Identifying Rice Species Through Utilization of Machine Learning Algorithms

Zachary Paluck

School of Computer Science & Engineering, California State University San Bernardino

Abstract

Through this paper the efficacy two machine learning methods were be tested against each other through the use of a dataset from the UC Irvine Machine Learning Repository. The dataset involves several measurements of photos of different grains of rice with the goal of identifying the species through these measurements.

Logistic Regression and K Nearest Neighbor are the two algorithms that were used to create models based on the preprocessed rice data, with both having been trained and tested off of the same preprocessed and split subset of the original dataset.

After being trained and tested the results were analyzed and each algorithm's performance was compared against the other with the intent to gauge possible improvements for future experiments along with each algorithm's potential use for similar identification problems.

1. Introduction

Before trying to apply machine learning solutions to the problem it is important that we actually understand the problem at hand and the implications that the problem and solution may have.

1.1. Problem Definition

We are attempting to identify the species of rice based on the properties present within images of grains of rice. The two species we are trying to differentiate between are Osmancik and Cammeo rice grains. These two grains are very similar in appearance yet they have subtle qualities that can be used to identify each grain.

With the aforementioned qualities being organized into a dataset we can process said dataset and apply machine learning algorithms to create models to predict whether a grain of rice is Osmancik or Cammeo. In this case we are using two algorithms to compare them to each other to see which can be most effectively applied to this problem and other like it.

1.2. Problem Significance

Two immediately apparent areas that this problem and other similar problems pertain to are in biological and manufacturing scopes.

As for the manufacturing aspect of this scenario being able to identify rice, another grain, or potentially any small mass produced or processed item can be utilized in identify defective product or possible detritus for which the batch may be separated or marked for further quality assurance.

Biologically the concept of identifying different object based on minute differences in features has sever potential applications. From a genetics standpoint being able to recognize specific features can be used to identify expressed features in a child of a crossbred flora or fauna. In a more medical application the ability to differentiate based on minute photographic differences may have a use in identifying diseased cells or organs.

2. Methodologies

For this experiment the two methods chosen to compare against each other are the logistic regression and the K nearest neighbor algorithms. Both of these were chosen due to their proclivity for classification problems as opposed to problems requiring discrete values as an output.

2.1. Logistic Regression

Logistic regression is a form of regression which as stated before hand is capable of handling classification tasks as opposed to a linear regression which is used for discrete solutions. A logistic regression model works by creating a sigmoid regression line. This line is made through the constant updating of the log likelihood function towards the steepest gradient increase. These updates are based on the partial derivative of the log likelihood function with respect to each theta value which correlates to each feature variable.

Once the sigmoid regression line is created it can then be used for classification. The line ranges from zero to one on the y axis with the scaling of the x axis being based on the fitting of the line described previously. To actually classify a new instance the features of said instance are projected from the other respective axes for each feature's value. The point to which they value is projected upon the line then determines the classification of the object as a binary classification with a y-value of below .5 being classified as one value and above .5 as the other.

2.2. KNN

KNN, K nearest neighbors, is an algorithm, which is also tailored for classification problems such as our rice classification problem. The main crux of KNN is classifying new instances based on its position relative to other classified instances. The K in KNN refers to how many neighbors the new instances classification should be based on.

To classify an instance the K nearest neighbors are checked for their classifications with the classification of the new instance being based on the majority classification present in said k neighbors. Generally K is an odd number as to avoid ties in binary classifications, however situations such as ties can be avoided through modifications of the KNN model. The way that distance is calculated is one way that a KNN model may be modified along with the number of neighbors to check. The weight that each classification holds in classifying new instances may also be modified to improve results.

3. Dataset and Experiment

3.1. Dataset

Sourced from the UC Irvine Machine Learning Repository, the Rice (Cammeo and Osmancik) dataset is a collection of data garnered from photos of grains of rice that are either the species Cammeo or Osmancik.

The dataset contains 3810 instances with features based off of information garnered from photos of grains of rice with each instance representing one photo. Each instance has seven features all of which are integer or continuous values which are measurements of the grain of rice in each photo. The first two features are the area and perimeter of the grain of rice measured in pixels. Next are the major and minor axis lengths which are the longest and shortest lines that can be drawn on the rice respectively. The final three features are the eccentricity measures the roundness of a ellipse that follows the curves of the rice, the convex area measures the smallest convex shell formed by the rice, and the extent is the ratio of the rice compared to the bounds of the rice in the image. Along with these features the target variable is a binary value representing the rice species as either Osmancik or Cammeo.

As for the structure of the dataset there are no missing values with all values having properly formatted values. One thing to note is that the dataset is somewhat unbalanced with 2180 instances being classified as Osmancik and 1630 being classified as Cammeo. This comes out to a split of about 57% Osmancik and 43% Cammeo.

3.2. Experiment

The creation of the two machine learning models was done within google colab while utilizing pandas for dataset manipulation and sklearn for data transformation, machine learning model fitting, and analysis of the resulting models.

To start the experiment the data was imported as two pandas dataframes using the ucimlrepo library from the UCI Machine Learning Repository. These two dataframes are the seven feature columns labeled as X and the one target column labeled as y. With the data accessible it was then time for preprocessing.

3.2.1. Preprocessing

As the dataset is missing no values and all columns are wanted the next step for preprocessing was to normalize the features dataframe which was done using the sklearn MinMaxScaler function from the preprocessing library. We normalized the data so that the KNN can be properly implemented with the additional benefit of making the next steps of preprocessing for the logistic regression easier with less varied values making the process less complex. The normalized data was then split into a training and test split of .8 to .2 meaning that 80% of the data is used for training with 20% being left for testing. It is also important to note that the same test and training split was used to test both models. With the data split and ready for training next came model selection.

3.2.2. Model Selection

For the logistic regression, polynomial features were used to create a better fitting model. to choose the polynomial degree cross validation was employed to find the polynomial degree which resulted in the highest average accuracy. Degrees from two to ten were tested in a seven fold cross validation with ten being the highest degree due to hardware limitation from colab. After the rounds of cross validation were completed the model with a polynomial degree of ten performed the best and as such was the selected value for the final model. The model was made using the default parameters of the LogisticRegression function with the polynomial generated from the X training set being used for the features and the y training set being used for the target values.

As for the K nearest neighbors model, the default parameters aside from the number of neighbors was used along with the unchanged training features and targets. The number of neighbors was selected with another seven fold cross validation with a K value of seventeen giving the highest average cross validation score. As such, seventeen was the number of neighbors used in the final model.

4. Evaluation and Comparison

With preprocessing and model selection completed the fitted models could be tested using the determined best parameters and transformed features.

4.1. Evaluation

Both of the models were tested using the same 20% test split as previously mentioned. For the logistic regression the test features, X_test, were transformed using the same ten degree polynomial feature transformation that the training set was created with. The KNN model did not have any such requirement and as such was tested using an X_test feature dataframe that went unmodified from its creation.

To test the models each created model was used to predict the target rice species value for the given test set using the sklearn predict method with the aforementioned respective test features. Then with the predictions made they were compared to the real results given by the y_test split made at the same time as the X_test dataframe. For the evaluation of both models this y_test dataframe remained unchanged as no changes were necessary. From the predicted values and the y_test values the accuracy score was found using the sklearn accuracy_score function from the sklearn metrics library.

4.1.1. Logistic Regression Evaluation

For the training and test split used in this paper, the logistic regression had an accuracy score of about 94.0%. Using sklearn confusion matrices also from the metrics library we can see where and how the model erred.

Figure 1. Shows the evaluation results of the logistic regression with 0 representing Cammeo rice and 1 representing Osmancik rice. We can see from the matrix that for the test set with 762 instances 308 of those instances were correctly identified as Cammeo rice and 408 correctly labeled as Osmancik. The two other squares represent the two false categorization with 19 instances being Osmancik yet labeled as Cammeo and 27 instances were identified as Osmancik when they were actually Cammeo. From this we can see that this model is slighlty more adept at identifying Osmancik than with Cammeo with the Osmancik's recall and f1-score both being a few percent more accurate than the Cammeo's.

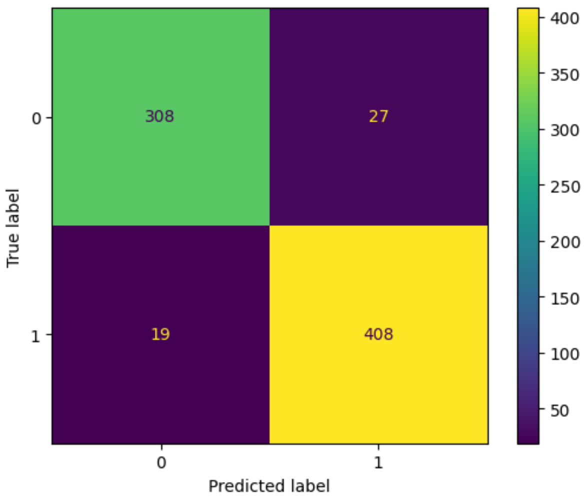

Figure 1. Logistic Regression Confusion Matrix

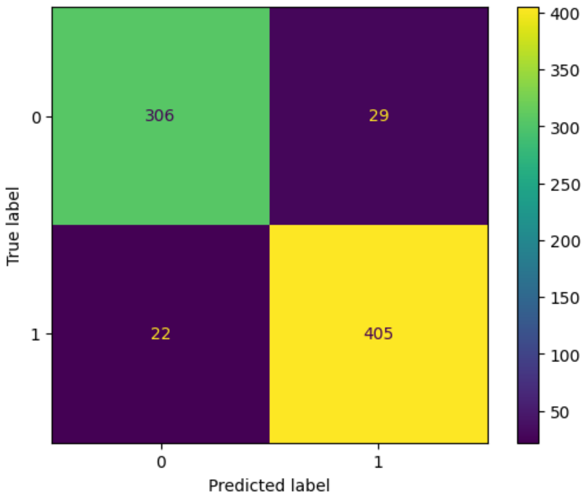

Figure 2. KNN Confusion Matrix

4.1.2. KNN Evaluation

The KNN model when tested in the same manner as the logistic regression model returned an accuracy score of 93.3%. Once again a confusion matrix was generated to represent the classifications made in this test.

Figure 2. shows the labeling distribution of the KNN test. Similar to Figure 1., the 0 also represents Cammeo rice and the 1 represents Osmancik rice. As such, 306 insatances were correctly identified as Cammeo with 22 being incorrectly identified as Cammeo rice. The Osmancik rice saw 405 correct instance labels with 29 instances being incorrectly labeled as Osmancik. The Osmancik saw better prediction accuracy with recall and f1-scores a few percent above the Cammeo rice's scores.

4.2. Comparison

Both models performed well with scores well above 90%. The logistic regression model consistently outperforms the KNN model with the given experimental setup. Despite this the gap between the models is not large with the precision, recall, f1-score, and the overall accuracy only varying by about a percent.

One interesting quirk of both models is that both had the same skewed incorrect predictions with there being less false positives for Cammeo than Osmancik. Another consistent aspect of this is that despite the Osmancik always having more false positives the recall and precision for both models have the Osmancik label performing better. I suspect that this pattern is caused by the somewhat unbalanced nature of the dataset with the models forming a slight bias for labeling instances as Osmancik.

5. Conclusion

Altogether, the performance of the two algorithms was remarkably similar. Both algorithms seemed to struggle in a similar way possibly at the fault of the dataset's balance. Despite some struggle the accuracy of both maintained relatively high throughout the testing of different splits with the models maintaining accuracies comfortably above 90%. These results show that these two algorithms have a good prospect of having pivotal applications in biological and manufacturing fields for situations similar to these. In addition the application of these algorithms are not limited to these fields and can be useful in just about any problem wherein the goal is to distinguish one object from another. The success here with the limited experimental scope implies that further implementations may see vast improvements.

5.1. Future Work

It was mentioned in the conclusion section that the scope of this experiment was somewhat limited and this expressed itself a few ways. The first most apparent one is that fact that for the purpose of this experiment only a couple parameters and other such potential modifications were applied. In future works the full modifiable extent of each model could be explored and further custom fit to the problem at hand. To expand on this further the models used were all from the sklearn libraries which means that the exploration of these algorithms is limited to the scope of how these libraries functions work. There are a multitude of other libraries and there is also the potential for custom implementations of these algorithms to achieve the users desired results.

One other immediate limitation that could be improved in the future was the implementation of the polynomial features for the logistic regression model. Through colab the highest degree that was able to be achieve without crashing due to hardware limitations was a degree of ten which was also the best performing. It is difficult to say how much improvement could be seen through a higher degree polynomial transformation as the improvements from degrees lower than ten to ten were already minuscule but it is uncertain if that trend would continue. Nevertheless it is a potential improvement.

References

[1] Rice (Cammeo and Osmancik). UCI Machine Learning Repository, 2019. DOI: https://doi.org/10.24432/C5MW4Z.

[2] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 2011;12:2825–2830.

[3] Buitinck L, Louppe G, Blondel M, Pedregosa F, Mueller A, Grisel O, Niculae V, Prettenhofer P, Gramfort A, Grobler J, Layton R, VanderPlas J, Joly A, Holt B, Varoquaux G. API design for machine learning software: experiences from the scikit-learn project. In ECML PKDD Workshop: Languages for Data Mining and Machine Learning. 2013; 108–122.