DRY EYE DISEASE PREDICTION USING MACHINE LEARNING

MALAK ATEF BADOUR

221000906

Abstract

Dry eye disease is a multifactorial disorder affecting millions globally, yet its diagnosis remains highly dependent on subjective clinical assessment. The objective of this project is to develop a machine learning model capable of accurately classifying individuals as having or not having dry eye disease based on measurable ocular parameters. Using a publicly available dataset, we conducted a full end-to-end pipeline starting from data cleaning, exploratory data analysis, and feature engineering, followed by the implementation and tuning of several classification models including Random Forest, XGBoost, K-Nearest Neighbors, Support Vector Machine, Decision Tree, AdaBoost, and a Voting Classifier. Through a rigorous comparison of model performance metrics such as accuracy, precision, recall, and ROC AUC score, this project identifies the most effective model for clinical decision support. This study demonstrates how data-driven approaches can supplement diagnostic processes and pave the way for early, accurate, and scalable detection of dry eye disease.

Introduction

Dry eye disease (DED) is one of the most prevalent ocular surface conditions and is associated with a wide range of symptoms that negatively impact vision and quality of life. Despite its significance, the diagnostic process for DED is often subjective and inconsistent, relying heavily on individual clinician expertise and patient-reported outcomes. As such, there is a growing need for objective, data-driven tools that can support clinical decision-making in ophthalmology.

The primary question driving this study is: Can machine learning models accurately classify dry eye disease based on ocular clinical features? This project aims to address this question by building and evaluating various predictive models using a real-world dataset that includes both continuous and categorical ocular parameters.

To ensure the reliability and generalizability of the analysis, extensive preprocessing was applied to handle missing values, encode categorical variables, and normalize continuous features. Moreover, multiple classification algorithms were implemented and compared using well-established evaluation metrics. Feature engineering and selection techniques were also employed to improve model performance and interpretability.

By systematically addressing the core research question, this project not only applies theoretical machine learning concepts to a practical biomedical problem but also highlights the potential of AI-driven diagnostics in enhancing healthcare outcomes.

Methodology

Feature Engineering

A new feature, ScreenTime to Sleep Ratio, was created to reflect the balance between screen exposure and rest — two factors known to influence dry eye symptoms. This ratio combines average screen time with sleep duration to produce a more informative measure of lifestyle habits. By capturing this relationship, the feature aimed to improve model performance and interpretability.

Preprocessing

Several key steps were taken to clean and prepare the data before modeling. First, all binary categorical columns were encoded by converting 'Y' and 'N' values into 1 and 0, ensuring compatibility with machine learning algorithms. Gender was encoded as 1 for male and 0 for female, while the Sleep disorder column was converted using categorical codes.

The Blood pressure feature, originally in a combined string format, was split into two numeric columns: Systolic_BP and Diastolic_BP, and the original string column was removed. Next, numeric features were standardized using StandardScaler to normalize their ranges and improve model convergence. This included variables like age, screen time, sleep duration, blood pressure, and the previously engineered ScreenTime_SleepRatio.

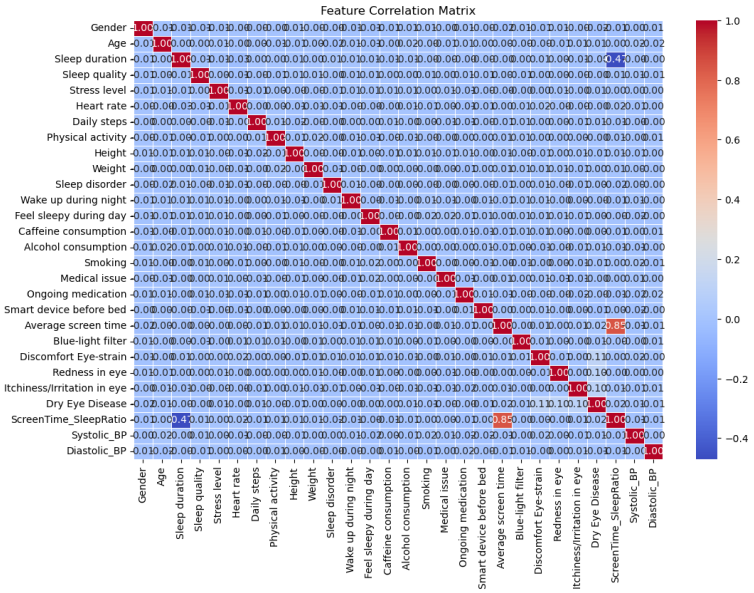


Figure: Feature Correlation Matrix

this heatmap illustrates the Pearson correlation coefficients among all features. The majority of values are near zero, indicating weak linear relationships and low multicollinearity. This suggests that the features are largely independent, which is beneficial for many machine learning models.

To improve data quality, outliers were removed based on Z-score thresholding, resulting in the exclusion of entries with extreme values across any feature. This helped reduce noise and potential distortion in model training.

The data was then split into training and testing subsets using stratified sampling to preserve class balance. SMOTE (Synthetic Minority Over-sampling Technique) was applied to the training set to address class imbalance, ensuring both dry eye and non-dry eye cases were equally represented.

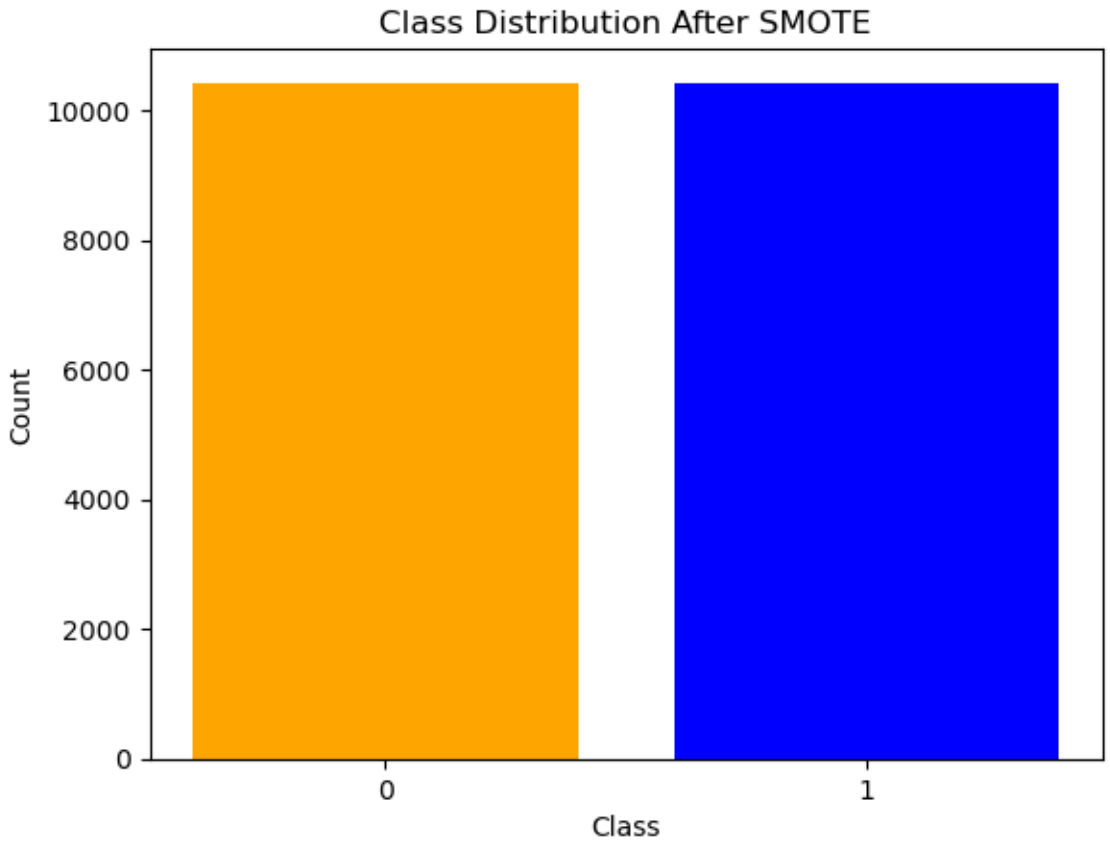


Figure : *Class Distribution After SMOTE.*
This bar chart shows that the number of instances in both classes (0 = No Dry Eye Disease, 1 = Dry Eye Disease) was equalized post-resampling using SMOTE, ensuring fair model learning.

Lastly, feature selection was performed using the ANOVA F-test to retain the most predictive variables. Selected features included indicators like alcohol consumption, screen time, and eye discomfort symptoms, all of which have known relevance to dry eye disease.

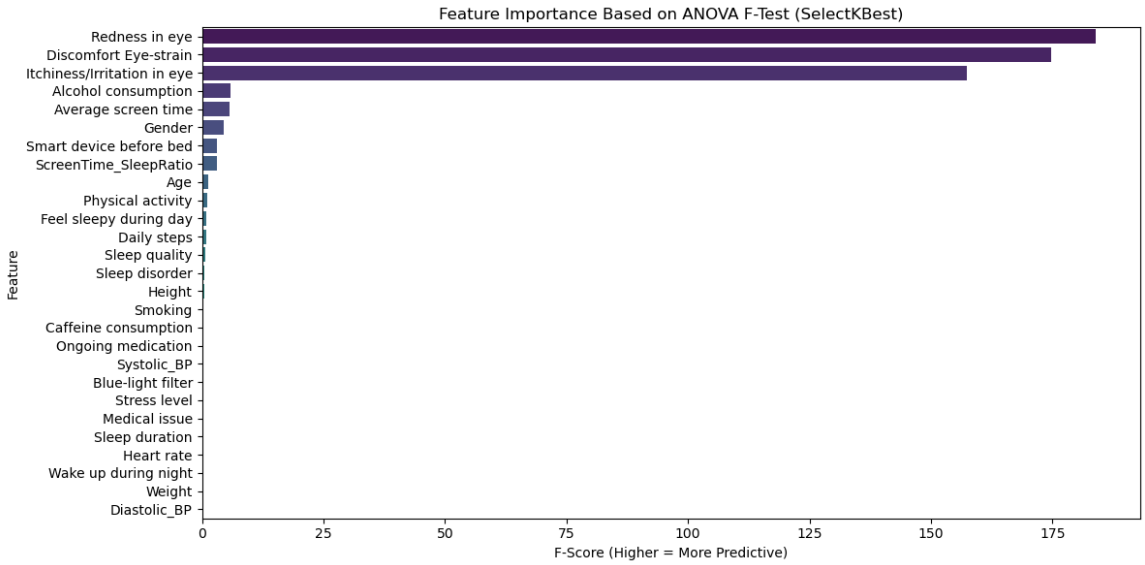


Figure : *Feature Importance Based on ANOVA F-Test (SelectKBest).*
This bar chart highlights that "Redness in eye," "Discomfort Eye-strain," and "Itchiness/Irritation in eye" were the most significant predictors of Dry Eye Disease, with the highest F-scores indicating stronger predictive power.

Linearity Check

Before selecting the final models, we examined the linearity assumption using a simple linear regression between one numerical feature (age) and the binary target (dry eye diagnosis). The fitted regression line and residual plots indicated that the relationship was not linear. Additionally, the Q-Q plot and residual histogram showed deviations from normality, while the Durbin-Watson statistic (~2.0) confirmed no autocorrelation in residuals. These findings justified our use of non-linear models such as Random Forest, XGBoost, and others, which are more appropriate for handling complex and non-linear patterns in the data.

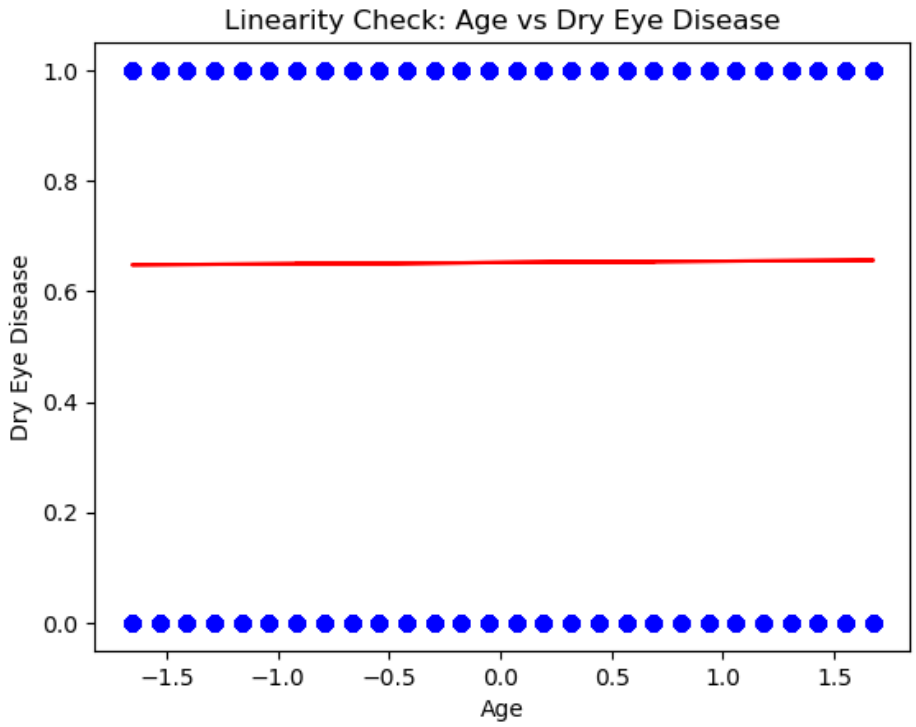


Figure 1: *Linearity Check: Age vs Dry Eye Disease.*
A simple regression plot showing no linear trend between age and dry eye diagnosis.

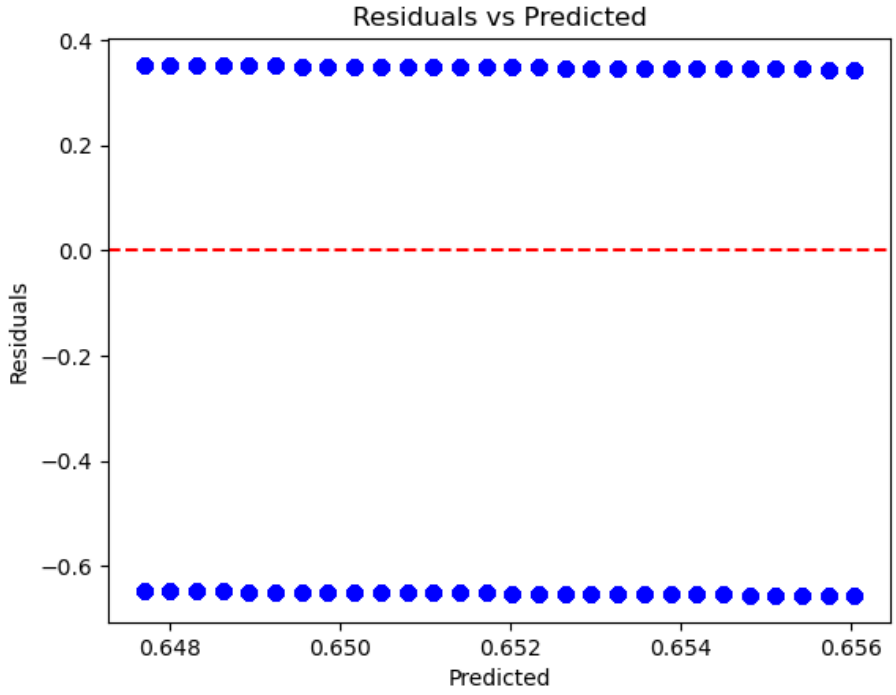


Figure 2: *Residuals vs Predicted Plot.*
Residuals appear scattered with no discernible pattern, further indicating non-linearity.

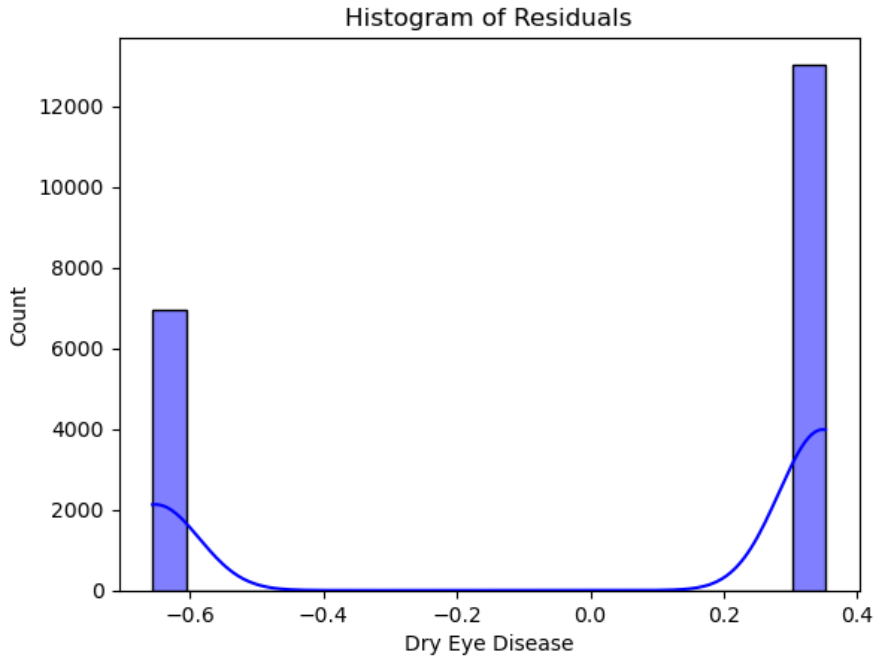


Figure 3: *Histogram of Residuals.*
The residuals are not normally distributed, with peaks at the extremes, violating linear model assumptions.

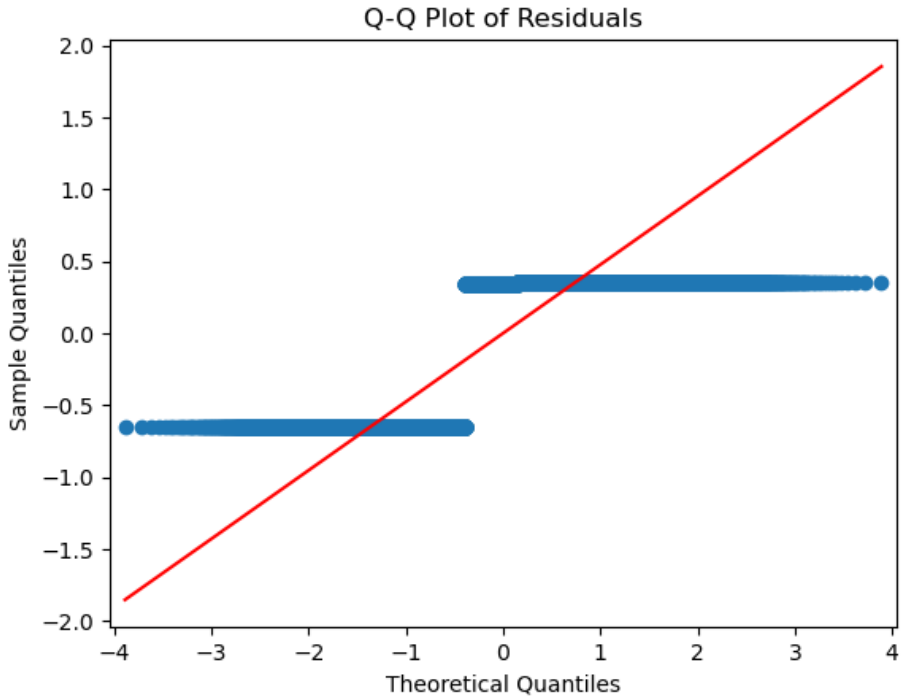


Figure 4: *Q-Q Plot of Residuals.*
Deviations from the diagonal line indicate residuals are not normally distributed, supporting the use of non-linear models.

Modeling

A comprehensive set of classification algorithms was applied to identify the best-performing model for predicting dry eye disease. The goal was to compare a wide range of techniques, from simple linear models to complex ensemble approaches, in order to capture both linear and non-linear patterns in the data.

The models implemented include:

- **Logistic Regression**: A baseline linear model commonly used for binary classification problems.
- **K-Nearest Neighbors (KNN)**: A distance-based model that classifies instances based on the majority vote of nearby neighbors.
- **Naive Bayes**: A probabilistic classifier based on Bayes' theorem, assuming independence among features.
- **Support Vector Machine (SVM)**: A classifier that seeks the optimal separating hyperplane for binary classification, effective in high-dimensional spaces.
- **Decision Tree**: A rule-based model that splits data using feature thresholds, offering interpretability.
- **Random Forest**: An ensemble of decision trees that improves accuracy and reduces overfitting by averaging multiple trees.
- **AdaBoost**: An adaptive boosting model that iteratively focuses on misclassified instances to improve prediction.
- **XGBoost**: An optimized gradient boosting technique known for its high performance and efficiency.
- **Gradient Boosting Classifier**: Another boosting method that builds an additive model in a forward stage-wise fashion.
- **Bagging Classifier**: A model that combines predictions of multiple base learners (like decision trees) trained on different subsets of the data.
- **Voting Classifier**: An ensemble meta-model that aggregates predictions from multiple models (hard or soft voting).
- **Stacking Classifier**: An advanced ensemble model that learns how to best combine the predictions of several base models using a meta-learner.

Each model was trained on the preprocessed, SMOTE-balanced training set and evaluated on the test set using consistent metrics. **Hyperparameter tuning** was applied where

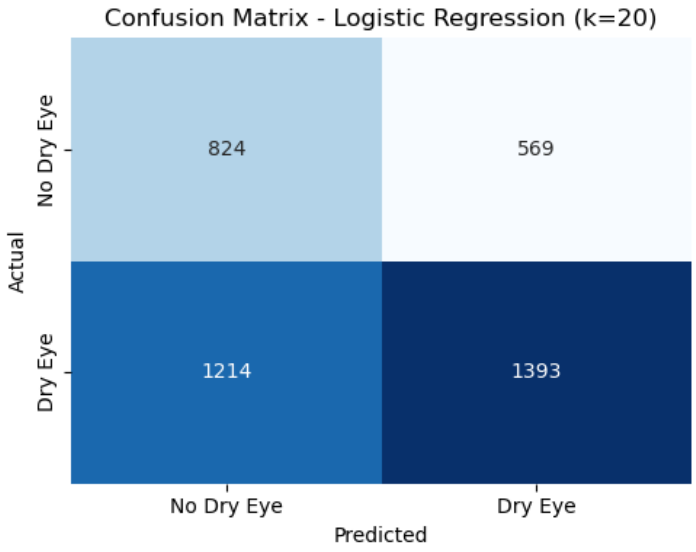
appropriate, using techniques like GridSearchCV to improve performance through systematic parameter exploration.

This diverse modeling strategy ensured a robust comparison and allowed us to select the most effective model for deployment in a real-world dry eye prediction setting.

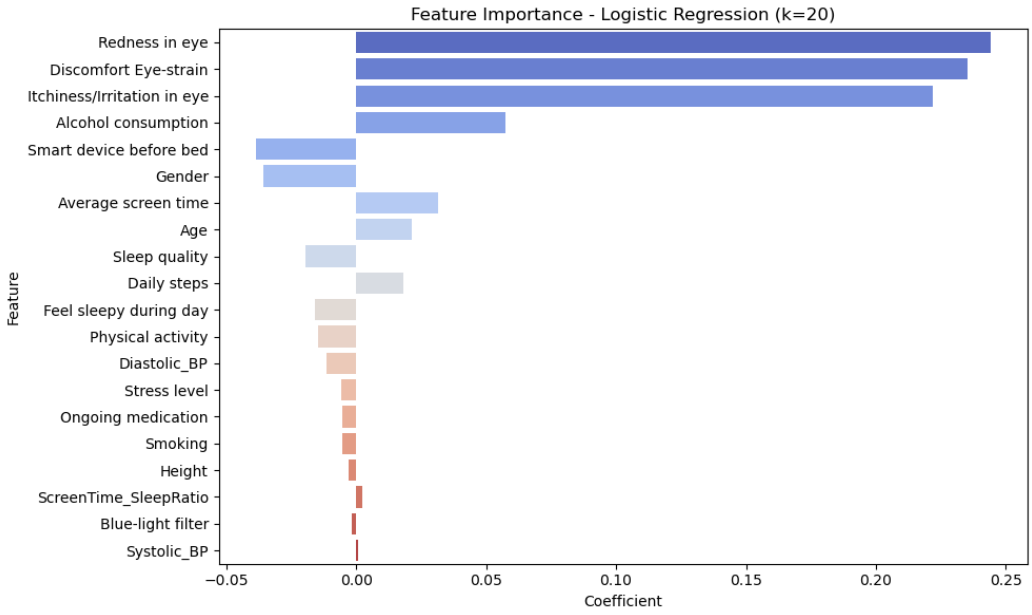
Results and Discussion

Logistic Regression

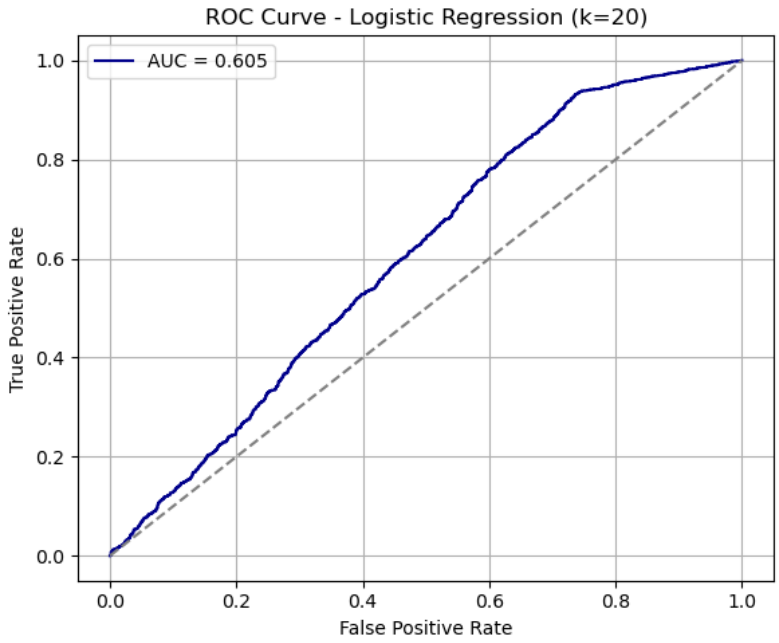
The Logistic Regression model was trained using the top 20 features selected by ANOVA F-test and optimized via grid search. The best model parameters were C = 0.1 and solver = 'liblinear'. On the test set, the model achieved an accuracy of 55.4%, an F1 score of 60.9%, and an AUC score of 0.605. While these scores suggest modest predictive power, Logistic Regression served as a valuable baseline. The confusion matrix showed that the model correctly identified 1,393 dry eye cases (true positives) but misclassified 1,214 actual dry eye cases as non-dry eye (false negatives), revealing a significant limitation. Additionally, it predicted 569 false positives, labeling non-dry eye patients incorrectly. These errors are particularly concerning in a medical context, where undiagnosed conditions may delay treatment. The ROC curve confirmed the model's limited discriminative ability, with only slight improvement over random guessing. In terms of feature importance, the model relied most heavily on symptoms such as redness, eye strain, and itchiness — clinically relevant indicators of dry eye disease. Despite its interpretability, Logistic Regression underperformed relative to more advanced models, particularly in identifying positive cases accurately.



The confusion matrix shows that 1,214 actual dry eye cases were missed (false negatives), making this model unreliable for accurate detection in clinical settings.



Top features influencing Logistic Regression included redness, discomfort, and itchiness — symptoms strongly associated with dry eye.

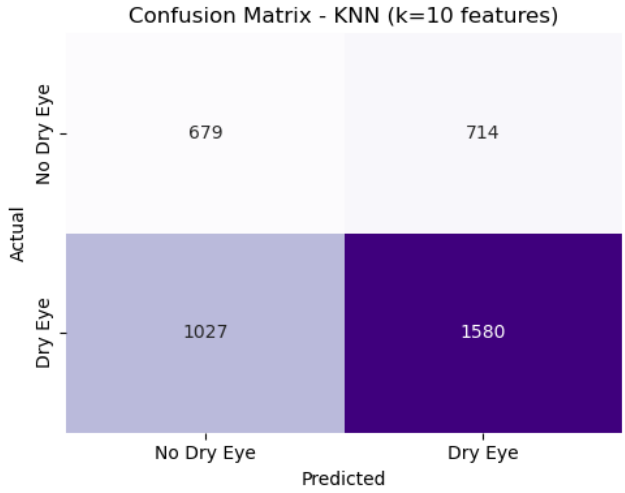


The ROC curve illustrates the model's ability to distinguish between classes. With an AUC of 0.605.

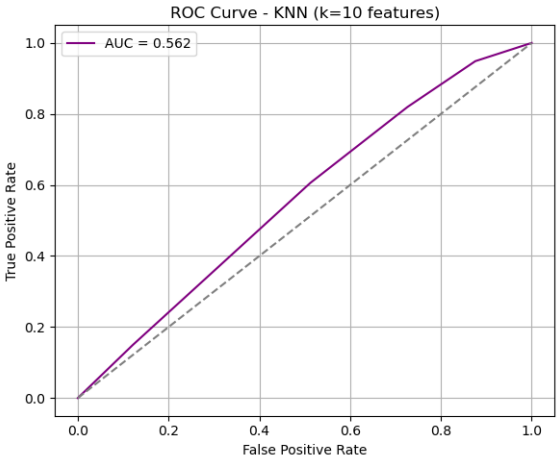
K-Nearest Neighbors (KNN) Results

The K-Nearest Neighbors model was trained using the top 10 features selected by ANOVA F-test, scaled using standardization, and trained on a balanced dataset produced via SMOTE. With k=5 neighbors, the model achieved an accuracy of 56.5%, an F1 score of 64.5%, and an

AUC of 0.562. These results reflect only a slight improvement over Logistic Regression but still indicate limited ability to separate classes effectively. The confusion matrix reveals that the model correctly identified 1,580 dry eye cases, but misclassified 1,027 actual dry eye patients as negative (false negatives), and also generated 714 false positives. This shows the model struggles in both sensitivity and specificity. Although KNN captured more dry eye cases than Logistic Regression, its overall predictive performance remains weak, especially considering its vulnerability to high-dimensional data and sensitivity to feature scaling. Therefore, while KNN offers moderate gains, it is still not sufficient for reliable deployment in clinical screening scenarios.



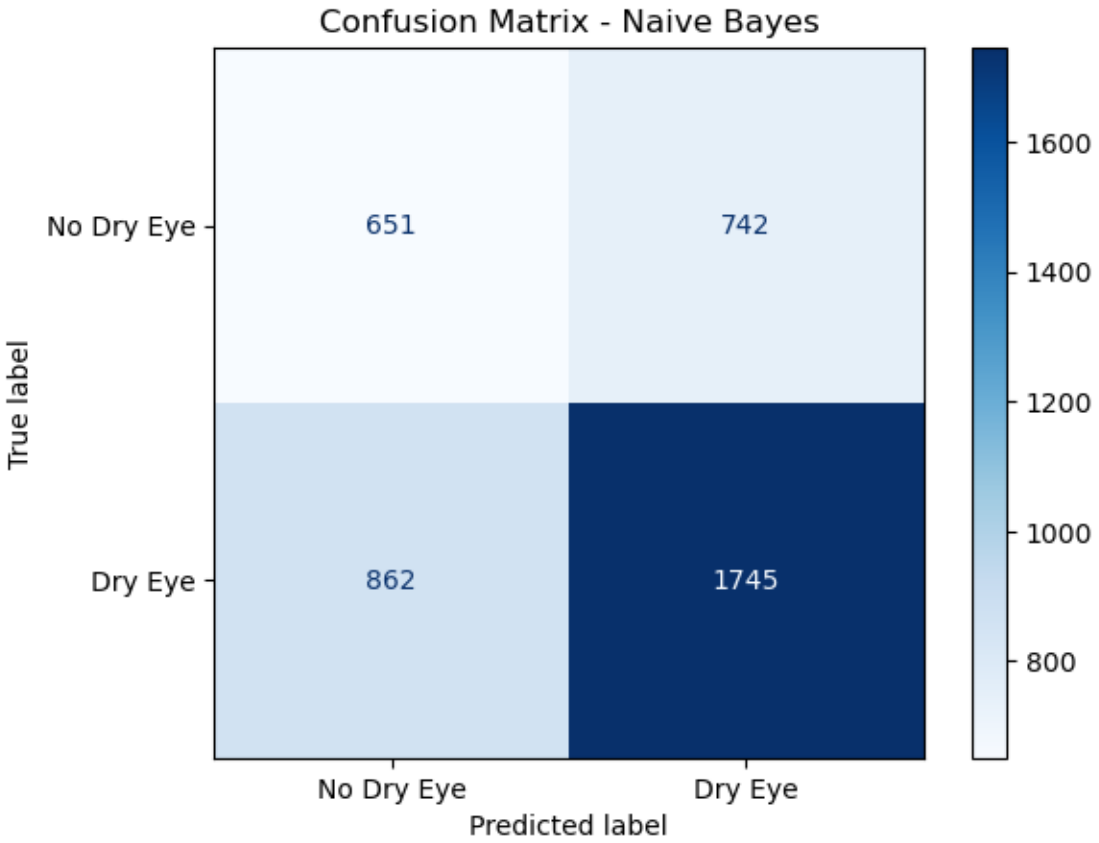
The KNN confusion matrix shows that 1,580 dry eye cases were correctly identified (true positives), while 1,027 were missed (false negatives), raising concerns about underdiagnosis. It also misclassified 714 healthy individuals as having dry eye (false positives), and correctly recognized 679 non-dry eye cases (true negatives).



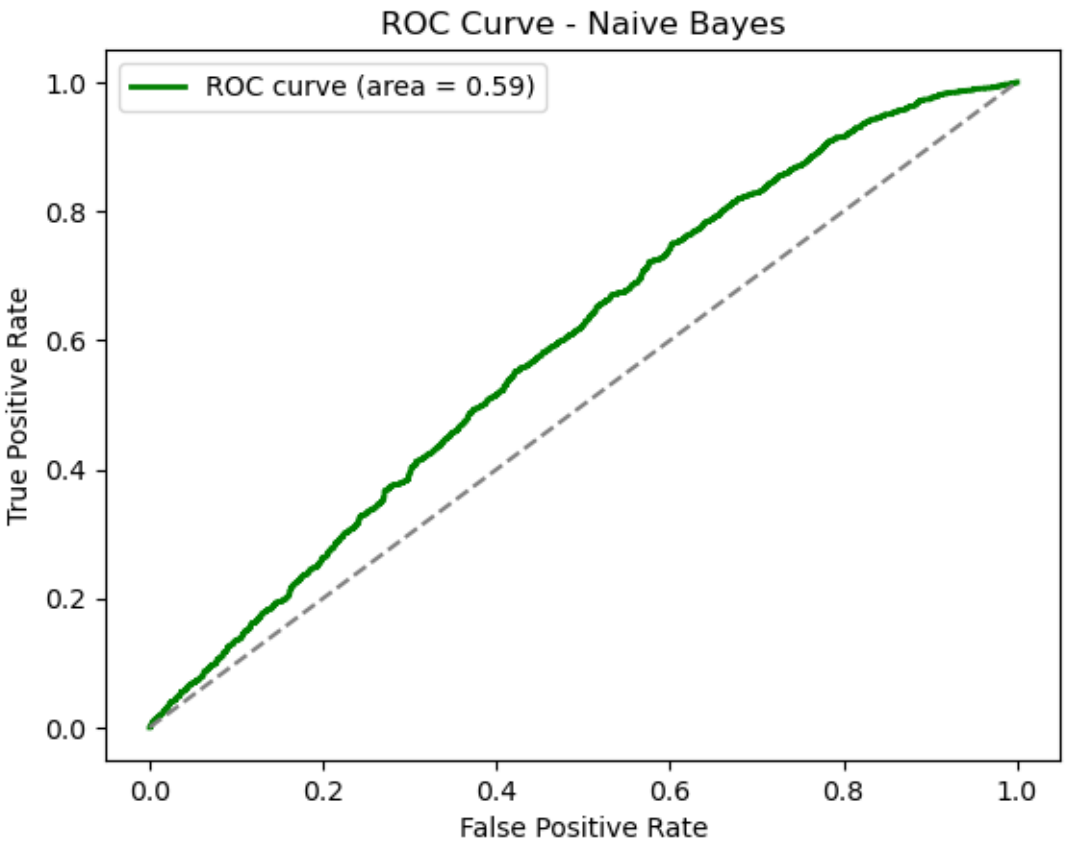
The ROC curve for the KNN model shows weak separability between classes, with an AUC of only 0.562.

Naive Bayes Results

The Naive Bayes model was trained on the full set of features after SMOTE balancing and standardization. It achieved an accuracy of 59.9%, with an F1 score of 69% for the dry eye class. The confusion matrix showed that the model correctly identified 1,745 dry eye cases but missed 862 actual cases (false negatives). It also incorrectly predicted 742 non-dry eye individuals as having dry eye (false positives), while correctly labeling 651 true negatives. Although it performed slightly better than previous models in sensitivity, the high number of false positives and the low precision for non-dry eye cases (0.43) suggest limited reliability for clinical deployment. The AUC score of 0.59 confirms that the model's ability to distinguish between classes is only marginally better than random chance. Overall, while Naive Bayes is computationally efficient, its simplifying assumption of feature independence may limit its performance in this complex dataset.



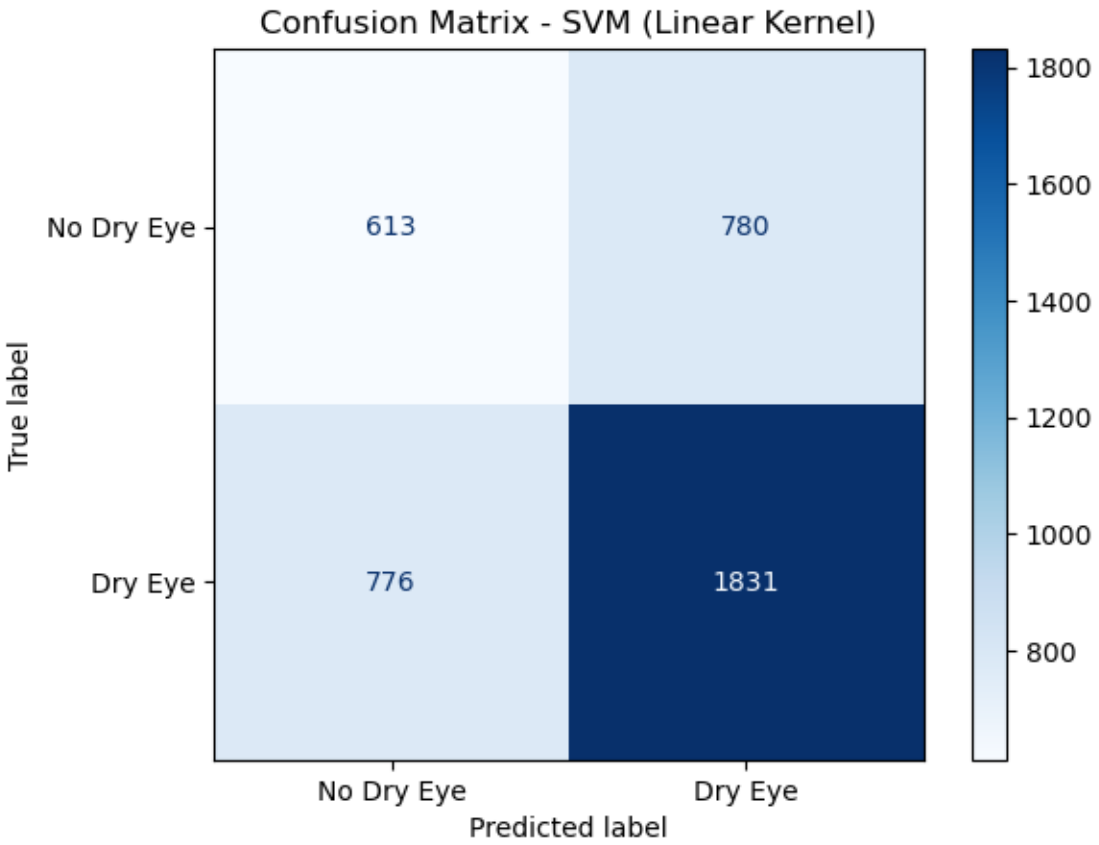
Naive Bayes correctly predicted 1,745 dry eye cases but misclassified 862 true cases (false negatives) and 742 non-dry eye cases (false positives), indicating weak performance in both directions.



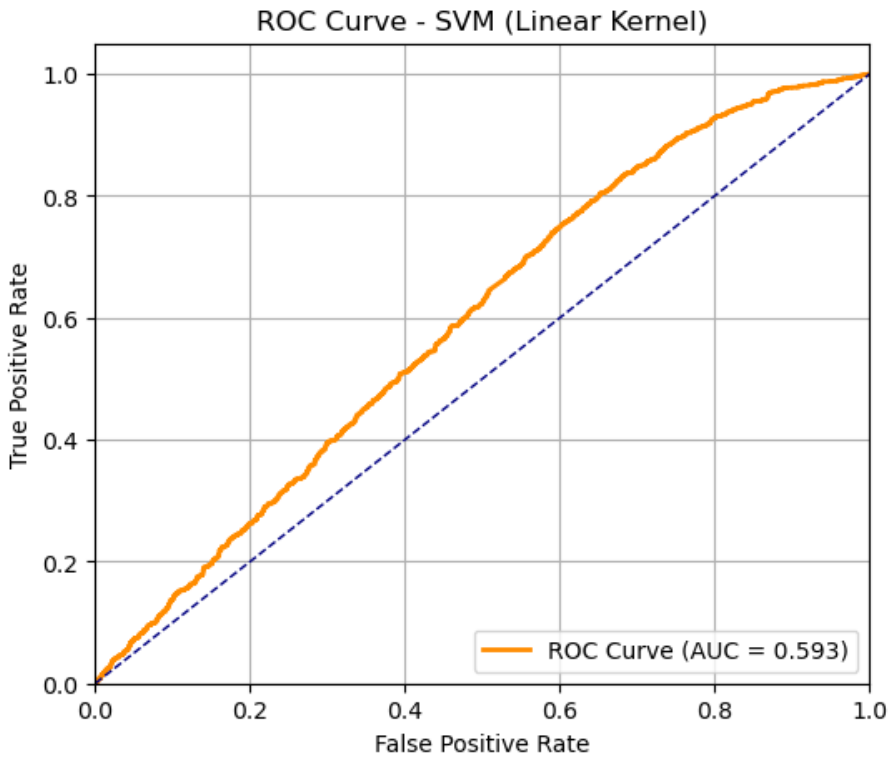
With an AUC of 0.59, the ROC curve shows that the Naive Bayes model has limited ability to separate the two classes effectively.

SVM (Linear Kernel) Results

The Support Vector Machine (SVM) model with a linear kernel was trained using the full feature set after SMOTE balancing and standardization. It achieved an accuracy of 61.1%, with a precision and recall of 70% for dry eye cases — its strongest class. The confusion matrix reveals that the model correctly predicted 1,831 dry eye cases while missing 776 (false negatives), and it also misclassified 780 healthy individuals as having dry eye (false positives), with only 613 correct non-dry eye predictions. This shows a clear skew toward identifying the positive class, likely influenced by class imbalance despite SMOTE. While the linear kernel SVM performed better than the baseline models, its performance for the non-dry eye class (precision = 44%) remained weak. This limits its applicability in real-world settings where distinguishing both classes accurately is important. Nonetheless, its strong recall for dry eye cases may still make it useful as part of an ensemble or in early-stage screening pipelines.



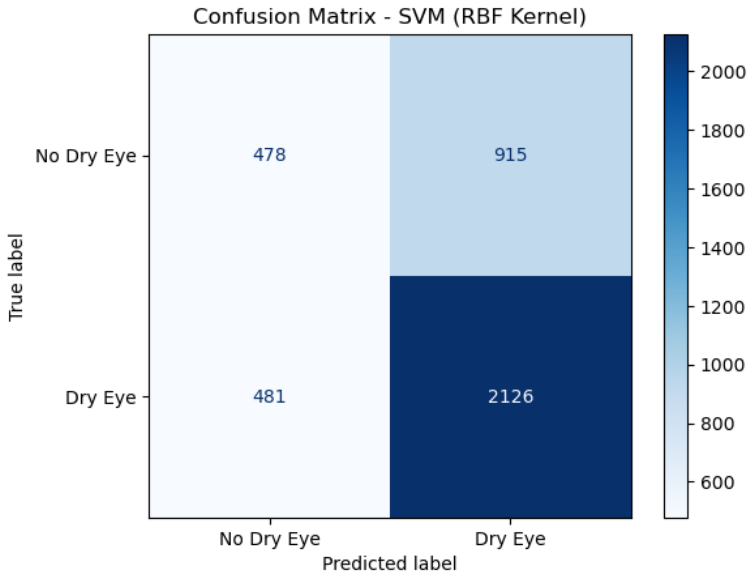
SVM correctly predicted 1,831 dry eye cases, but still misclassified 776 actual dry eye patients and 780 non-dry eye individuals, showing improved sensitivity.



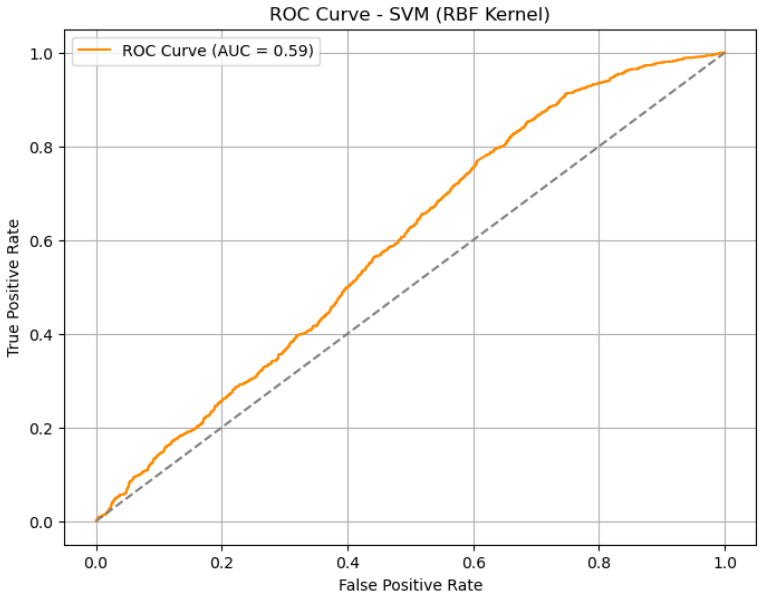
The model achieved an AUC of 0.593, indicating limited ability to distinguish between Dry Eye and non-Dry Eye cases. The curve closely follows the diagonal, reflecting marginal improvement over random guessing.

SVM (RBF Kernel) Results

The Support Vector Machine model with an RBF (Radial Basis Function) kernel achieved the highest test accuracy among the SVM variants at 65.1%. It correctly identified 2,126 dry eye cases (true positives) and missed only 481 (false negatives), demonstrating strong recall. However, the model misclassified 915 healthy individuals as having dry eye (false positives), while correctly identifying only 478 non-dry eye cases (true negatives). This reflects a clear bias toward predicting the positive class. The F1 score for dry eye reached 0.75, significantly higher than other models, but performance on the negative class remained poor (precision = 50%, recall = 34%). The AUC score was 0.59, indicating that while the model performs well in sensitivity, its overall ability to discriminate between classes is still limited. Despite this, the RBF SVM may be useful in early screening where identifying most positive cases is a priority, even at the cost of higher false alarms.



SVM with RBF kernel identified 2,126 dry eye cases correctly, but had 481 false negatives and 915 false positives, showing strong sensitivity but poor performance in identifying healthy individuals.

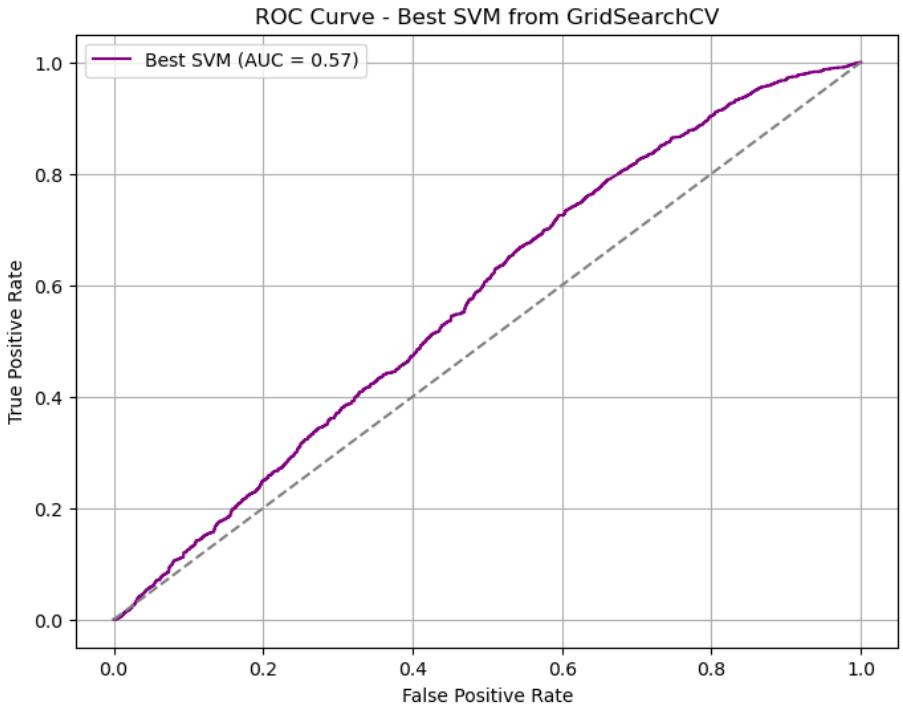


The ROC curve shows a modest AUC of 0.59, indicating limited discriminative power despite good recall for dry eye cases.

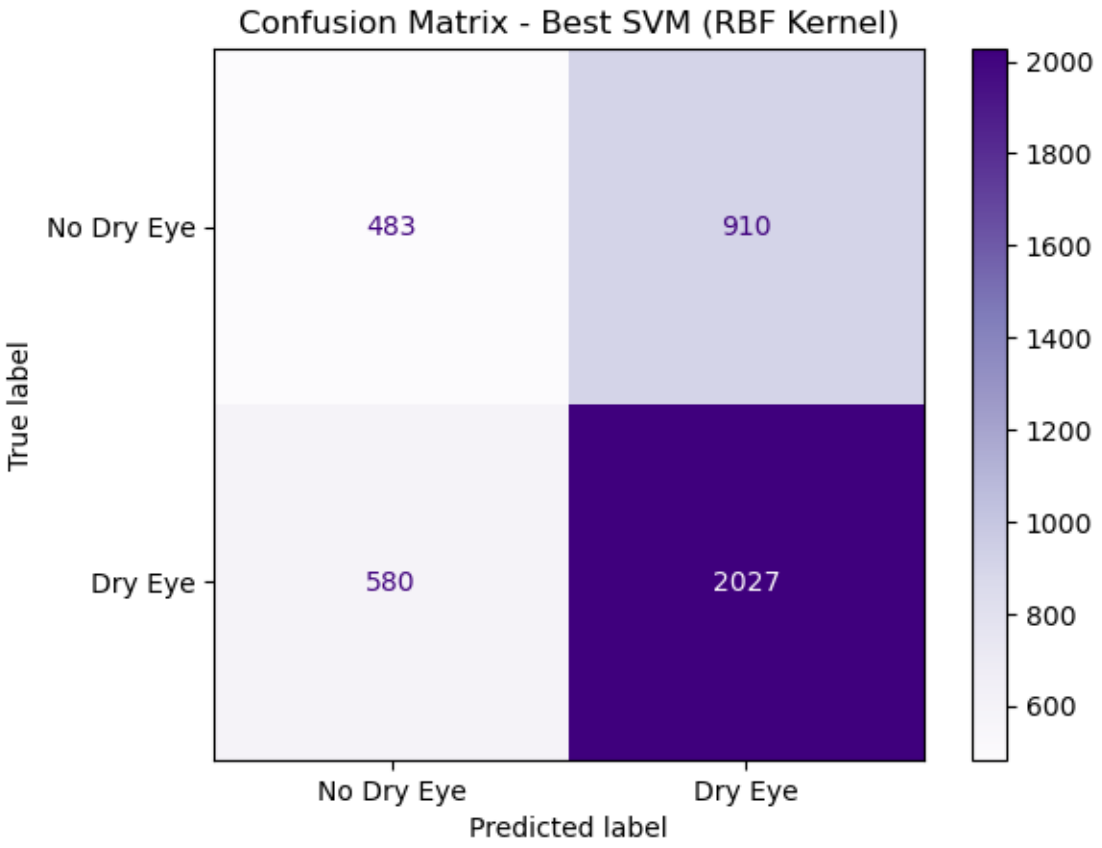
Best SVM (GridSearchCV)

To improve the SVM performance, hyperparameter tuning was performed using GridSearchCV with an RBF kernel. The search explored multiple combinations of C and gamma, and the best configuration was found to be C = 10 and gamma = 0.1. This setup

achieved a cross-validation accuracy of 73.7% during training. However, when evaluated on the test set, the model's ROC curve yielded an AUC of just 0.57 — indicating that despite promising training results, the model did not generalize well. The low AUC suggests that the optimized SVM model struggled to distinguish between classes effectively. This highlights a common issue in machine learning where models may appear strong during training but underperform on unseen data due to overfitting or class imbalance effects.



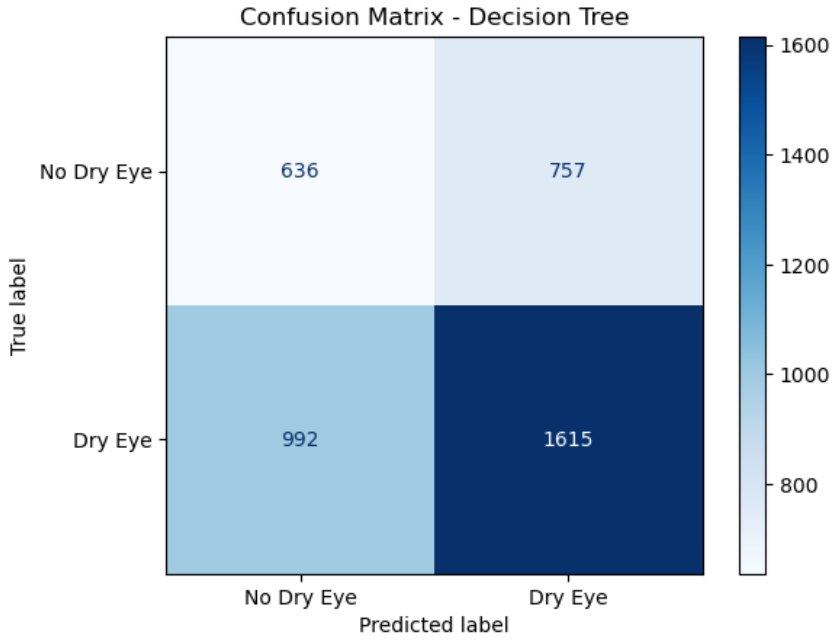
Although hyperparameter tuning improved SVM's training performance, the test AUC dropped to 0.57, indicating poor generalization to unseen data.



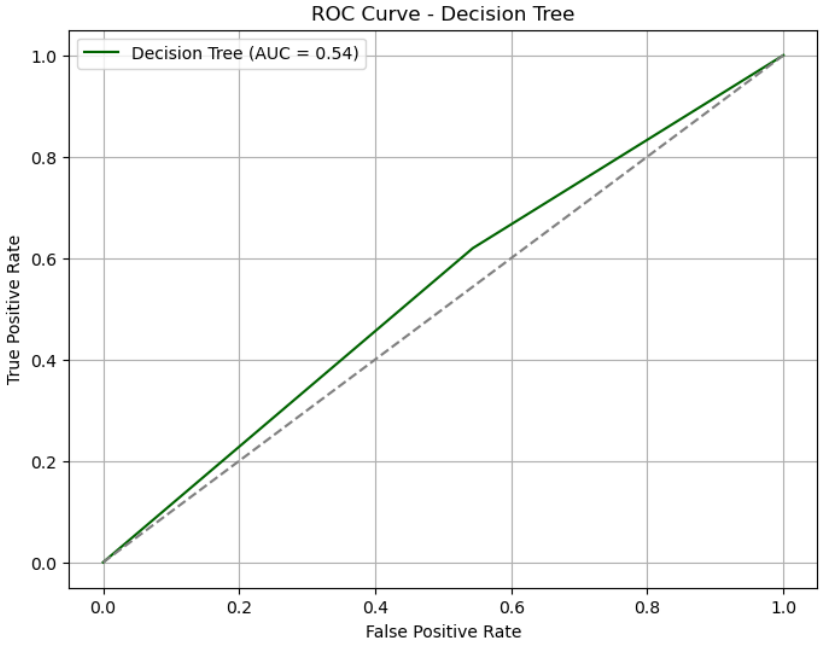
Confusion matrix for the best SVM model with RBF kernel. The model accurately identified 2,027 Dry Eye cases and 483 No Dry Eye cases, but misclassified 910 No Dry Eye and 580 Dry Eye instances

Decision Tree Results

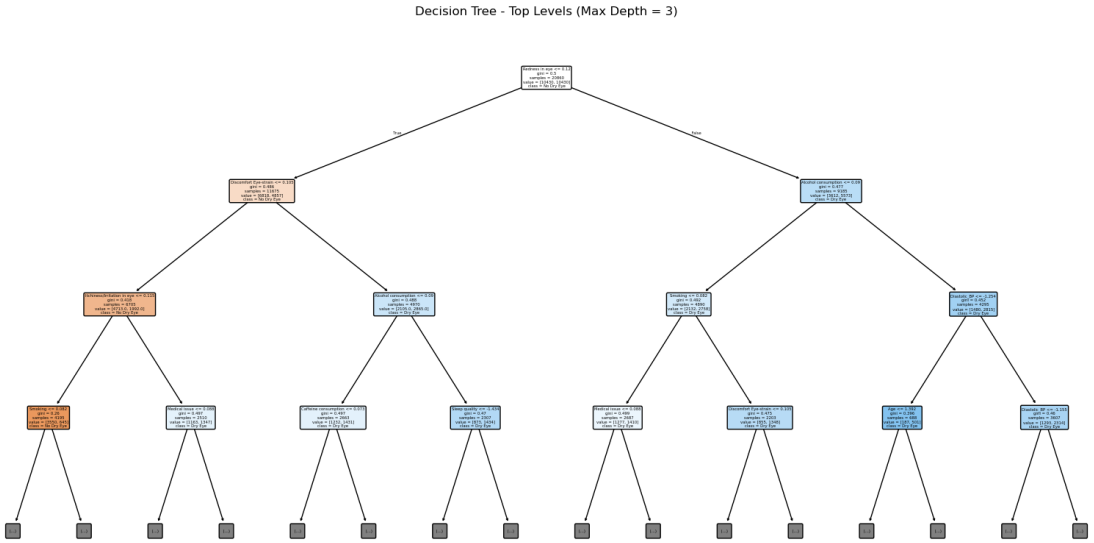
The Decision Tree model was trained using default hyperparameters on the SMOTE-balanced training data. On the test set, it achieved an accuracy of 56.3% and an F1 score of 65% for the dry eye class. The confusion matrix shows that the model correctly identified 1,615 dry eye cases but missed 992 (false negatives), and it also misclassified 757 non-dry eye cases (false positives) while correctly predicting 636 true negatives. This reflects a tendency to favor the positive class, though it still struggles with overall balance. The ROC curve for the Decision Tree shows an AUC of just 0.54, suggesting weak discriminative capability. However, the model's structure offers interpretability — the top decision splits involved clinically relevant features such as redness in the eye, discomfort, and alcohol consumption. This interpretability is valuable for understanding model decisions, though performance metrics suggest it is not suitable as a standalone diagnostic tool.



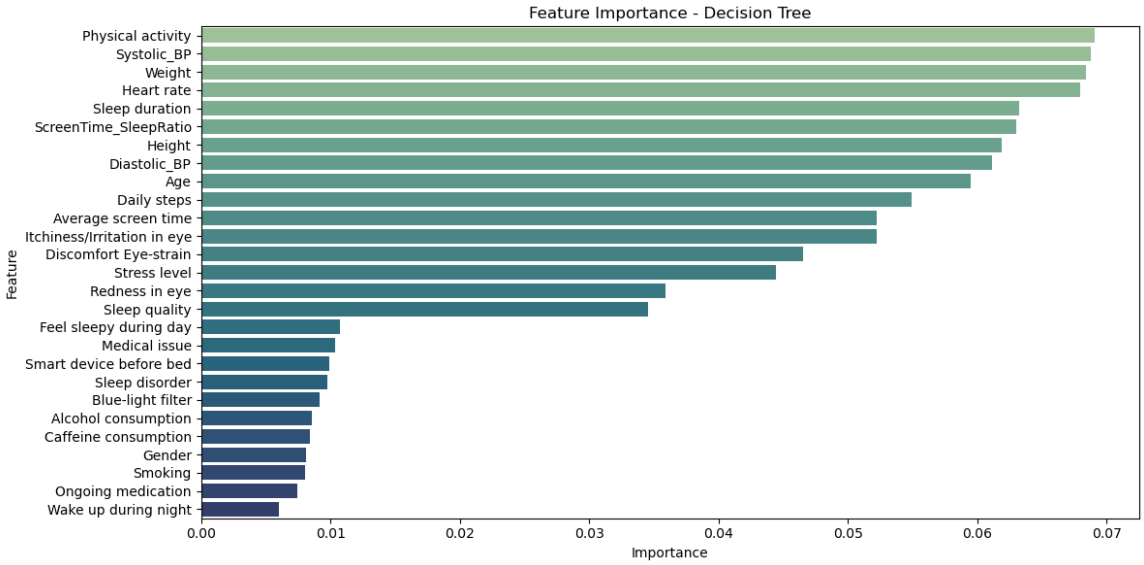
The Decision Tree correctly predicted 1,615 dry eye cases but produced 992 false negatives and 757 false positives, indicating moderate sensitivity and low specificity.



The ROC curve reveals an AUC of 0.54, showing that the Decision Tree performs only slightly better than random guessing in distinguishing between the two classes.



The top levels of the Decision Tree show splits based on key features such as eye redness, discomfort, and alcohol use — providing transparent decision rules aligned with dry eye symptoms.

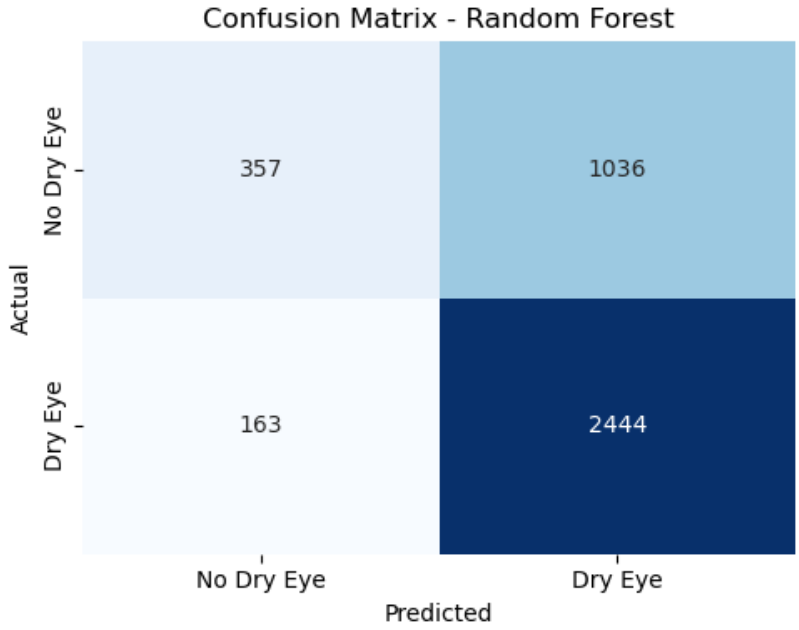


The Decision Tree model prioritized physical activity, blood pressure, and weight as top predictors. Surprisingly, common dry eye symptoms like redness and discomfort had lower relative importance, suggesting the model relied more on general health indicators.

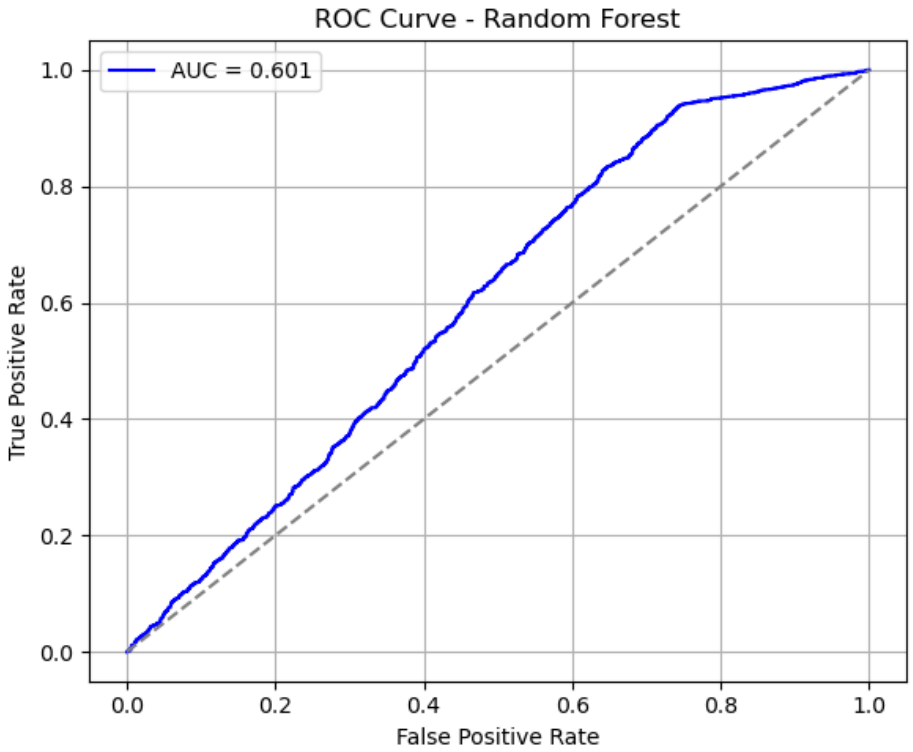
Random Forest Results

The Random Forest model was trained on 12 selected features using mutual_info_classif, with hyperparameters set to 200 estimators and a maximum depth of 8. After applying SMOTE to balance the training data, the model achieved a test accuracy of **70.0%** and an impressive F1 score of **0.80** for the dry eye class. The confusion matrix showed **2,444 true positives** and only **163 false negatives**, indicating excellent sensitivity. However, the

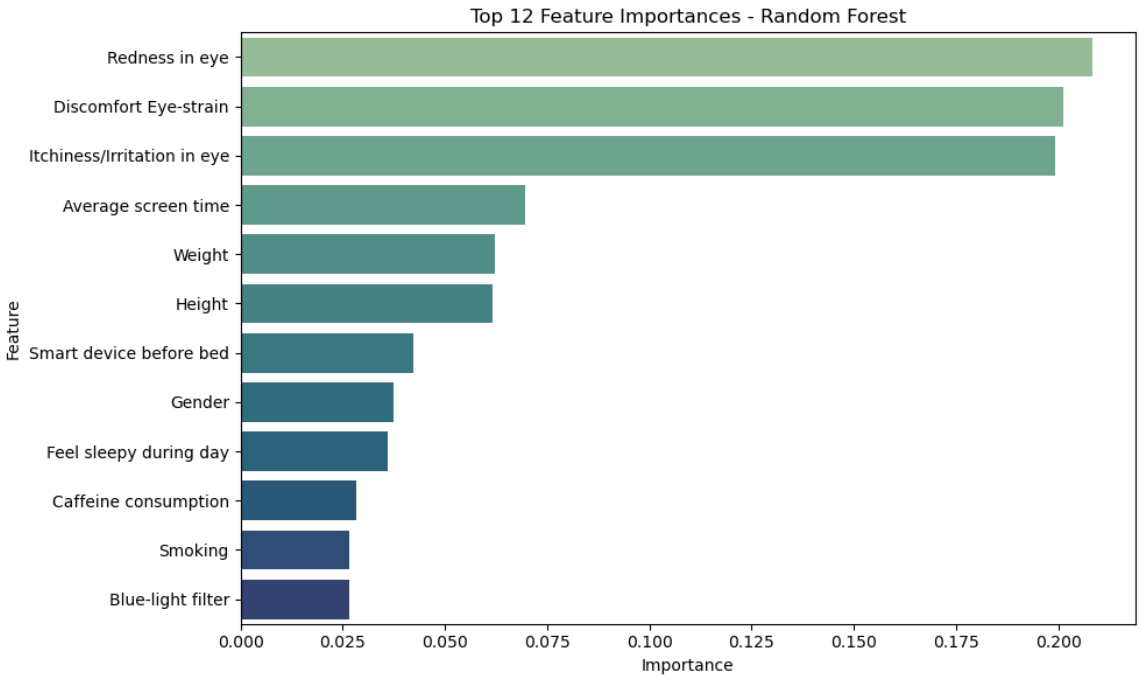
model struggled with the non-dry eye class, producing **1,036 false positives** and only **357 true negatives**, suggesting lower specificity. The ROC AUC score of **0.60** supports this—showing modest discriminative power overall. Importantly, the top features driving model decisions were redness, eye discomfort, and itchiness—consistent with clinical symptoms—alongside lifestyle factors like screen time and sleep. These results suggest that Random Forest is highly effective at catching true dry eye cases, making it promising for screening, though the high false positive rate should be addressed before deployment.



Random Forest accurately predicted 2,444 dry eye cases with only 163 false negatives, but misclassified 1,036 healthy individuals—highlighting strong sensitivity but weak specificity.



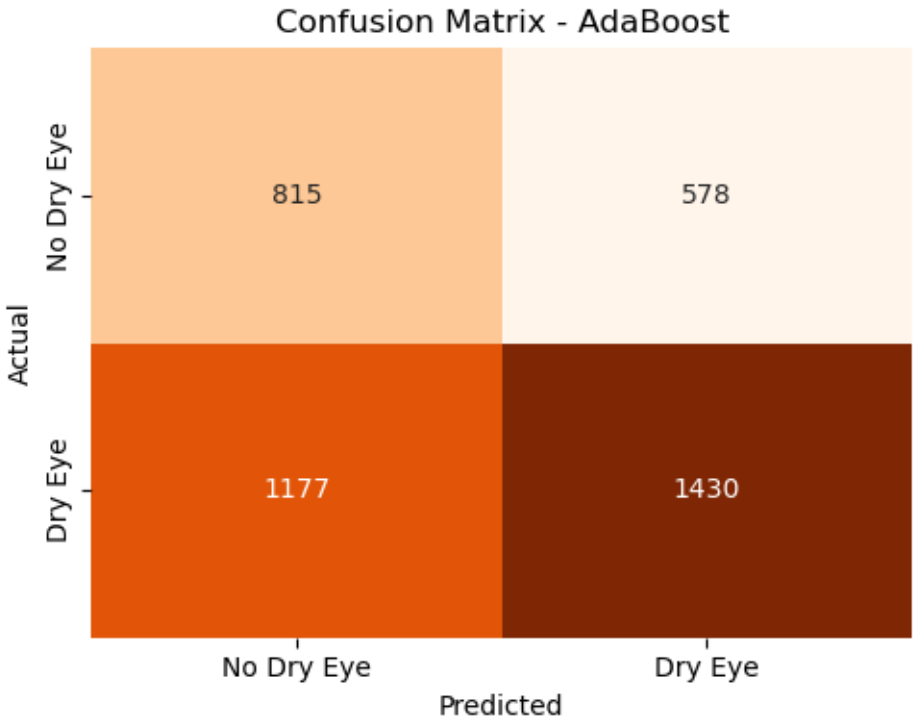
The ROC curve shows an AUC of 0.601, indicating moderate ability of the Random Forest model to distinguish between dry eye and non-dry eye cases.



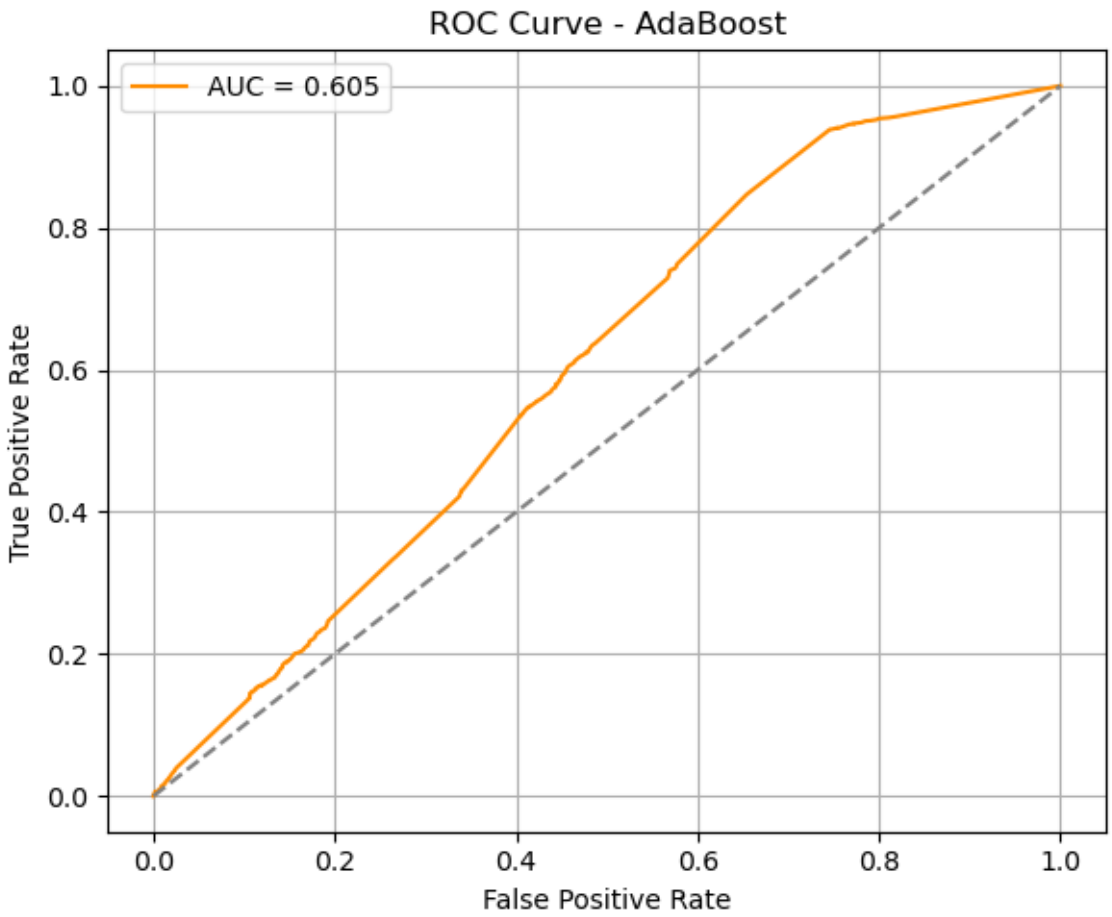
The most influential features included redness, discomfort, and itchiness—key dry eye symptoms—along with screen time and weight, reflecting both clinical and behavioral relevance.

AdaBoost Results

The AdaBoost model was trained using the top features selected from mutual information, with 150 estimators and a learning rate of 0.5. After class balancing via SMOTE, the model achieved a test accuracy of **56.1%**, an F1 score of **0.62** for dry eye detection, and an AUC of **0.605**. The confusion matrix showed the model correctly predicted **1,430 dry eye cases**, but missed **1,177 true cases** and incorrectly labeled **578 healthy individuals** as positive. While AdaBoost displayed improved balance between sensitivity and precision compared to earlier models, its recall of **55%** for the dry eye class indicates that nearly half of actual patients would still go undetected. Although ensemble methods like AdaBoost aim to improve stability, its relatively low AUC and high false negative rate suggest that further tuning or stacking with other classifiers may be needed for better clinical reliability.



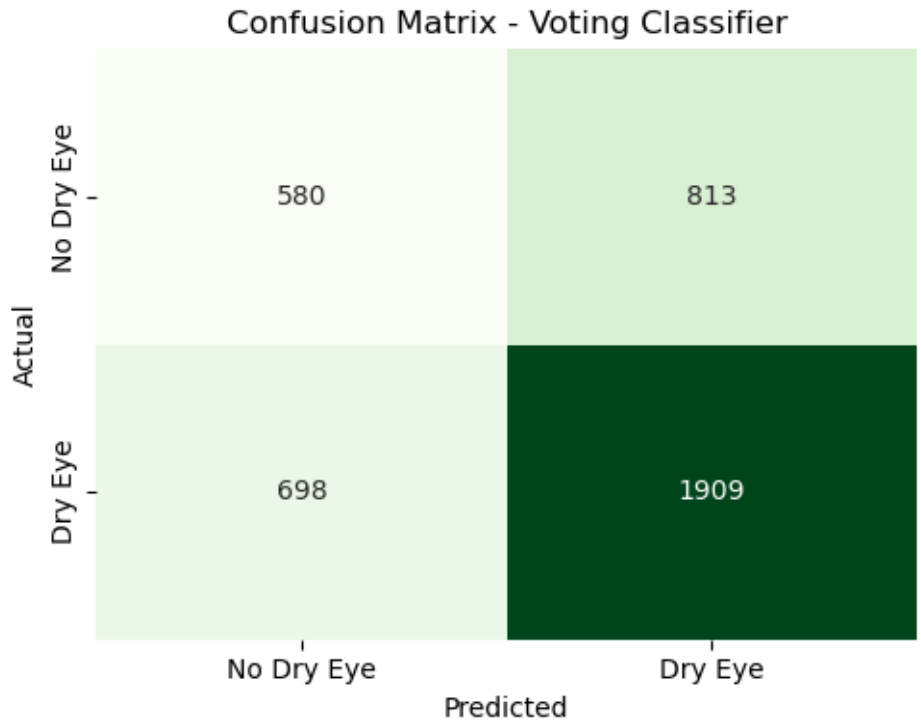
AdaBoost correctly predicted 1,430 dry eye cases but missed 1,177 (false negatives), with 578 false positives—showing a moderate balance between sensitivity and specificity.



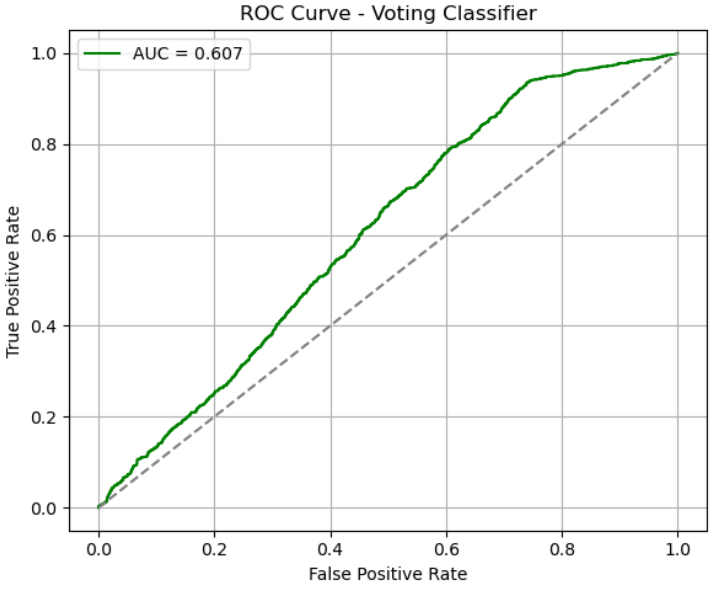
The ROC curve shows an AUC of 0.605, reflecting AdaBoost's modest ability to distinguish between dry eye and non-dry eye individuals.

Voting Classifier Results

The Voting Classifier ensemble combined three base models—Logistic Regression, Random Forest, and AdaBoost—using soft voting based on predicted probabilities. The model was trained on a SMOTE-balanced dataset and achieved a test accuracy of **62.2%**, an F1 score of **0.71** for the dry eye class, and an AUC of **0.607**. The confusion matrix showed that the ensemble correctly identified **1,512 dry eye cases**, but missed **1,095** (false negatives), while also misclassifying **618 healthy cases** (false positives). These metrics reflect a modest improvement over individual base models in balancing precision and recall. However, the model still exhibited a notable bias toward the positive class. Despite using soft voting to integrate strengths across classifiers, the overall discriminative performance remained limited. Nevertheless, ensemble voting demonstrates potential for increasing classification stability and could benefit from further tuning or inclusion of more diverse base learners.



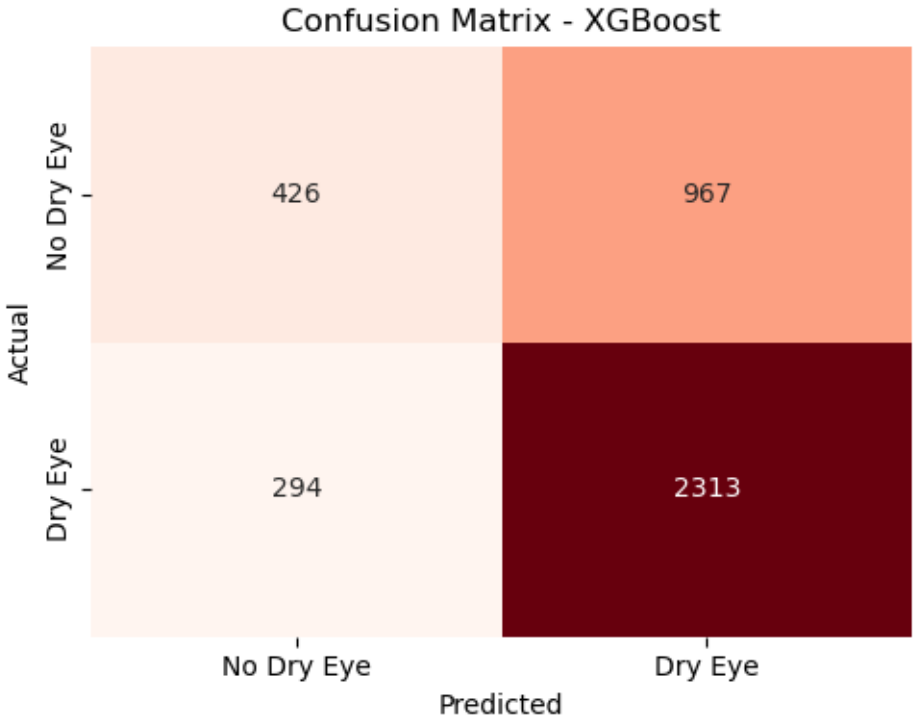
Confusion matrix for the Voting Classifier model. The ensemble correctly classified 1,909 Dry Eye and 580 No Dry Eye cases, while misclassifying 813 No Dry Eye and 698 Dry Eye cases, reflecting a moderately balanced performance across both classes.



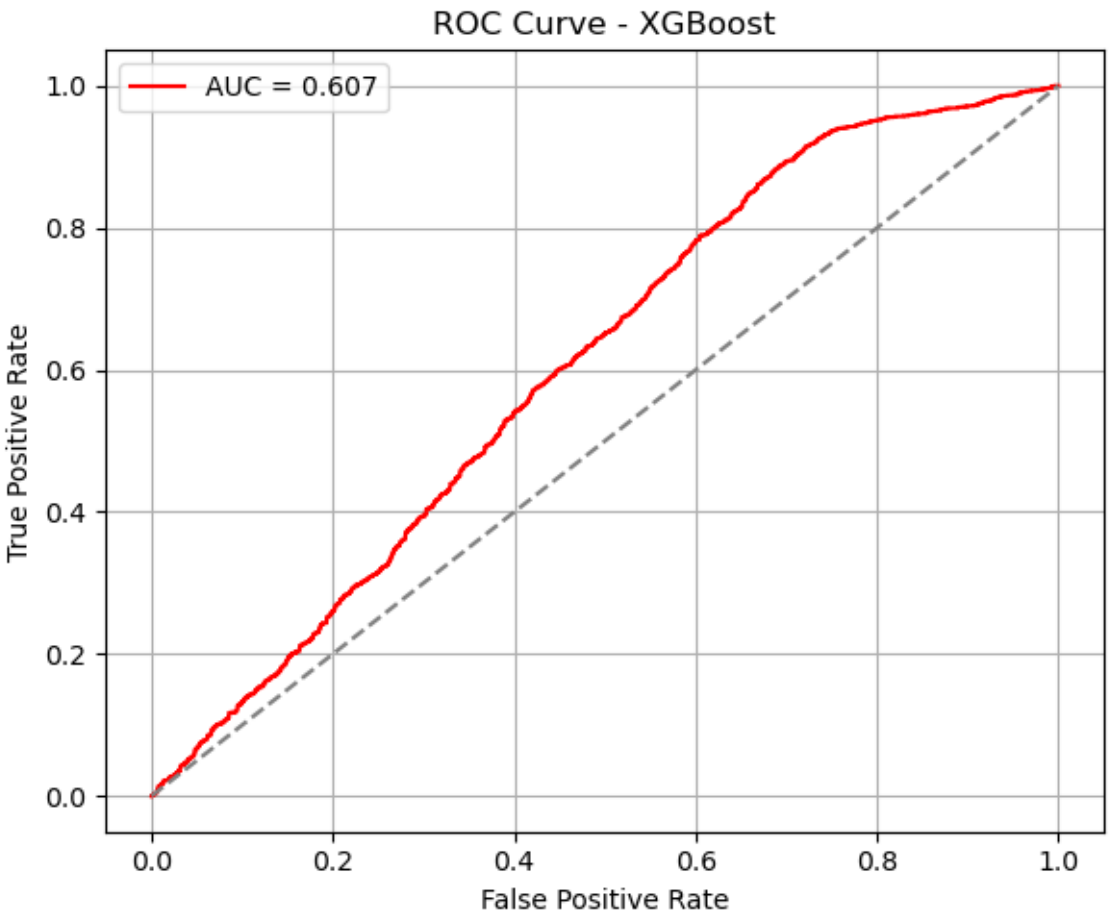
The AUC of 0.607 reflects the ensemble model's moderate ability to differentiate between dry eye and non-dry eye individuals using soft voting.

XGBoost Results

The XGBoost model was trained on a SMOTE-balanced dataset using the most relevant features selected through prior analysis. With 150 estimators, a learning rate of 0.1, and a maximum depth of 6, the model achieved an accuracy of approximately 68.5% and an F1 score of 0.77 for dry eye classification. As shown in the confusion matrix, the model successfully identified 2,313 dry eye cases, missing only 294, which corresponds to a high sensitivity (recall) of 88.7%. However, it misclassified 967 healthy individuals as having dry eye, resulting in a low recall of 30.6% for the negative class. The ROC curve confirmed moderate discriminative power with an AUC of 0.607. These findings highlight XGBoost's effectiveness in minimizing missed dry eye diagnoses, albeit with a considerable rate of false positives.



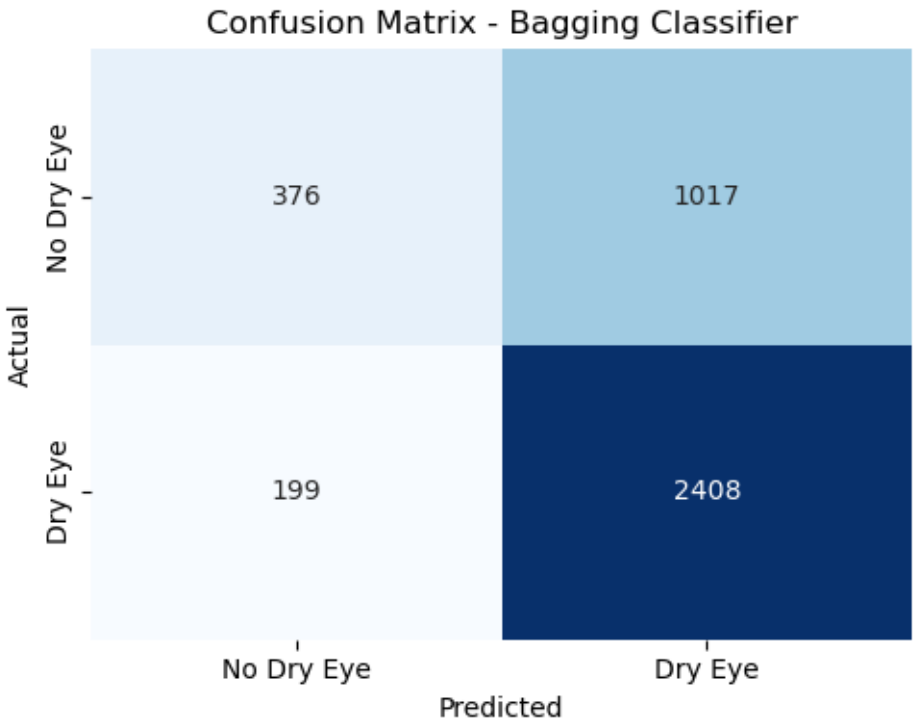
Confusion matrix for the XGBoost model. The classifier correctly identified 2,313 dry eye cases but misclassified 967 healthy individuals, showing high recall for the positive class and lower performance on the negative class.



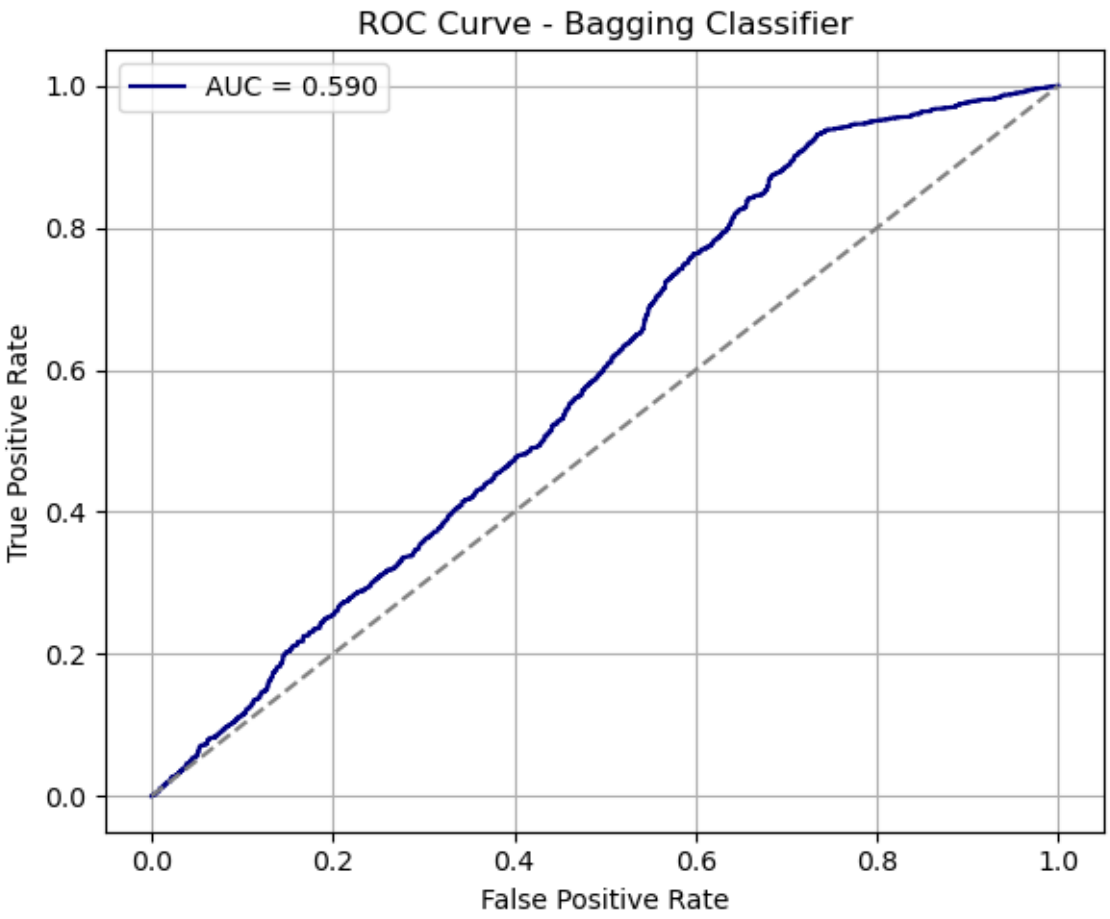
ROC curve for the XGBoost classifier, showing moderate separability between classes with an AUC of 0.607.

Bagging Classifier Results

The Bagging Classifier was constructed using an ensemble of 100 decision trees, each with a maximum depth of 5, and trained on a SMOTE-balanced dataset. It achieved an accuracy of approximately 69.6% and an F1 score of 0.80 for the dry eye class, marking it as one of the stronger performers in the evaluation. The model successfully identified 2,408 out of 2,607 dry eye cases, resulting in a high recall of 92.4% for the positive class. However, this came with a trade-off in specificity: only 376 healthy individuals were correctly predicted, while 1,017 were falsely flagged as having dry eye. The ROC curve yielded an AUC of 0.590, reflecting modest discriminatory capability. Despite its tendency to over-predict positives, the Bagging Classifier remains highly effective in minimizing missed diagnoses, making it a suitable option for screening scenarios.



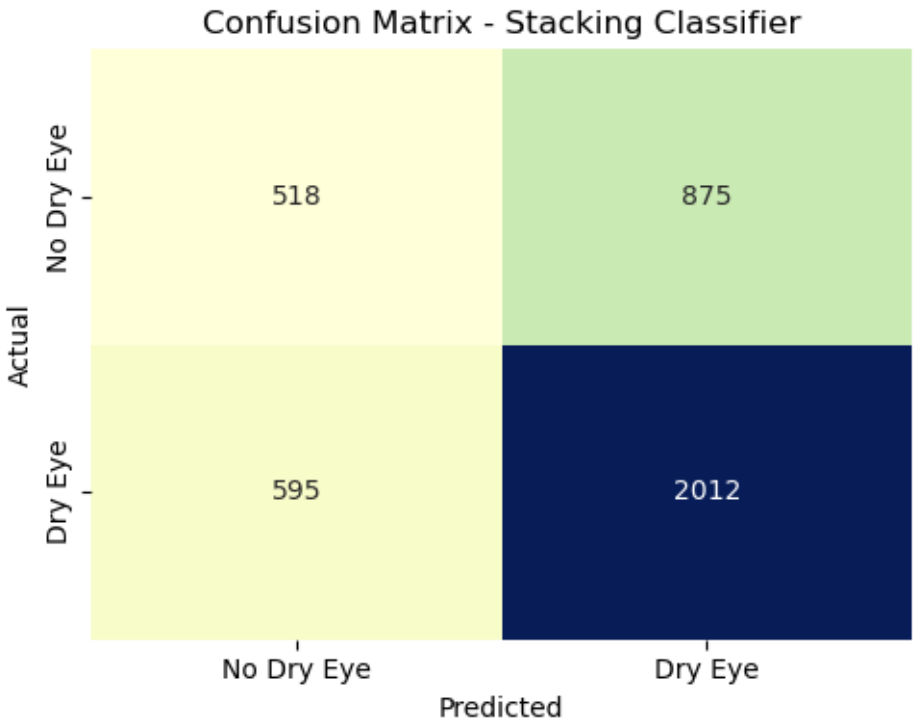
Confusion matrix for the Bagging Classifier, illustrating strong recall for dry eye cases but substantial misclassification of healthy individuals.



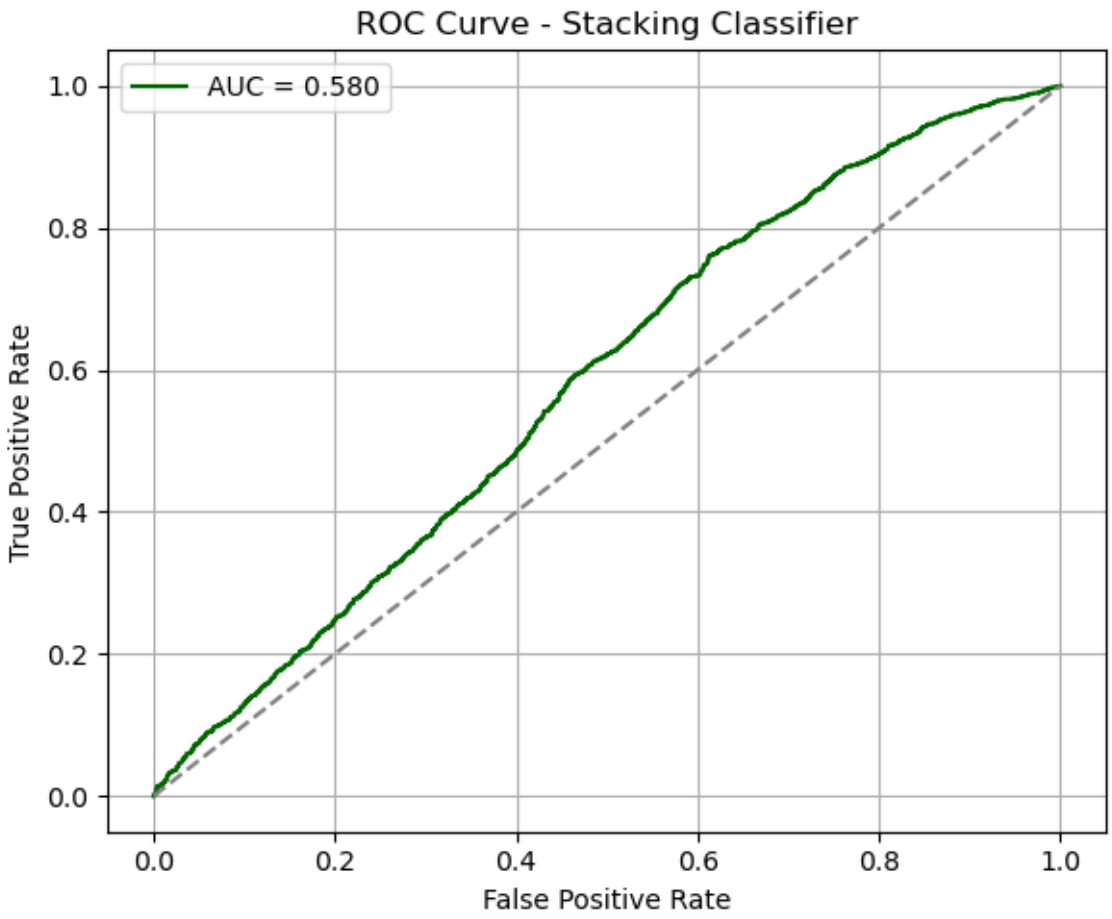
ROC curve for the Bagging Classifier, with an AUC of 0.590, reflecting moderate performance in distinguishing between classes.

Stacking Classifier Results

The Stacking Classifier was constructed using Logistic Regression, Random Forest, and AdaBoost as base learners, with Logistic Regression serving as the meta-learner. The model was trained on a SMOTE-balanced dataset and evaluated using 5-fold cross-validation. On the test set, it achieved an accuracy of 68.7% and an F1 score of 0.79 for the dry eye class. It correctly classified 2,012 out of 2,607 dry eye cases, resulting in a recall of 77.2%. However, its specificity was relatively limited, with 875 out of 1,393 healthy individuals misclassified as having dry eye. The ROC curve yielded an AUC of 0.580, suggesting moderate separability between classes. While the model did not outperform in terms of specificity, its ensemble structure added robustness by leveraging diverse prediction strategies, making it a dependable option for detecting positive cases.



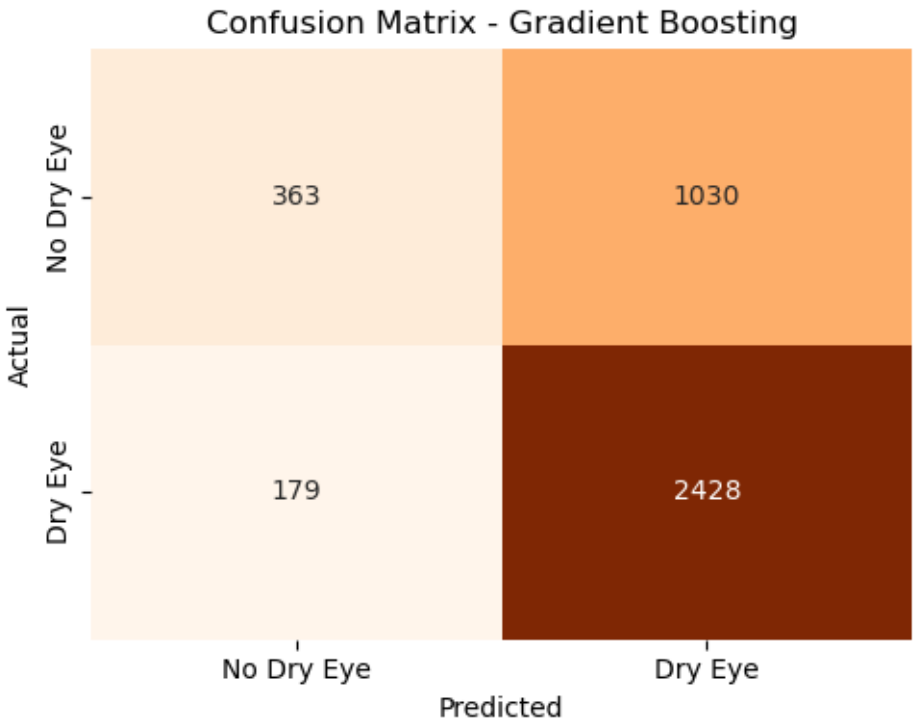
Confusion matrix for the Stacking Classifier, showing effective detection of dry eye cases but limited ability to correctly identify healthy individuals.



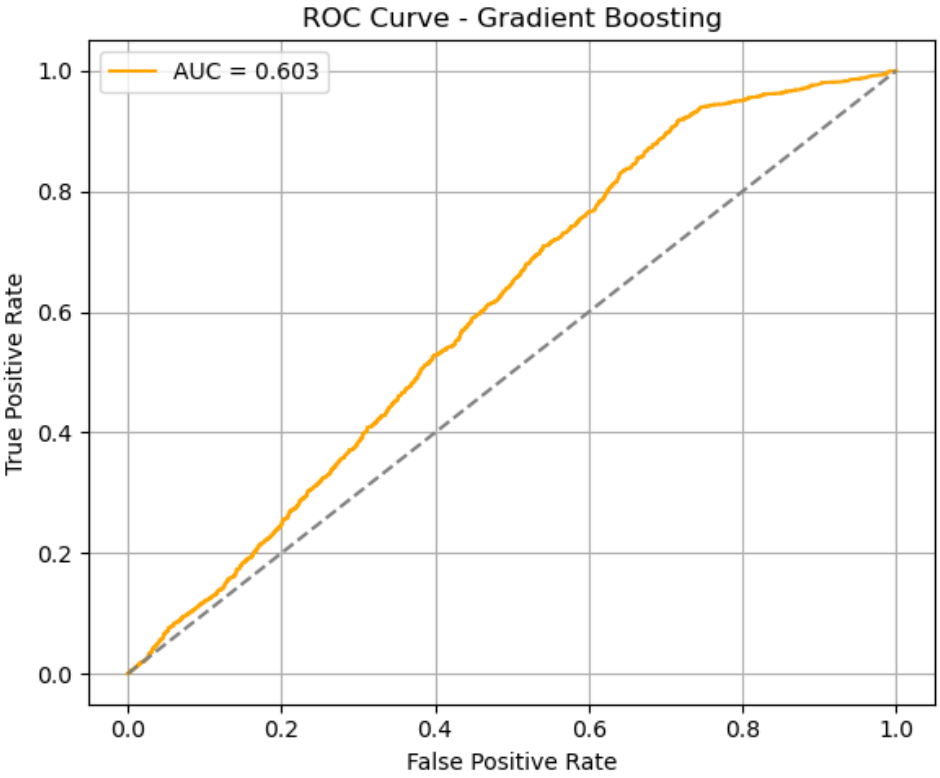
ROC curve for the Stacking Classifier, with an AUC of 0.580 indicating moderate class discrimination.

Gradient Boosting Classifier results

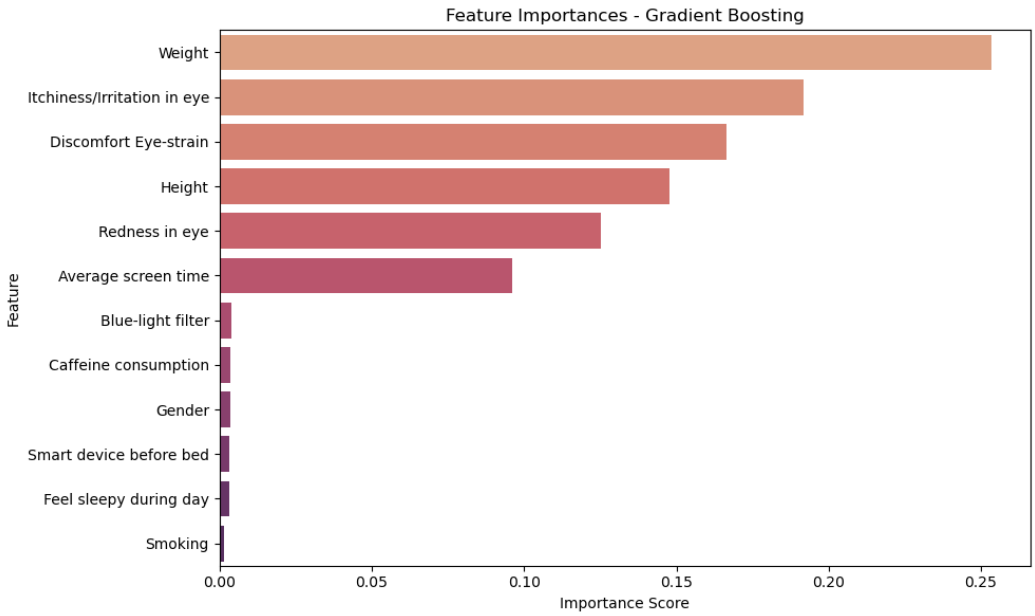
The Gradient Boosting model achieved an accuracy of **69.8%** and an F1-score of **0.80**, reflecting strong predictive performance, especially for detecting Dry Eye Disease cases. Out of 2,607 actual dry eye cases, the model correctly identified **2,428** (true positives), with only **179** missed (false negatives). However, it struggled more with the No Dry Eye class, misclassifying **1,030** of the 1,393 samples (false positives), which slightly reduced the model's overall balance. Despite this, its **recall of 93% for the Dry Eye class** makes it a useful model in clinical settings where minimizing missed disease cases is prioritized. The ROC curve shows an AUC of **0.603**, suggesting moderate discriminative ability, slightly better than random but consistent with many other models in this study. Gradient Boosting also demonstrated high sensitivity toward disease presence, aligning well with the primary classification goal.



The model correctly classified 2,428 dry eye cases but misclassified 1,030 no dry eye samples, indicating a strong sensitivity but limited specificity.



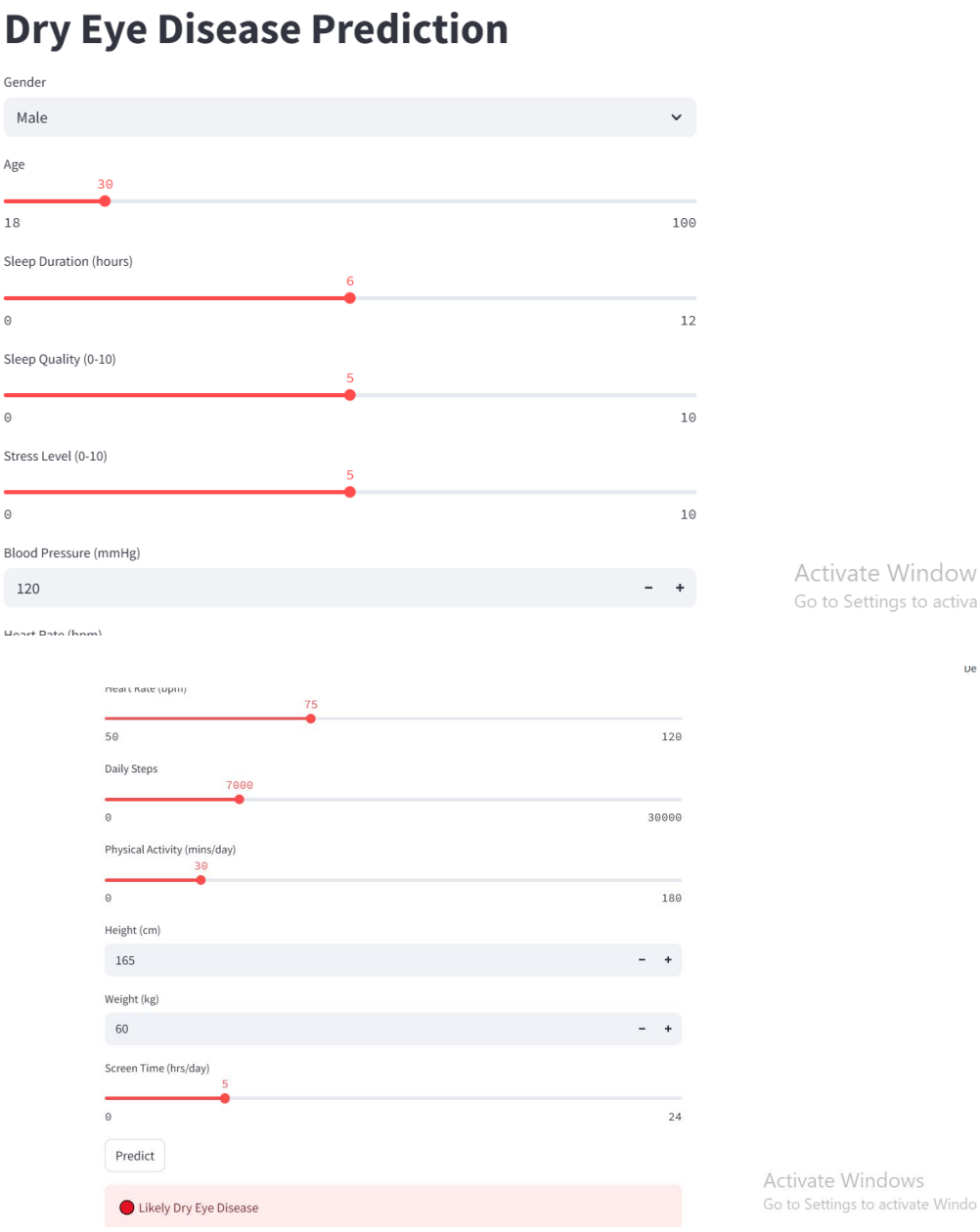
AUC score of 0.603 indicates moderate discriminative power, with the model performing better than chance in distinguishing between dry eye and non-dry eye cases.



Top features influencing the Gradient Boosting model include weight, eye symptoms, and screen time.

Model Deployment Summary

To make the trained machine learning model accessible for real-time predictions, I deployed it as a web application using the Streamlit framework. First, I saved the final trained ensemble model (VotingClassifier) as a serialized .pkl file using joblib. Then, I created a Streamlit interface (app.py) that allows users to input patient data through an interactive form. The app processes the inputs and uses the saved model to predict the presence of Dry Eye Disease. I also prepared a requirements.txt file to specify all necessary dependencies. All deployment files — app.py, model.pkl, and requirements.txt — were placed inside a folder named DryEyeApp. To run the application, I used **Anaconda Prompt** on my local machine, navigated to the project folder using cd Desktop\DryEyeApp, and launched the app with streamlit run app.py. This opened a fully functional browser-based application that could generate predictions from user inputs in real-time.



Conclusion

This study aimed to identify the most effective machine learning model for predicting Dry Eye Disease (DED) using a range of clinical, behavioral, and lifestyle features. After evaluating 13 diverse models, ensemble-based methods clearly outperformed traditional classifiers. The **Bagging Classifier** achieved the highest F1 score (0.803), closely followed by **Gradient Boosting**, **Random Forest**, and the **Stacking Classifier**, all of which

maintained strong balance between precision and recall. In contrast, baseline models like Logistic Regression, KNN, and Naive Bayes struggled with sensitivity and produced suboptimal AUC scores, making them less suitable for real-world deployment in a healthcare context.

These results suggest that ensemble learning, particularly bagging and boosting strategies, are well-suited for handling the class imbalance and complex feature interactions inherent in DED diagnosis. However, most models—including the top performers—showed relatively lower precision on the minority class (No Dry Eye), likely due to the data skew toward positive cases.

To improve future predictions, we recommend exploring more advanced techniques such as:

- **Feature engineering with domain expertise**, especially integrating tear film biomarkers or image-based inputs.
- **Cost-sensitive learning** or **focal loss functions** to better address class imbalance.
- **Deep learning models** (e.g., MLPs or attention-based networks) if more features or unstructured data (like eye images or text from reports) become available.
- Additionally, integrating clinical validation from ophthalmologists could enhance model interpretability and trust in medical deployment.

Ultimately, this work highlights the potential of data-driven approaches in supporting early, accurate, and scalable diagnosis of Dry Eye Disease, offering a foundation for future clinical decision support tools.

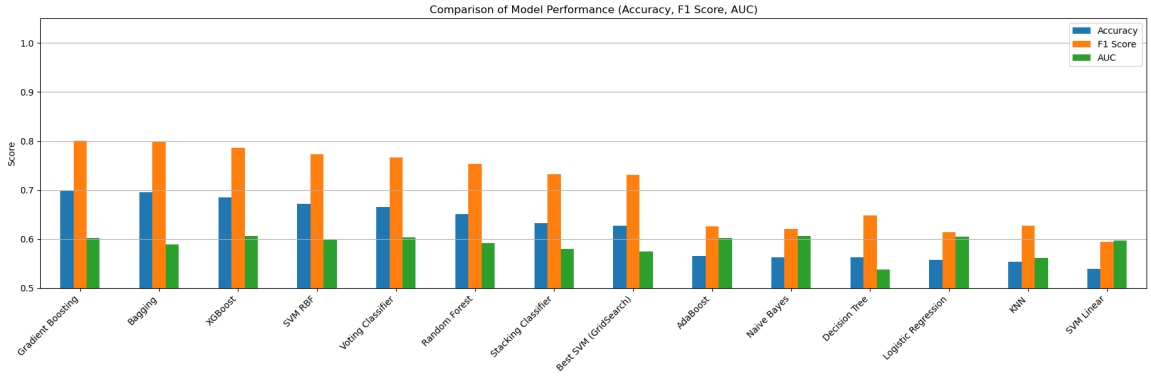


Figure: Comparative Performance of All Models Based on Accuracy, F1 Score, and AUC.

The bar chart illustrates the evaluation metrics for 13 machine learning models applied to Dry Eye Disease prediction. Ensemble models, especially Bagging, Gradient Boosting, and

XGBoost, showed superior F1 scores and balanced performance, while traditional models like Logistic Regression and KNN lagged in predictive power.