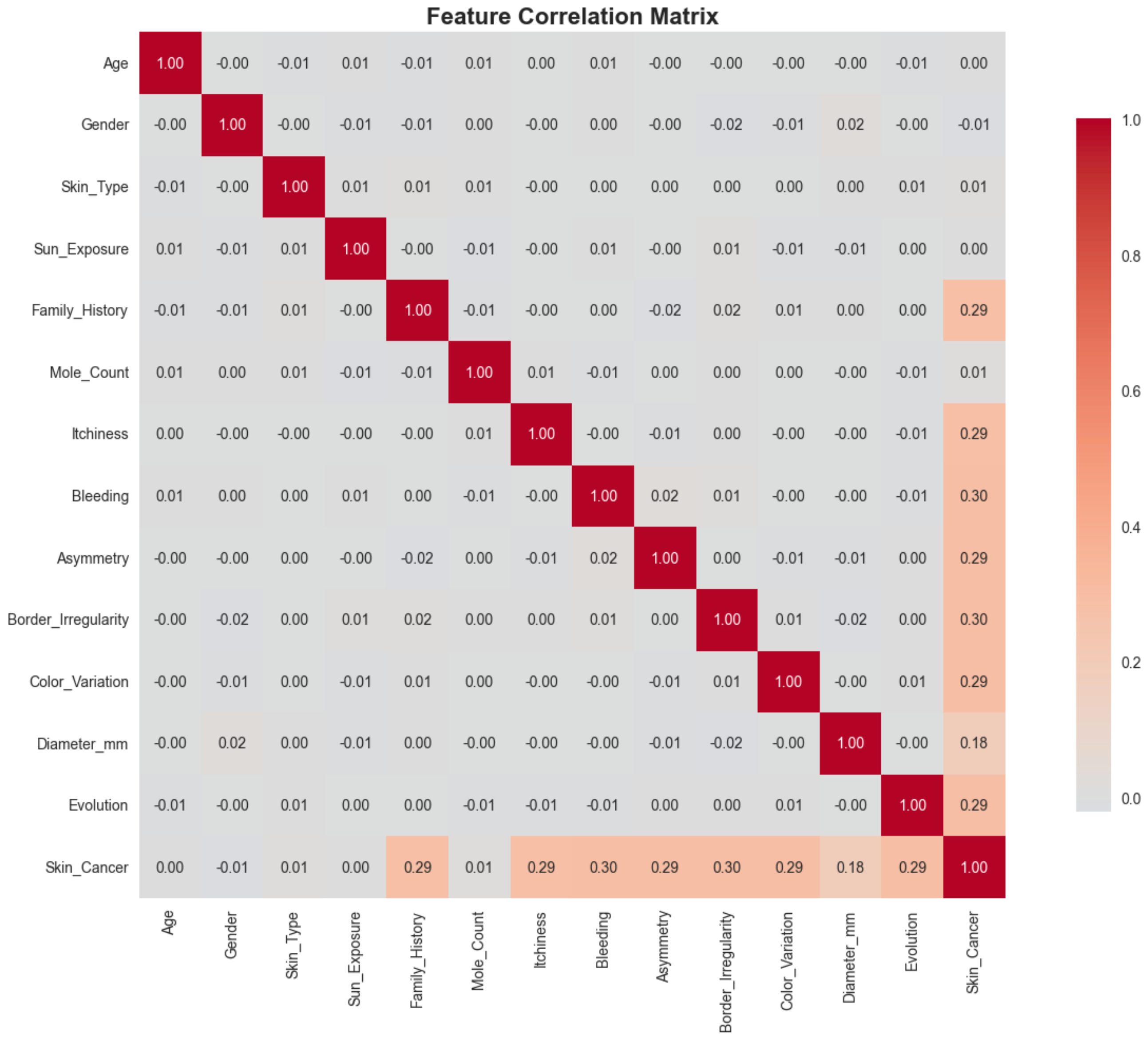
Categorical Feature Analysis with Bar Charts

Bar chart analysis of categorical features provides a comprehensive view of how different categorical variables relate to the target outcome. This includes analyzing features such as lesion location, family history, skin type, or other clinical indicators. These visualizations help identify which categories within each feature are most strongly associated with positive or negative outcomes. For instance, certain anatomical locations might show higher cancer rates, or specific skin types might be more prone to malignant conditions. This analysis is essential for understanding the clinical significance of each categorical variable and guides feature selection and encoding strategies.



Correlation Heatmap for All Features

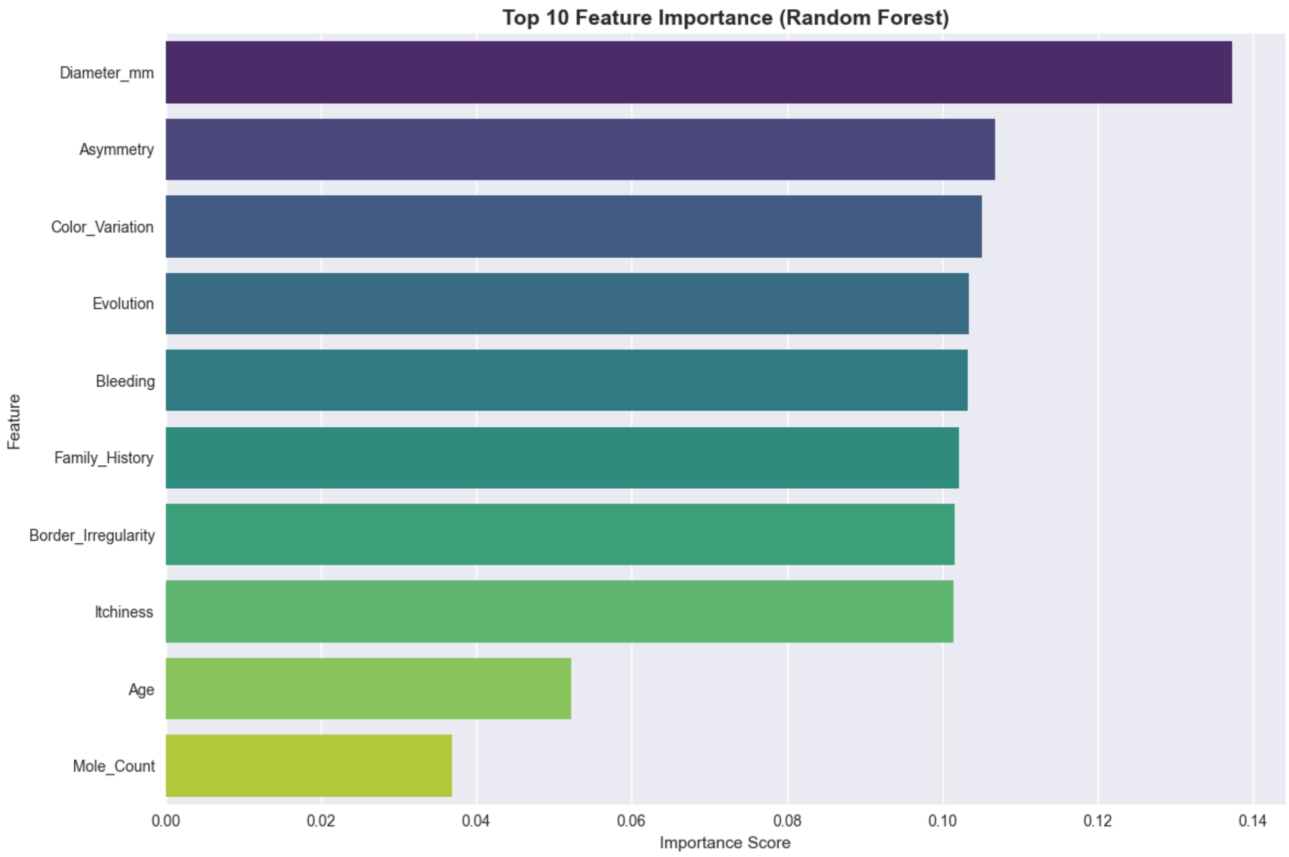
The correlation heatmap serves as a powerful tool for understanding the relationships between all features in the dataset. This visualization reveals multicollinearity issues, where highly correlated features might provide redundant information and potentially affect model stability. Strong correlations between features can indicate underlying biological or clinical relationships that might be leveraged for feature engineering or dimensionality reduction. The heatmap also helps identify features that are strongly correlated with the target variable, providing initial insights into which variables might be most predictive. Additionally, it can reveal unexpected relationships between features that might require further investigation or domain expert consultation.



	Age	Gender	Skin_Type	Sun_Exposure	Family_History	Mole_Count	Itchiness	Bleeding	Asymmetry	Border_Irregularity	Color_Variation	Diameter_mm	Evolution	Skin_Cancer
Age	1.00	-0.00	-0.01	0.01	-0.01	0.01	0.00	0.01	-0.00	-0.00	-0.00	-0.00	-0.01	0.00
Gender	-0.00	1.00	-0.00	-0.01	-0.01	0.00	-0.00	0.00	-0.00	-0.02	-0.01	0.02	-0.00	-0.01
Skin_Type	-0.01	-0.00	1.00	0.01	0.01	0.01	-0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01
Sun_Exposure	0.01	-0.01	0.01	1.00	-0.00	-0.01	-0.00	0.01	-0.00	0.01	-0.01	-0.01	0.00	0.00
Family_History	-0.01	-0.01	0.01	-0.00	1.00	-0.01	-0.00	0.00	-0.02	0.02	0.01	0.00	0.00	0.29
Mole_Count	0.01	0.00	0.01	-0.01	-0.01	1.00	0.01	-0.01	0.00	0.00	0.00	-0.00	-0.01	0.01
Itchiness	0.00	-0.00	-0.00	-0.00	-0.00	0.01	1.00	-0.00	-0.01	0.00	-0.00	-0.00	-0.01	0.29
Bleeding	0.01	0.00	0.00	0.01	0.00	-0.01	-0.00	1.00	0.02	0.01	-0.00	-0.00	-0.01	0.30
Asymmetry	-0.00	-0.00	0.00	-0.00	-0.02	0.00	-0.01	0.02	1.00	0.00	-0.01	-0.01	0.00	0.29
Border_Irregularity	-0.00	-0.02	0.00	0.01	0.02	0.00	0.00	0.01	0.00	1.00	0.01	-0.02	0.00	0.30
Color_Variation	-0.00	-0.01	0.00	-0.01	0.01	0.00	-0.00	-0.00	-0.01	0.01	1.00	-0.00	0.01	0.29
Diameter_mm	-0.00	0.02	0.00	-0.01	0.00	-0.00	-0.00	-0.00	-0.01	-0.02	-0.00	1.00	-0.00	0.18
Evolution	-0.01	-0.00	0.01	0.00	0.00	-0.01	-0.01	-0.01	0.00	0.00	0.01	-0.00	1.00	0.29
Skin_Cancer	0.00	-0.01	0.01	0.00	0.29	0.01	0.29	0.30	0.29	0.30	0.29	0.18	0.29	1.00

Top 10 Most Important Features Visualization

The visualization of the top 10 most important features creates a focused view of the key predictors in the dataset. This ranking typically includes both numerical and categorical features, providing a comprehensive picture of what drives the model's predictions. The importance scores help quantify the relative contribution of each feature, allowing for informed decisions about feature selection and model complexity. This visualization is particularly useful for communicating results to stakeholders and clinical professionals, as it highlights the most clinically relevant variables. The top features often align with known medical knowledge about cancer risk factors, providing validation of the model's learning process.



FEATURE SCALING

Feature scaling is the method to limit the range of variables so that they can be compared on common grounds. It is performed on continuous variables.

It can vary your results a lot while using certain algorithms and have a minimal or no effect in others.

Most of the time, your dataset will contain features highly varying in magnitudes, units and range. If left alone, these algorithms only take in the magnitude of features neglecting the units. The results would vary greatly between different units, like 1kg and 1000 gms. The features with high magnitudes will weigh in a lot more than the features with low magnitudes, for example, in calculations of distance between two different data points.

To handle this problem, we use feature scaling techniques, to bring all the features to the same level.

ONE HOT ENCODING

One hot encoding is a process by which categorical variables are converted into numerical form that could be provided to ML algorithms to do a better job in prediction. It is because, some ML libraries do not take categorical variables as input. Thus, we convert them into numerical variables. Presence of a level is represented by 1 and absence is represented by 0. For every level present, one dummy variable will be created.

MODELS EVALUATION

Accuracy

Accuracy measures the proportion of correct predictions out of total predictions made by the model. It's calculated as (True Positives + True Negatives) / Total Predictions. While intuitive and easy to understand, accuracy can be misleading with imbalanced datasets. For example, if 95% of samples are negative, a model predicting all negatives achieves 95% accuracy but fails to identify any positive cases. Accuracy works best when classes are balanced and misclassification costs are equal. It provides a general overview of model performance but doesn't reveal which classes are being predicted correctly or incorrectly.

Precision

Precision measures the proportion of true positive predictions among all positive predictions made by the model. Calculated as True Positives / (True Positives + False Positives), it answers "Of all instances predicted as positive, how many were actually positive?" High precision indicates low false positive rates, meaning the model rarely incorrectly labels negative cases as positive. This metric is crucial when false positives are costly, such as in spam detection or medical diagnosis. A model with high precision is conservative in making positive predictions, ensuring that when it predicts positive, it's likely correct. However, high precision might come at the cost of missing actual positive cases (low recall).

Recall (Sensitivity)

Recall measures the proportion of actual positive cases that were correctly identified by the model. Calculated as True Positives / (True Positives + False Negatives), it answers "Of all actual positive cases, how many did the model correctly identify?" High recall indicates the model successfully captures most positive instances, minimizing false negatives. This metric is critical when missing positive cases is costly, such as in disease screening or fraud detection. A model with high recall is aggressive in identifying positive cases but might also produce many false positives. The trade-off between precision and recall is fundamental in classification tasks, often requiring domain-specific prioritization.

F1-Score

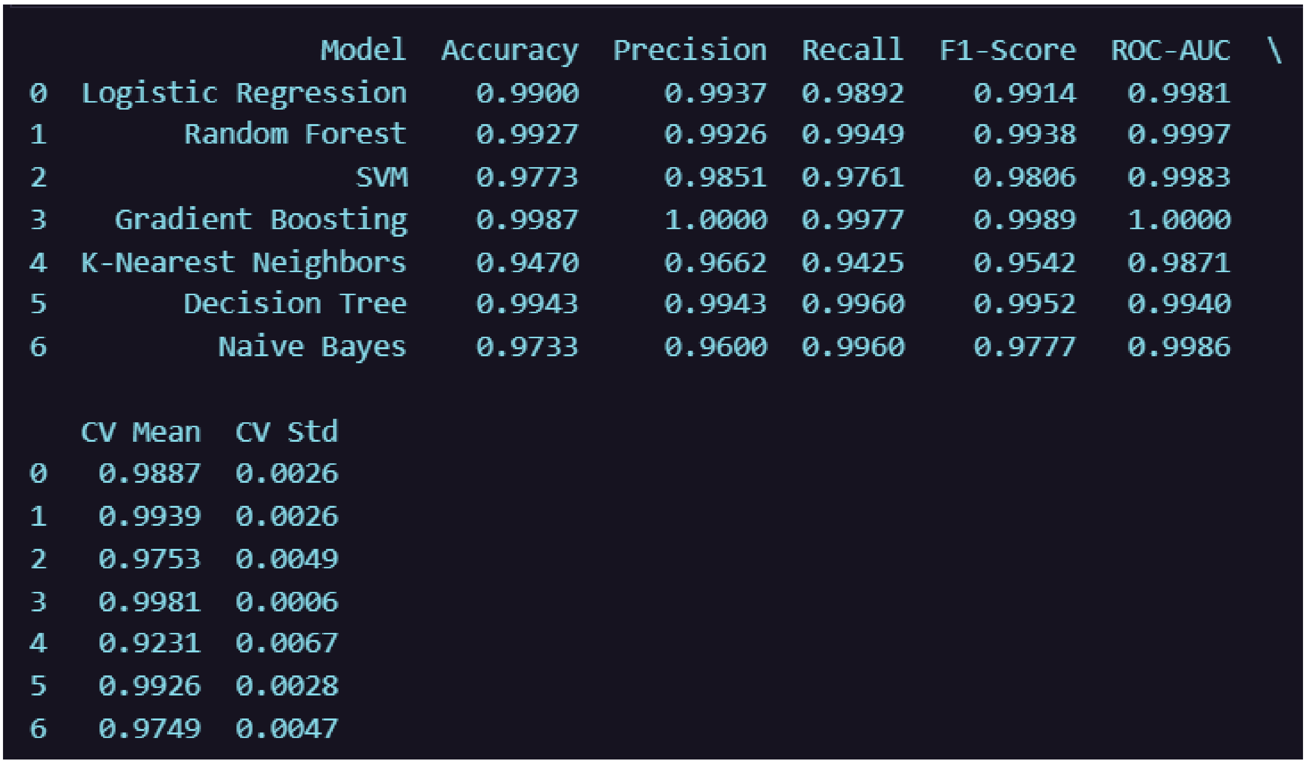
F1-Score is the harmonic mean of precision and recall, providing a single metric that balances both measures. Calculated as 2 × (Precision × Recall) / (Precision + Recall), it's particularly useful when you need to balance between false positives and false negatives. The harmonic mean ensures that both precision and recall must be reasonably high for a good F1-score; if either is very low, the F1-score will be low. This metric is valuable for imbalanced datasets where accuracy might be misleading. F1-Score ranges from 0 to 1, with 1 being perfect. It's widely used in machine learning competitions and real-world applications where both precision and recall matter equally.

ROC-AUC (Receiver Operating Characteristic - Area Under Curve)

ROC-AUC evaluates model performance across all classification thresholds by plotting True Positive Rate (Recall) against False Positive Rate (1-Specificity). The Area Under the Curve (AUC) summarizes this relationship into a single score between 0 and 1. An AUC of 0.5 indicates random performance, while 1.0 represents perfect classification. ROC-AUC is threshold-independent and provides insights into model discriminative ability. It's particularly useful for comparing models and understanding trade-offs between sensitivity and specificity. However, it can be optimistic with imbalanced datasets, as it doesn't directly account for class distribution. ROC-AUC is widely used in binary classification tasks and model selection processes.

Confusion Matrix

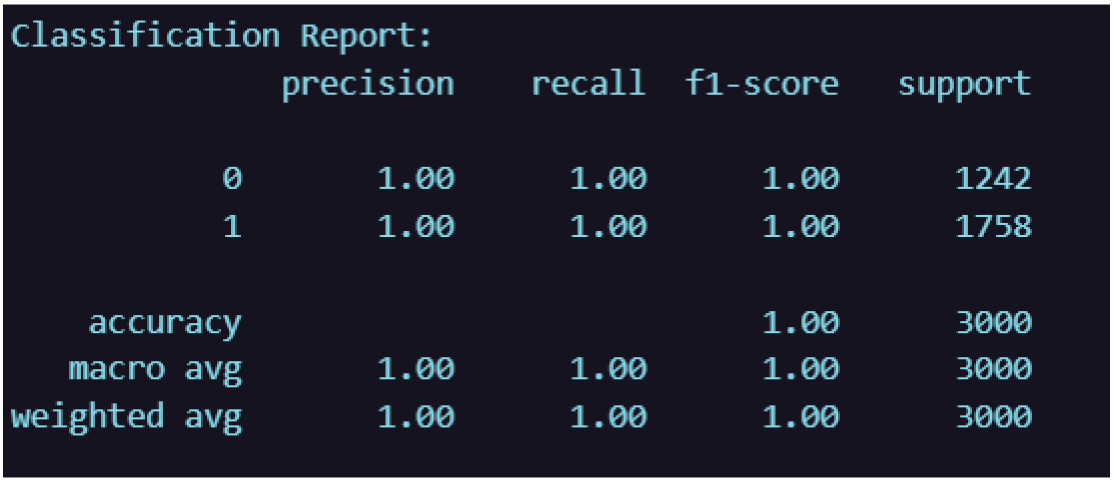
A confusion matrix is a table that visualizes the performance of a classification model by showing actual vs. predicted classifications. For binary classification, it's a 2×2 matrix showing True Positives, False Positives, True Negatives, and False Negatives. This matrix provides the foundation for calculating all other metrics and offers detailed insights into model errors. It reveals which classes are being confused with each other, helping identify specific areas for improvement. The confusion matrix is essential for understanding model behavior, especially in multi-class problems where it shows the complete error pattern. It's the starting point for most classification evaluation analyses and helps in making informed decisions about model adjustments.



```
                 Model  Accuracy  Precision  Recall  F1-Score  ROC-AUC  \
0  Logistic Regression    0.9900     0.9937  0.9892    0.9914   0.9981   
1        Random Forest    0.9927     0.9926  0.9949    0.9938   0.9997   
2                  SVM    0.9773     0.9851  0.9761    0.9806   0.9983   
3    Gradient Boosting    0.9987     1.0000  0.9977    0.9989   1.0000   
4  K-Nearest Neighbors    0.9470     0.9662  0.9425    0.9542   0.9871   
5        Decision Tree    0.9943     0.9943  0.9960    0.9952   0.9940   
6          Naive Bayes    0.9733     0.9600  0.9960    0.9777   0.9986   

   CV Mean  CV Std  
0   0.9887  0.0026  
1   0.9939  0.0026  
2   0.9753  0.0049  
3   0.9981  0.0006  
4   0.9231  0.0067  
5   0.9926  0.0028  
6   0.9749  0.0047  
```

Classification Report



```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1242
           1       1.00      1.00      1.00      1758

    accuracy                           1.00      3000
   macro avg       1.00      1.00      1.00      3000
weighted avg       1.00      1.00      1.00      3000
```

Model Deployment Components

Model Deployment Components: Short Explanation

1. skin_cancer_model.pkl - The Trained Model

This file contains the actual machine learning algorithm with all its learned parameters and decision-making rules. It's the "brain" that makes predictions by applying mathematical relationships discovered during training. Without this file, no predictions can be made.

Contains:

- Trained algorithm (Random Forest trees, SVM parameters, etc.)
- Decision boundaries and feature relationships
- All weights and parameters learned from training data

Purpose: Makes actual predictions on new patient data using learned patterns.

2. scaler.pkl - Feature Scaling Object

This file stores the scaling parameters (mean, standard deviation) used to normalize numerical features during training. It ensures new data is scaled exactly the same way as training data.

Contains:

- Mean and standard deviation for each numerical feature
- Scaling transformation parameters from training data

Purpose: Prevents prediction errors caused by different feature scales (e.g., age: 20-80 vs diameter: 1-10mm).

Critical: Without consistent scaling, the model receives incorrectly formatted data and makes wrong predictions.

3. label_encoders.pkl - Categorical Variable Encoders

This file contains the mappings that convert categorical text values (like "Male/Female") into numbers that machine learning models can understand. It ensures consistent encoding between training and deployment.

Contains:

- Dictionary of encoder objects for each categorical feature
- Text-to-number mappings (e.g., Male=0, Female=1)

Purpose: Converts text categories to numbers using the exact same mapping as training.

Critical: If "Male" was encoded as 0 during training but as 1 during deployment, predictions would be wrong.

DEPLOYMENT OF MODEL

The skin cancer detection model has been successfully deployed as an interactive web application using Streamlit, a Python-based framework for building and deploying machine learning applications. This deployment provides a user-friendly interface that allows healthcare professionals and users to input patient data and receive real-time cancer risk predictions.

Why Streamlit for Model Deployment

Streamlit was chosen as the deployment platform due to its simplicity, rapid development capabilities, and seamless integration with Python-based machine learning models. Unlike traditional web development that requires extensive knowledge of HTML, CSS, and JavaScript, Streamlit allows data scientists to create interactive web applications using pure Python code. This approach significantly reduces development time while maintaining professional functionality.

The framework provides native support for common machine learning libraries including scikit-learn, pandas, and numpy, making it ideal for deploying our trained models without additional complexity. Streamlit's automatic UI generation and real-time updates create a responsive user experience that updates predictions instantly as users modify input parameters.

Application Architecture

The deployed application follows a clean, modular architecture that separates concerns between data processing, model inference, and user interface components:

Frontend Interface:

- Interactive input forms for patient data entry
- Real-time prediction display with confidence scores
- Visual feedback through progress bars and color-coded results
- Responsive design that works across different devices

Backend Processing:

- Model loading and initialization using joblib
- Data preprocessing pipeline with saved scalers and encoders
- Prediction generation with probability estimates
- Error handling and input validation

Data Flow:

1. User inputs patient information through Streamlit widgets
2. Application validates and preprocesses the input data
3. Trained model generates predictions and confidence scores
4. Results are displayed through intuitive visualizations

Key Features Implemented

Interactive Data Input

The application provides intuitive input mechanisms for all model features:

- Numerical inputs: Slider widgets for age, mole count, and lesion measurements
- Categorical inputs: Dropdown menus for gender, skin type, and family history
- Boolean inputs: Checkbox widgets for binary features
- Real-time validation: Immediate feedback on input ranges and format

Prediction Display

Results are presented through multiple visualization methods:

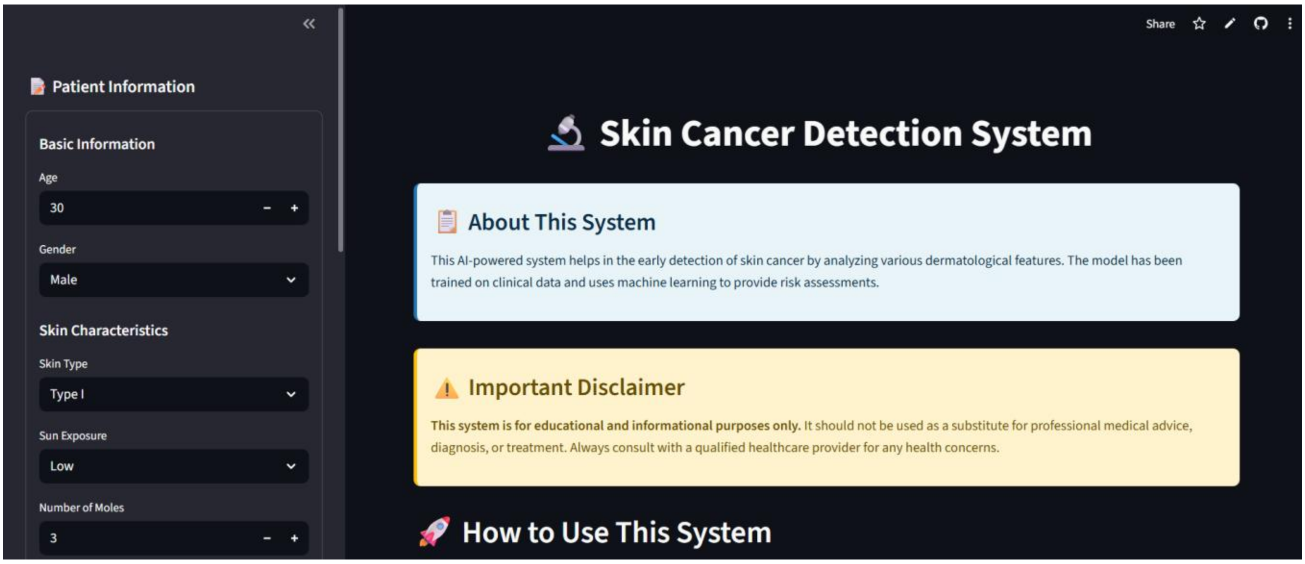
- Risk classification: Clear indication of malignant vs. benign prediction
- Confidence scores: Probability percentages for each class
- Visual indicators: Color-coded results (red for high risk, green for low risk)
- Explanatory text: Contextual information about the prediction

Model Performance Metrics

The application displays key performance indicators:

- Model accuracy and evaluation metrics
- Feature importance rankings
- Prediction confidence levels
- Model version and last update information

Some snapshot from the deployed system



Patient Information

Basic Information

Age: 30

Gender: Male

Skin Characteristics

Skin Type: Type I

Sun Exposure: Low

Number of Moles: 3

Skin Cancer Detection System

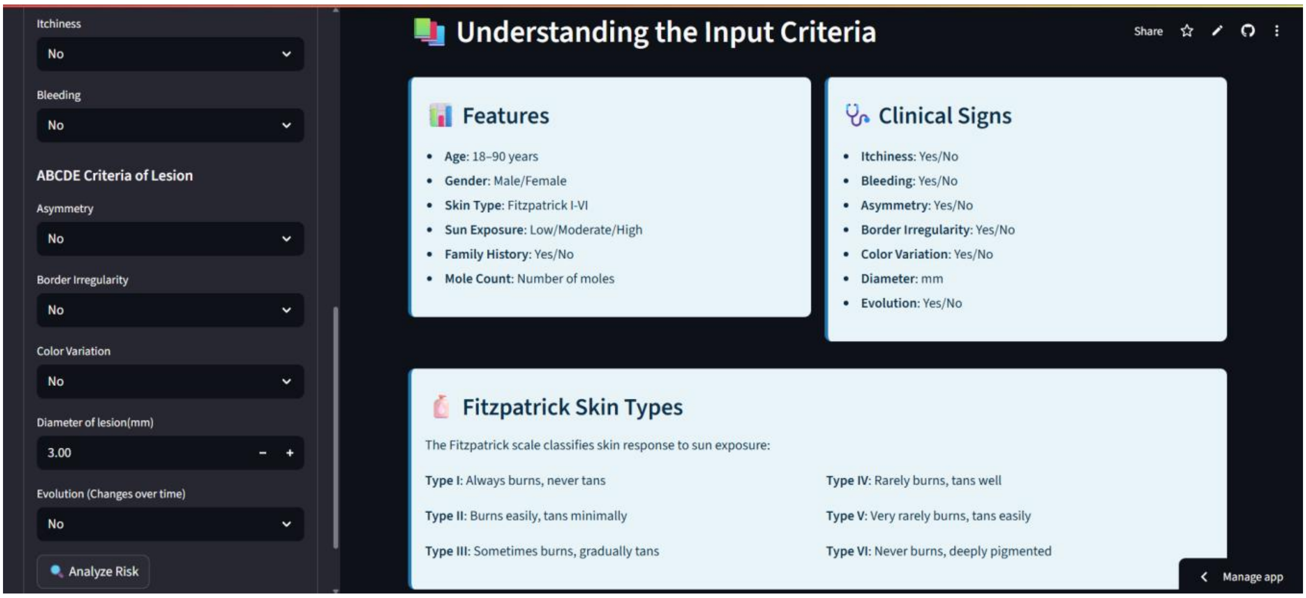
About This System

This AI-powered system helps in the early detection of skin cancer by analyzing various dermatological features. The model has been trained on clinical data and uses machine learning to provide risk assessments.

⚠ Important Disclaimer

This system is for educational and informational purposes only. It should not be used as a substitute for professional medical advice, diagnosis, or treatment. Always consult with a qualified healthcare provider for any health concerns.

How to Use This System



Itchiness

No

Bleeding

No

ABCDE Criteria of Lesion

Asymmetry

No

Border Irregularity

No

Color Variation

No

Diameter of lesion(mm)

3.00

Evolution (Changes over time)

No

Understanding the Input Criteria

Features

- **Age**: 18–90 years
- **Gender**: Male/Female
- **Skin Type**: Fitzpatrick I-VI
- **Sun Exposure**: Low/Moderate/High
- **Family History**: Yes/No
- **Mole Count**: Number of moles

Clinical Signs

- **Itchiness**: Yes/No
- **Bleeding**: Yes/No
- **Asymmetry**: Yes/No
- **Border Irregularity**: Yes/No
- **Color Variation**: Yes/No
- **Diameter**: mm
- **Evolution**: Yes/No

Fitzpatrick Skin Types

The Fitzpatrick scale classifies skin response to sun exposure:

Type I: Always burns, never tans

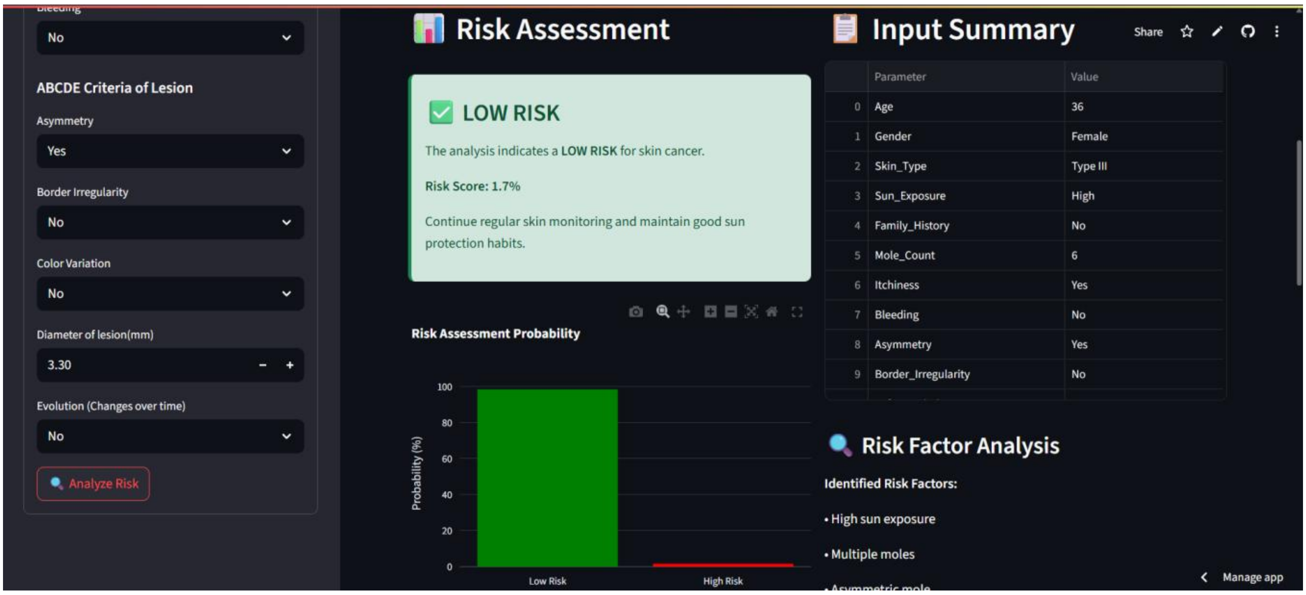
Type II: Burns easily, tans minimally

Type III: Sometimes burns, gradually tans

Type IV: Rarely burns, tans well

Type V: Very rarely burns, tans easily

Type VI: Never burns, deeply pigmented



No

ABCDE Criteria of Lesion

Asymmetry

Yes

Border Irregularity

No

Color Variation

No

Diameter of lesion(mm)

3.30

Evolution (Changes over time)

No

Risk Assessment

✅ LOW RISK

The analysis indicates a **LOW RISK** for skin cancer.

Risk Score: 1.7%

Continue regular skin monitoring and maintain good sun protection habits.

Risk Assessment Probability

Input Summary

	Parameter	Value
0	Age	36
1	Gender	Female
2	Skin_Type	Type III
3	Sun_Exposure	High
4	Family_History	No
5	Mole_Count	6
6	Itchiness	Yes
7	Bleeding	No
8	Asymmetry	Yes
9	Border_Irregularity	No

Risk Factor Analysis

Identified Risk Factors:

- High sun exposure
- Multiple moles
- Asymmetric mole

CONCLUSION

The skin cancer detection system represents a successful integration of machine learning techniques with healthcare applications, demonstrating both technical proficiency and practical utility. While the system shows promise for supporting clinical decision-making, its ultimate value will be determined through clinical validation and real-world implementation.

The project highlights the potential of AI to enhance healthcare delivery while emphasizing the importance of careful validation, ethical considerations, and integration with existing clinical workflows. As healthcare increasingly adopts AI-powered tools, projects like this provide valuable insights into the development of effective, responsible, and clinically meaningful applications.

The successful completion of this project demonstrates not only technical competence in machine learning and software development but also an understanding of the complexities involved in translating academic research into practical healthcare solutions. This foundation provides a strong basis for future work in medical AI and healthcare technology development.

Through this comprehensive approach to skin cancer detection, the project contributes to the growing field of healthcare AI while providing a practical tool that could potentially improve patient outcomes through earlier detection and more accessible screening capabilities.

REFERENCES

1. **Siegel, R. L., Miller, K. D., Wagle, N. S., & Jemal, A. (2023).** Cancer statistics, 2023. *CA: A Cancer Journal for Clinicians*, 73(1), 233-254. https://doi.org/10.3322/caac.21763

2. **Apalla, Z., Nashan, D., Weller, R. B., & Castellsagué, X. (2017).** Skin cancer: epidemiology, disease burden, pathophysiology, diagnosis, and therapeutic approaches. *Dermatology and Therapy*, 7(1), 5-19. https://doi.org/10.1007/s13555-016-0165-y

3. **Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017).** Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639), 115-118. https://doi.org/10.1038/nature21056

4. **Brinker, T. J., Hekler, A., Utikal, J. S., Grabe, N., Schadendorf, D., Klode, J., ... & von Kalle, C. (2018).** Skin cancer classification using convolutional neural networks: systematic review. *Journal of Medical Internet Research*, 20(10), e11936. https://doi.org/10.2196/11936

5. **Haenssle, H. A., Fink, C., Schneiderbauer, R., Toberer, F., Buhl, T., Blum, A., ... & Uhlmann, L. (2018).** Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists. *Annals of Oncology*, 29(8), 1836-1842. https://doi.org/10.1093/annonc/mdy166

6. **Breiman, L. (2001).** Random forests. *Machine Learning*, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324

7. **Cortes, C., & Vapnik, V. (1995).** Support-vector networks. *Machine Learning*, 20(3), 273-297. https://doi.org/10.1007/BF00994018

8. **Chen, T., & Guestrin, C. (2016).** XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794). https://doi.org/10.1145/2939672.2939785

9. **Friedman, J. H. (2001).** Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 29(5), 1189-1232. https://doi.org/10.1214/aos/1013203451

10. **Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013).** Applied logistic regression (Vol. 398). John Wiley & Sons.

11. **Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011).** Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

12. **Fawcett, T. (2006).** An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), 861-874. https://doi.org/10.1016/j.patrec.2005.10.010

13. **Sokolova, M., & Lapalme, G. (2009).** A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4), 427-437. https://doi.org/10.1016/j.ipm.2009.03.002

14. **Lones, M. A. (2021).** How to avoid machine learning pitfalls: a guide for academic researchers. *arXiv preprint arXiv:2108.02497*. https://doi.org/10.48550/arXiv.2108.02497

15. **Topol, E. J. (2019).** High-performance medicine: the convergence of human and artificial intelligence. *Nature Medicine*, 25(1), 44-56. https://doi.org/10.1038/s41591-018-0300-7

16. **Rajpara, S. M., Botello, A. P., Townend, J., & Ormerod, A. D. (2009).** Systematic review of dermoscopy and digital dermoscopy/artificial intelligence for the diagnosis of melanoma. *British Journal of Dermatology*, 161(3), 591-604. https://doi.org/10.1111/j.1365-2133.2009.09093.x

17. **Marchetti, M. A., Codella, N. C., Dusza, S. W., Gutman, D. A., Helba, B., Kalloo, A., ... & Halpern, A. C. (2018).** Results of the 2016 International Skin Imaging Collaboration International Symposium on Biomedical Imaging challenge: Comparison of the accuracy of computer algorithms to dermatologists for the diagnosis of melanoma from dermoscopic images. *Journal of the American Academy of Dermatology*, 78(2), 270-277. https://doi.org/10.1016/j.jaad.2017.08.016

18. **Celebi, M. E., Kingravi, H. A., Uddin, B., Iyatomi, H., Aslandogan, Y. A., Stoecker, W. V., & Moss, R. H. (2007).** A methodological approach to the classification of dermoscopy images. *Computerized Medical Imaging and Graphics*, 31(6), 362-373. https://doi.org/10.1016/j.compmedimag.2007.01.003