**1.1 Introduction**

Most nations' economies rely on the agriculture industry, which is vital for ensuring food security and livelihood. Despite its importance, the industry faces several obstacles such as climate change, pests, diseases, and erratic weather patterns that adversely impact crop yields and food production [1], [2]. Precise and dependable crop prediction is critical in addressing these challenges and safeguarding food security. Predicting crops involves evaluating the projected yield for each season by considering factors such as soil type and weather conditions [3].

Historically, predicting crop yields has been accomplished through manual means, which are slow, ineffective, and vulnerable to mistakes. However, due to the availability of substantial data and technological advancements, machine learning (ML) has become a promising alternative for precise and efficient crop prediction [4]. ML algorithms can examine vast amounts of information, recognize patterns, and provide precise predictions [5]. Numerous investigations have been conducted to design ML-based models for crop prediction, utilizing various techniques such as neural networks, decision trees, support vector machines, and Bayesian networks [6].

Ensemble techniques have become increasingly popular for enhancing the precision and stability of ML models in recent times [7], [8]. For more accurate predictions than single models, ensemble approaches combine multiple models. Ensemble techniques can compensate for the shortcomings of individual models and deliver more durable and dependable predictions by using multiple models [9].

The primary focus of this study is crop prediction for smart agriculture using an ensemble technique that combines best-performing ML algorithms such as Decision Tree, Logistic Regression, Random Forest, Gaussian Naïve Bayes, and K-Nearest Neighbour. These algorithms were chosen based on their popularity, simplicity, and effectiveness in classification tasks [10]. The objective of this study is to develop a dependable and precise ML-based model for crop prediction utilizing an ensemble technique. The model is trained and assessed on a dataset comprising past crop yields and weather data for a particular region [11]. The model's precision is evaluated utilizing metrics like accuracy, ROC, and F1 score. The outcomes of this study are compared to other research studies in crop prediction using ML.

This chapter addresses the significant concern of dependable and accurate crop prediction by utilizing an ensemble technique that merges five ML algorithms. The research makes a valuable contribution to the area of agriculture and ML by presenting an effective and precise strategy for crop prediction that has the potential to significantly improve crop yields and food production. The results of this study have the potential to guide policymakers, farmers, and stakeholders in making informed decisions and implementing strategies to boost food security and combat the effects of climate change on the agriculture industry.

The remaining sections of this chapter are structured as follows: In Section 2, we present an overview of the existing work in crop prediction utilizing ML. Section 3 outlines the methodology employed in this study, including the dataset, preprocessing, and ensemble technique. In Section 4, we present the findings of the study, including the performance of the individual models and the ensemble model. Finally, in Section 5, the outcomes and concluding remarks of the study are presented.

* 1. **Related works**

This section discusses the related work of the current study. Vazquez-Carillo et al. proposed an ML approach for predicting crop yields using satellite imagery [12]. The authors used a random forest model to predict yields for three different crops: wheat, soybeans, and corn. In this study, the authors used satellite imagery from the European Space Agency's Sentinel-2 satellite, which captures images of the Earth's surface at a spatial resolution of 10 meters. They used data from the growing season of each crop, which is defined as the period from planting to harvest. The satellite imagery was processed to extract various features related to vegetation indices, land cover, and climate variables. The authors trained a random forest model on this data to predict the yield for each crop. They compared their results to those of traditional statistical models commonly used for crop yield prediction, such as multiple linear regression and principal component regression [13]. The experimental results exhibited that the random forest model outperformed these conventional models, achieving a mean absolute error of 0.14, 0.11, and 0.18 for wheat, soybeans, and corn, respectively. The study also analyzed the importance of different features in predicting crop yields. The authors found that vegetation indices, such as the normalized difference vegetation index (NDVI), were the most important features for predicting yields. The study showed that the use of satellite imagery and machine learning can provide accurate and timely predictions of crop yields, which can help farmers and policymakers make more informed decisions related to crop management and food security. Overall, this study demonstrates the potential of machine learning and satellite imagery for crop yield prediction. By using such techniques, farmers and policymakers can gain vital knowledge on agricultural productivity and be better equipped to manage crops and ensure food security.

Zhang et al. in their work, provided an overview of recent studies on crop yield prediction using machine learning [14]. The authors analyzed various machine learning algorithms used in crop yield prediction, including decision trees, random forests, support vector machines, artificial neural networks, and deep learning [15]. They investigated the advantages and limitations of each approach and highlighted the need for more research to identify the most effective algorithms for different crops and regions [16]. The article also considered the data sources used in crop yield prediction, such as weather data, soil data, and satellite imagery [17]. The authors emphasized the importance of high-quality data and data integration in achieving accurate yield predictions. Additionally, the article discussed the performance metrics used in crop yield recommendation, such as mean absolute error, root means squared error, and coefficient of determination [18]. The authors highlighted the need for standardized evaluation metrics to compare the performance of different models.

The review article also discussed the applications of crop yield prediction using machine learning, such as precision agriculture, crop insurance, and food security [19]. The authors emphasized the potential of machine learning for improving agricultural practices and increasing crop yields [20]. Overall, the article provides a comprehensive overview of recent studies on crop yield prediction using machine learning [21]. The article highlights the potential of machine learning for improving crop yield prediction and the need for more research to identify the most effective approaches for different crops and regions.

Singh et al. in their work presented a detailed and comprehensive review of recent research on crop yield prediction using machine learning techniques [9]. The authors begin by highlighting the significance of crop yield prediction in agriculture and the potential benefits of using machine learning for this purpose. They then provide an overview of the different types of machine learning algorithms commonly used for crop yield prediction, including supervised and unsupervised learning methods [22]. The authors also discussed the different types of data sources used in crop yield prediction research, such as climate data, soil data, and remote sensing data, and how these data sources can be integrated to improve prediction accuracy. They further explore the evaluation metrics used in crop yield prediction studies and provide critical analysis of their strengths and weaknesses. The authors highlighted the importance of developing standardized datasets and the need for more transparency and interpretability in machine learning models to facilitate their adoption in practical applications. They also discuss the challenges associated with collecting high-quality data in agricultural settings and suggest potential solutions, such as the use of crowdsourcing and citizen science approaches [23]. For scholars and practitioners interested in crop yield prediction using machine learning approaches, this review paper is a helpful resource. It highlighted the potential benefits and challenges associated with using ML in smart agriculture and identified the important areas for future research and development. Compared to the above works the proposed work is different in the classification task which combines the best-performing machine learning classifiers to predict the type of the crop.

* 1. **Proposed Methodology**

Figure 1 depicts the block diagram of the proposed crop prediction system. The modules are discussed as follows:

* + 1. **Data Collection**

The dataset is constructed by taking Crop Production Dataset which is taken from the well-known data repository, Kaggle [11]. This dataset was created by enhancing rainfall, climate, and fertilizer data sets that were previously accessible to India. With over 2000 scenarios collected as a text file along with 7 special features (N, P, K, temperature, humidity, pH, and rainfall) and the final column ‘label’ selected as the output, with 22 classifications of crops ('rice', 'maize', 'chickpea', 'kidneybeans', 'pigeonpeas', 'mothbeans', 'mungbean', 'blackgram', 'lentil', 'pomegranate', 'banana', 'mango', 'grapes', 'watermelon', 'muskmelon', 'apple', 'orange', 'papaya', 'coconut', 'cotton', 'jute' and 'coffee').

**1.3.2 Data Processing**

The data input into the computer undergoes certain machine learning algorithms to make it more usable i.e., making it more meaningful and informative. We use such algorithms which have been talked about below.

There is a correlation between two variables when they are closely related to each other. The two primary types of correlations are positive and negative correlations. In correlation, when one variable increases, the other increases as well; when one changes direction, the other changes as well. A negative correlation occurs when two variables change in opposing directions, for instance, when one increases while the other decreases. Depending on the correlation coefficient, the value can range from -1 to 1. The correlation coefficient is computed as follows:

(1)

The correlation formula allows for the calculation of the correlation coefficient between two variables, px, and py. The mean values of px and py are represented by and , respectively. Different values of px and py are represented by pxi and pyi.

Scatter plots depict the correlation between two variables or the extent to which one variable impacts the other, using dots in a two-dimensional plane. They share similarities with line graphs as they also display data points along the horizontal and vertical axes. In this graph, we set the ‘N’ feature as our x-axis and ‘P’ as the y-axis and we see the N and P values at which our different crops grow.

Pair plots produce a grid of scatter plots and histograms that display the pairwise correlations between various variables in a dataset. Using a pair plot can help us see how each variable in a dataset with numerous numeric variables links to each other. The histograms on the diagonal display the distribution of each variable, and each scatter plot in the grid depicts the relationship between two variables.

We display and contrast each feature's distribution, shape, and core tendency from our dataset. These graphics aid in finding trends, outliers, and potential problems in the data.

We use kernel density estimation (KDE) to produce a histogram. The distribution and density of the feature's values are revealed by this visualisation.

A box plot with a rotating kernel density plot on each side make up a violin plot. It offers details on the spread, central tendency, and shape of the distribution.

The distribution's quartiles, median, and any probable outliers are shown in a box plot. It is a helpful tool for locating skewness and outliers in the data.

**1.3.3 Data Reduction**

Sometimes the collected data will have null or empty values which needs to be cleaned or we need to delete redundant features. Using the feature selection technique, we select the data features that have the greatest impact on the target variable. To put it another way, we pick the top predictors for the desired outcome. The features are chosen using the SelectKBest method based on the k highest score. Upon checking our dataset, we find that it had no null value; and using SelectKBest we see that all 7 features are useful thus we need to use them all.

Principal Component Analysis (PCA) is a dimensionality reduction technique that's often used in machine learning and data analysis to transform high-dimensional data into a lower-dimensional representation while preserving as much of the original variance as possible.

Principal Component Analysis with 3 components (PCA-3) and k-best feature selection offers distinct advantages in dimensionality reduction. PCA-3 transforms high-dimensional data into a 3D space by retaining the most informative features, effectively reducing complexity while preserving variance. It eliminates correlations between features and enables meaningful data visualization, aiding in identifying patterns and relationships that might be obscured in higher dimensions. However, interpreting the significance of each component can be challenging, and the resulting model may lack straightforward feature-to-outcome explanations.

On the other hand, k-best feature selection simplifies models by retaining a subset of the most relevant features, enhancing interpretability. By focusing on select features, it reduces computational complexity and can prevent overfitting. Yet, this approach may ignore interactions and dependencies among features, potentially leading to information loss if crucial relationships are excluded. In contrast to PCA-3, k-best selection often results in a model where the relationship between features and outcome is more transparent.

**1.3.4 Data Splitting**

The data is divided into two subsets, mainly the training and the testing. Training is used to train the model and testing is used to evaluate the data. Splitting ensures that the data models created using the training data are accurate, which is checked with the help of the testing data. We split the data in a ratio of 7:3 (training: testing). We use random\_state to ensure the output after splitting the data is the same every time.

**1.3.5 Classification Techniques**

The library "lazypredict" is utilized to simplify the process of building basic models with minimal code, and to assist in identifying the most effective models without requiring parameter tuning. So, using it we found the top 5 classification models based on which we develop the rest of the model.

* + - 1. **Gaussian Naïve Bayes**

The Bayes theorem-based probabilistic machine learning method known as naive Bayes is commonly used in a variety of classification applications. In real-world circumstances, there may be several X variables, hence the Bayes rule provides a formula for determining the probability of Y given the condition X, which is calculated as follows:

(2)

Naive Bayes can be applied when the features are independent of one another, meaning that each X variable stands alone. This technique is more effective than the Bayes formula.

When X is a continuous variable, we can use the probability density function of a distribution to determine the likelihood, if X follows a specific distribution. If X has a Gaussian or normal distribution, the probability density function can be replaced with the normal distribution's probability density, referred to as Gaussian Naive Bayes. To calculate this formula, the mean and variance of X for a given class c of Y must be determined, denoted by sigma and µ, respectively. By replacing the parameters with a new input value for the variable, the Gaussian probability density function can be utilized to create predictions. Thus, the Gaussian function estimates the likelihood of the new input value.

* + - 1. **Decision Tree**

The decision tree is a well-liked and effective method for categorizing and predicting data. Each internal node denotes a test on a feature, each branch denotes a test result, and each terminal node also called the leaf node denotes a class label. It adopts a tree-like form that mimics a flowchart.

To classify an instance using the decision tree, the procedure begins at the root node of the tree, evaluating the attribute specified by the node, and moving down the tree's branch that matches the attribute's value. Then, the subtree rooted at the new node undergoes the same process again. The choice of attributes is the main difficulty in creating a DT. That is, it is challenging to choose which qualities to employ as root nodes or internal nodes given the abundance of available attributes. Information Gain and Gini Index are two methods that can be used to this end:

(3)

where T refers to the current state and X to the selected attribute;

(4)

where p+ represents the probability of Yes/Good and p- the probability of No/Bad.

* + - 1. **Random Forest**

To improvise the model’s classification accuracy, the Random Forest classifier uses numerous decision trees that have been trained on different subsets of the input data. Instead of relying just on one decision tree, the Random Forest uses forecasts from all of the trees and bases its prediction on the predictions that received the most votes.

Because Random Forest is straightforward to grasp, it is very well-liked. Classification and regression can also be carried out with it. In addition, the model's accuracy is greatly improved, and the overfitting problem is solved, even for huge, highly dimensional data sets. Data extrapolation cannot be done using Random Forest since the findings could be unreliable. Although Random Forest may be used for both classification and regression problems, classification jobs are where it shines. Moreover, it fails to deliver accurate findings when working with sparse data and it takes more resources, larger data, and more time for implementation.

* + - 1. **K-Nearest Neighbour (KNN)**

In supervised learning, the K-nearest neighbor algorithm is a method for making predictions about the grouping of data points based on proximity. Due to its assumption that similar points can be found near one another, it is typically used as a classification algorithm. It is important to note that the KNN technique only stores a training dataset and does not go through a training phase, placing it in the category of "lazy learning" models. Additionally, this suggests that all calculations are completed at the same time as a prediction or categorization. It is also known as an instance-based or memory-based learning strategy because it significantly relies on memory to maintain all of its training data.

To identify the data points that are nearest to a particular query point, it is necessary to compute the dissimilarity between the query point and the other data points. The commonly used distance metric is Euclidean distance, which applies only to real-valued vectors. The straight-line distance using the Euclidean distance is determined between the query location and the point being measured.

K-NN techniques determine classification by examining neighboring points to determine the number of neighbors that should be examined. The closest neighbor instance will be assigned to the same class if k = 1. To prevent overfitting or underfitting, it is necessary to consider multiple values of k when setting it. The higher the value of k, the lower the variance, while the smaller the value of k, the higher the variance, but the lower the bias. High values of k may be better for data with more outliers or noise, thus influencing the choice of k. The optimal value for k is generally determined by cross-validation techniques and odd values of k.

* + - 1. **Logistic Regression**

The initial step involves scaling the dataset utilizing the StandardScaler, which performs standardization. Standardization helps to ensure that the individual features are roughly normally distributed with a mean of zero and a variance of one, which is a typical requirement for many machine learning estimators. The output is predicted using logistic regression in the case of a categorical dependent variable. The output in this case must be discrete or categorical. In contrast to the precise values of 0 and 1, yes or no, true or false, etc., it provides probabilistic values between 0 and 1. When calculating the probability in logistic regression, the logistic or sigmoid function is used. The logistic function is a straightforward S-shaped curve that converts input into a value between 0 and 1.

**1.4 Result and Discussions**

This section discusses the experimental results obtained using the proposed model. Firstly, the relationship between the features is visualized using the correlation coefficient, scatter plot, and pair plot that is depicted in Figure 2, Figure 3, and Figure 4 respectively. We also visualise each feature using histogram, violin plot, and box plot. Then the feature importance has been plotted and depicted in Figure 6. Table 1 tabulates the performance of the classifiers after executing the lazypredict function. From these results, the top-performing classifiers have been selected to build the ensemble classifier. Figure 8 depicts the top five performing classifiers.

* + 1. **Voting Ensemble Classifier**

A voting classifier is a machine learning estimator that combines multiple base models or estimators to make predictions by averaging their results. The predictions of each estimator can be aggregated using different voting criteria. There are two types of voting criteria:

* Hard voting, also known as majority voting, is a simple technique where the final prediction is made by selecting the most common prediction among the individual models. For example, if there are three models and two of them predict the class "A," while the third predicts the class "B," the hard voting approach would choose a class "A" as the final prediction. Hard voting is most effective when the individual models are diverse and make independent errors. In other words, if the models are making the same mistakes, hard voting may not improve the overall performance.
* Soft voting, on the other hand, is a more sophisticated technique where the final prediction is based on the probability estimates of each model. Each model assigns a probability to each class, and the final prediction is made by taking the average of these probabilities across all models. Soft voting is more robust than hard voting when individual models have high accuracy but low confidence. By considering the probability estimates, soft voting can make more informed decisions and provide more accurate predictions.

We made four voting ensemble classifiers:

1. In the first model, we combined Decision Tree, Logistic Regression, Random Forest, Gaussian Naïve Bayes, and K-Nearest Neighbour classification using the soft voting method and achieved an accuracy of 100%.
2. In the second model, we combined Decision Tree, Gaussian Naïve Byes, and Random Forest classification using the soft voting method and achieved an accuracy of 100%.
3. In the third model, we combined Decision Tree, Logistic Regression, Random Forest, Gaussian Naïve Bayes, and K-Nearest Neighbour classification using the hard voting method and achieved an accuracy of 70.7%.
4. In the fourth model, we combined Decision Tree, Gaussian Naïve Byes, and Random Forest classification using the hard voting method and achieved an accuracy of 100%.

**1.4.2 Multiclass ROC Curve**

ROC curves show the effectiveness of binary classification models graphically. A one-vs-all method can be used to assess the effectiveness of a multiclass classification model by extending the ROC curve. Figure 9 depicts the ROC plots of the proposed approach. In a multiclass classification problem, there are multiple classes, and each class can be treated as a binary classification problem, where one class is positive, and the rest of the classes are negative classes. To generate the ROC curve, the true positive rate (sensitivity) is plotted along with the false positive rate (1-specificity) for each class. The multiclass ROC curve can be used to compare the performance of different multiclass classification models, by selecting an appropriate threshold for each class, and identifying which classes are harder to predict.

**1.5 Conclusion**

In conclusion, the chapter presented an efficient and accurate approach for crop prediction using machine learning techniques. By utilizing ensemble techniques, two models were developed which were able to predict the correct crop with 100% accuracy. This approach has the potential to significantly improve the crop yield and economic output of the agriculture industry. Additionally, the research demonstrates the power and potential of machine learning in agriculture, particularly in crop prediction. Further research can be conducted to expand the dataset and explore the use of other machine-learning techniques for crop prediction. Overall, this research paper contributes to the field of agriculture and machine learning and provides valuable insights for future developments in this area.

**References**

1. Fenu, G., & Malloci, F. M. (2019, November). An application of machine learning technique in forecasting crop disease. In *Proceedings of the 3rd International Conference on Big Data Research* (pp. 76-82).
2. Pant, J., Pant, R. P., Singh, M. K., Singh, D. P., & Pant, H. (2021). Analysis of agricultural crop yield prediction using statistical techniques of machine learning. *Materials Today: Proceedings*, *46*, 10922-10926.
3. Kalimuthu, M., Vaishnavi, P., & Kishore, M. (2020, August). Crop prediction using machine learning. In *2020 third international conference on smart systems and inventive technology (ICSSIT)* (pp. 926-932). IEEE.
4. Van Klompenburg, T., Kassahun, A., & Catal, C. (2020). Crop yield prediction using machine learning: A systematic literature review. *Computers and Electronics in Agriculture*, *177*, 105709.
5. Nischitha, K., Vishwakarma, D., Ashwini, M. N., & Manjuraju, M. R. (2020). Crop prediction using machine learning approaches. *International Journal of Engineering Research & Technology (IJERT)*, *9*(08), 23-26.
6. Gupta, A., Nagda, D., Nikhare, P., & Sandbhor, A. (2021). Smart crop prediction using IoT and machine learning. *International Journal of Engineering Research & Technology (IJERT)*, 2278-0181.
7. Kumar, R., Singh, M. P., Kumar, P., & Singh, J. P. (2015, May). Crop Selection Method to maximize crop yield rate using machine learning technique. In *2015 international conference on smart technologies and management for computing, communication, controls, energy and materials (ICSTM)* (pp. 138-145). IEEE.
8. Rajak, R. K., Pawar, A., Pendke, M., Shinde, P., Rathod, S., & Devare, A. (2017). Crop recommendation system to maximize crop yield using machine learning technique. *International Research Journal of Engineering and Technology*, *4*(12), 950-953.
9. Dharani, M. K., Thamilselvan, R., Natesan, P., Kalaivaani, P. C. D., & Santhoshkumar, S. (2021, February). Review on crop prediction using deep learning techniques. In *Journal of Physics: Conference Series* (Vol. 1767, No. 1, p. 012026). IOP Publishing.
10. Reddy, D. J., & Kumar, M. R. (2021, May). Crop yield prediction using machine learning algorithm. In *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 1466-1470). IEEE.
11. Crop prediction dataset, <https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset>, Accessed 11, January 2023.
12. Vazquez-Carillo(2020) Crop Yield Prediction using Machine Learning and Satellite Imagery.
13. Kim, Y. H., Yoo, S. J., Gu, Y. H., Lim, J. H., Han, D., & Baik, S. W. (2014). Crop pests prediction method using regression and machine learning technology: Survey. *IERI procedia*, *6*, 52-56.
14. Zhang, C., Marzougui, A., & Sankaran, S. (2020). High-resolution satellite imagery applications in crop phenotyping: an overview. *Computers and Electronics in Agriculture*, *175*, 105584.
15. Agarwal, S., & Tarar, S. (2021). A hybrid approach for crop yield prediction using machine learning and deep learning algorithms. In *Journal of Physics: Conference Series* (Vol. 1714, No. 1, p. 012012). IOP Publishing.
16. Champaneri, M., Chachpara, D., Chandvidkar, C., & Rathod, M. (2016). Crop yield prediction using machine learning. *Technology*, *9*, 38.
17. Kumar, Y. J. N., Spandana, V., Vaishnavi, V. S., Neha, K., & Devi, V. G. R. R. (2020, June). Supervised machine learning approach for crop yield prediction in agriculture sector. In *2020 5th International Conference on Communication and Electronics Systems (ICCES)* (pp. 736-741). IEEE.
18. Abbas, F., Afzaal, H., Farooque, A. A., & Tang, S. (2020). Crop yield prediction through proximal sensing and machine learning algorithms. *Agronomy*, *10*(7), 1046.
19. Palanivel, K., & Surianarayanan, C. (2019). An approach for prediction of crop yield using machine learning and big data techniques. *International Journal of Computer Engineering and Technology*, *10*(3), 110-118.
20. Ghadge, R., Kulkarni, J., More, P., Nene, S., & Priya, R. L. (2018). Prediction of crop yield using machine learning. *Int. Res. J. Eng. Technol.(IRJET)*, *5*.
21. Muruganantham, P., Wibowo, S., Grandhi, S., Samrat, N. H., & Islam, N. (2022). A systematic literature review on crop yield prediction with deep learning and remote sensing. *Remote Sensing*, *14*(9), 1990.
22. Rao, M. S., Singh, A., Reddy, N. S., & Acharya, D. U. (2022). Crop prediction using machine learning. In *Journal of Physics: Conference Series* (Vol. 2161, No. 1, p. 012033). IOP Publishing.
23. Abbas, F., Afzaal, H., Farooque, A. A., & Tang, S. (2020). Crop yield prediction through proximal sensing and machine learning algorithms. *Agronomy*, *10*(7), 1046.

**Figure Legend**

Figure 1: Block diagram of the proposed approach

A picture containing text, screenshot, diagram, parallel

Description automatically generated

Figure 2: Relation Between the 7 features using Correlation Heatmap

A screenshot of a graph

Description automatically generated with medium confidence

Figure 3: Scatter Plot between N and P

A picture containing text, screenshot, colorfulness

Description automatically generated

Figure 4: Pair Plot between all the features

A screenshot of a computer generated image

Description automatically generated

Figure 5: Visualisation of all the features using a histogram, violin plot, and a box plot

**A green and black line

Description automatically generated**

**A green and black graph

Description automatically generated**

**A green and black graph

Description automatically generated**

**A graph of a temperature

Description automatically generated with medium confidence**

**A green and blue graph

Description automatically generated**

**A green and black graph

Description automatically generated**

**A green and blue graph

Description automatically generated**

Figure 6: Importance of Features

A picture containing text, screenshot, display, software

Description automatically generated

Figure 7: Principal Component Analysis

**A graph on a grid

Description automatically generated**

Figure 8: Bar Plot for top 5 classification

A picture containing text, screenshot, rectangle, colorfulness

Description automatically generated

Figure 9: Multiclass ROC Curve

A picture containing text, screenshot, diagram, number

Description automatically generated

Table 1: Performance of Classifiers using LazyPredict

| **Model** | **Accuracy** | **Balanced Accuracy** | **F1 Score** | **Time Taken** |
| --- | --- | --- | --- | --- |
| GaussianNB | 0.99 | 0.99 | 0.99 | 0.02 |
| RandomForestClassifier | 0.99 | 0.99 | 0.99 | 0.39 |
| QuadraticDiscriminantAnalysis | 0.99 | 0.99 | 0.99 | 0.03 |
| DecisionTreeClassifier | 0.99 | 0.99 | 0.99 | 0.03 |
| BaggingClassifier | 0.99 | 0.99 | 0.99 | 0.08 |
| ExtraTreesClassifier | 0.98 | 0.99 | 0.98 | 0.20 |
| LGBMClassifier | 0.98 | 0.98 | 0.98 | 1.23 |
| LabelPropagation | 0.97 | 0.97 | 0.97 | 0.11 |
| LabelSpreading | 0.97 | 0.97 | 0.97 | 0.18 |
| SVC | 0.97 | 0.97 | 0.97 | 0.19 |
| LinearSVC | 0.96 | 0.96 | 0.96 | 0.11 |
| CalibratedClassifierCV | 0.96 | 0.96 | 0.96 | 0.54 |
| KNeighborsClassifier | 0.96 | 0.96 | 0.96 | 0.03 |
| LogisticRegression | 0.96 | 0.96 | 0.96 | 0.20 |
| LinearDiscriminantAnalysis | 0.95 | 0.95 | 0.95 | 0.03 |
| NuSVC | 0.94 | 0.94 | 0.94 | 0.36 |
| SGDClassifier | 0.88 | 0.89 | 0.89 | 0.09 |
| ExtraTreeClassifier | 0.88 | 0.88 | 0.88 | 0.02 |
| NearestCentroid | 0.86 | 0.86 | 0.85 | 0.02 |
| Perceptron | 0.86 | 0.86 | 0.86 | 0.05 |
| PassiveAggressiveClassifier | 0.81 | 0.81 | 0.81 | 0.05 |
| BernoulliNB | 0.78 | 0.78 | 0.77 | 0.02 |
| RidgeClassifier | 0.64 | 0.66 | 0.57 | 0.02 |
| RidgeClassifierCV | 0.64 | 0.66 | 0.56 | 0.04 |
| AdaBoostClassifier | 0.21 | 0.23 | 0.17 | 0.21 |
| DummyClassifier | 0.03 | 0.05 | 0.00 | 0.02 |