

POKER HAND CLASSIFICATION

Group Name: Poker Kings
Rakesh Nemu and Sameer Srivastava
rakeshnemu237@gmail.com, sameer.srivastava2013@gmail.com

May 1, 2024

Abstract

Poker hand classification poses a challenging machine learning task due to its imbalanced nature, with rare hand types like the Royal Flush leading to poor accuracy. This project addresses this imbalance using feature engineering. We transformed the given UCI[3] dataset into a new dataset and evaluated various classification algorithms (KNN, Decision Tree, Random Forest, Naive Bayes, and AdaBoost) on the transformed dataset to improve classification accuracy. Results demonstrate significant accuracy enhancement. A precise classification model benefits novice players by easily identifying the ten possible Poker hands and also strength of the corresponding poker hand.

1 Introduction

Poker is probably the world's most popular card game at the moment. It's not a very complicated game. Each play can make hands using 5 cards, 2 of which are in your hand and three on the table. There are different kinds of hands that one player can have and each of the hands have a ranking based on Poker Rules. Poker hand classification presents a challenging machine learning problem, aiming to categorize poker hands into 10 predefined classes based on the cards' suits and ranks. The dataset is imbalanced because certain classes are statistically rarer than others for example the Royal Flush, being significantly less likely to occur, with only 480 possible combinations, compared to others, such as straight types, which have 1,224,000 possible combinations. The distribution of all the classes is given in [figure 1]. This imbalance poses a challenge for machine learning models, as they struggle to accurately classify the minority classes, leading to reduced accuracy rates. Aim is to suggest a feature transformation for the given dataset so that classification accuracy significantly improves for different models. Accurate classification models are useful for novice learners of poker to easily identify the poker hands based on card ranks and suit. Given 10 dimensional Dataset is transformed into 19 dimensional

0: Nothing in hand, 49.95202%
1: One pair, 42.37905%
2: Two pairs, 4.82207%
3: Three of a kind, 2.05118%
4: Straight, 0.37185%
5: Flush, 54 instances, 0.21591%
6: Full house, 36 instances, 0.14394%
7: Four of a kind, 0.02399%
8: Straight flush, 0.01999%
9: Royal flush, 0.01999%

Figure 1: Class Distribution

dataset so that it becomes machine friendly (learning process of classifiers becomes easier) and using the transformed dataset accuracy comparison of different models are done.

2 Literature review

Thomas Angeland et.al.[1] addresses the challenge of class imbalance in machine learning, focusing on multi-class datasets with more than two classes. It highlights the lack of research in handling imbalanced multi-class datasets, where minority classes are often misclassified. The authors propose testing various methods on the Poker Hands dataset to improve classification performance. The text discusses the dataset predictors and classes for a poker hand classification task, detailing the features such as the suit and rank of cards and the corresponding poker hand classes. It also explains the stratified sampling approach used to create training, validation, and testing sets, ensuring representative distribution across different poker hand classes to avoid biased classification results. Additionally, the text mentions data augmentation techniques like SMOTE, which aims to balance the training set by generating synthetic samples for minority classes, potentially improving the classifier's performance without affecting the final conclusions drawn from the testing set. No significant improvement is observed after using the standard imbalance handling methods in fact in certain cases, for eg. in Naive Bayes accuracy (.38) decreased after using SMOTE (0.15) .

Walinton Cambronero[2] describes the methodology used to create classifiers that can classify a 5-cards poker hand entirely based in Machine Learning as opposed to classical rule-based programming. The results obtained for each of the following algorithms are discussed in detail: Multi-layer Perceptron Neural Network, Support Vector Machines, Decision Trees and K-Nearest Neighbors. Additionally, a simple linear transformation from

the original 10D space to a new 17D for the dataset is proposed, which significantly improves the performance of the classifiers. A linear transformation is preferable due to its reduced computational requirements and makes the data machine friendly. Macro F1 score is used to measure accuracy of the model because other metrics where weights are given to those classes with more number of samples can give false impression of high accuracy. F1 score treats all the classes unweighted to handle this problem.

Suraiya Jabin[4] aims to solve the poker hand classification problem using artificial neural networks (ANNs) with supervised (backpropagation) and unsupervised (self-organizing map or SOM) learning paradigms. The categorical poker dataset from UCI was encoded into numeric input vectors representing the suits and ranks of 5-card hands, with target outputs encoding the 8 poker hand ranks from 2 (two pairs) to 9 (royal flush). Multiple feedforward backpropagation architectures (e.g. 85-20-20-20-8 neurons) were tested with different training functions like trainlm, trainscg, and trainrp. The SOM used competitive learning on the input vectors to cluster the hands into the 8 ranks. Results showed single hidden layer backpropagation networks performed poorly, requiring at least 3 hidden layers with >10 neurons each. The best backpropagation model achieved 80-95% regression on train/validation/test sets. The SOM with a [1x8] map topology achieved slightly better accuracy of 100% training, 90% validation, 85% test. Both neural network approaches outperformed existing minimax game theory methods, with the SOM converging faster. The paper demonstrates ANNs can successfully solve this difficult non-numeric categorical data classification problem, with a best overall accuracy of 94% using the SOM model.

3 Proposed methodology

To handle the problem discussed in Introduction part. A four steps methodology is proposed:

1. Transforming the Dataset
2. Training different classification models on dataset containing 10 lacs sample points.
3. Testing the model accuracy on 25000 sample points provided in UCI[3] dataset and testing the best model on 26 lacs exhaustive sample dataset created by us.
4. Comparing accuracy among used models and also with state of the art result.

3.1 Transformation of the dataset:

Transformation of the dataset is done to make dataset more machine friendly, The original dataset model gives an artificial importance to the order in which the cards appear (samples are ordered lists of 5 cards) and it does not explicitly encode the cardinality of each suite or rank. Original Dataset as described under experimental result section is transformed from 10 dimensions to 19 dimensions. Transformation is done in two stages, at first the original 10 dimensional dataset is transformed into 17 dimensions as suggested

by Walinton Cambronero[2].

Transformation process is given below:

Attributes 1 through 13: The 13 ranks, i.e. 1: Ace, 2: Two, 3: Three, ..., 10: Ten, 11: Jack, 12: Queen, 13: King. Attributes 14 through 17: The 4 suites, i.e. 14: Hearts, 15: Spades, 16: Diamonds, 17: Clubs

Domain: [0-5]. Each dimension represents the rank or suite cardinality in the hand.

Last dimension: Poker hand [0-9].

Example:

transformation for the Royal Flush of Hearts: Representation in original dimensions (10D)

Data: 1,1,1,10,1,11,1,12,1,13,9

Encodes: Hearts-Ace, Hearts-Ten, Hearts-Jack, Hearts-Queen, Hearts-King, Royal-Flush

Representation in new dimensions (17D)

Data: 1,0,0,0,0,0,0,0,1,1,1,1,5,0,0,0,9

Encodes: 1st column = 1 ace, 10th through 13th columns = 10, Jack, Queen and King, 14th column = 5 cards are hearts, and 18th column a Royal Flush.

After transforming the original dataset into 17 dimension , to further improve accuracy, addition of two more features is purposed to make the final 19 dimensional transformation of the dataset. description of column no 18 and 19 is given as follows:

1.If five consecutive cards are present in a hand which means 5 consecutive ones are present in column 1 to 13 then the value of 18th column is 1 otherwise 0

2.If Ranking positions 0(Ace), 9(jack), 10(jack), 11(Queen), 12(king) has 1 (they are present in hand) and all belong to same suit (any of the feature value between 14 to 17 is 5) then value of 19th column is 1 otherwise 0.

Above given example of Royal Flush of Hearts after this transformation will look like this:

Data: 1,0,0,0,0,0,0,0,1,1,1,1,5,0,0,0,0,1,9

Encodes: 1st column = 1 ace, 10th through 13th columns = 10, Jack, Queen and King, 14th column = 5 cards are hearts, 18th column=0 no consecutive cards are present, 19 column =1 satisfies the condition mentioned above and 20th column= 9 denotes royal flush.

3.2 Models Used for Classification:

A brief description of models used in classification is discussed below:

K-Nearest Neighbors: KNN Classifier is a lazy learning, instance-based algorithm that classifies a new instance based on the majority vote of its K nearest neighbors in the training data. It makes no assumptions about the underlying data distribution and can handle non-linear decision boundaries.

Decision Tree: A Decision Tree is a tree-like model where each internal node represents a test on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label. It learns simple decision rules from the training data to build the

tree in a top-down recursive manner.

Random Forest: Random Forest is an ensemble learning algorithm that constructs multiple decision trees during training and outputs the class that is the mode of the classes predicted by individual trees. It reduces the decision trees' overfitting issues by introducing randomness in the model construction. Random Forests are robust to outliers and noise, can handle high dimensional data well, but are more complex than a single decision tree.

Naive Bayes: Naive Bayes is based on Bayes' theorem and assumes that all features are conditionally independent given the class label. It calculates the probability of each class given the feature values, and outputs the class with the highest probability.

Ada Boost: AdaBoost (Adaptive Boosting) is an ensemble meta-algorithm that combines multiple weak learners (e.g. decision trees) to create a strong learner. It iteratively trains weak learners by focusing more on previously misclassified instances and combining them using weighted majority voting. AdaBoost often achieves very high accuracy, is adaptive to errors, but can be sensitive to noisy data and outliers.

3.3 Metric Used for Accuracy Measurement :

Macro F1 score is used to compare the accuracy among different models and confusion matrix is used to judge the class wise accuracy for each model.

F1 Score: for a single class, it is defined as harmonic mean of the precision and recall for that class.

Precision: It is the fraction of instances predicted as positive that are actually positive,

Recall : It is the fraction of actual positive instances that are correctly predicted as positive.

Macro F1 score is calculated by first computing the F1 score for each class individually, and then taking the unweighted mean of these F1 scores across all classes. The reason for choosing macro F1 score is it gives equal importance to all the classes and does not give weight to those classes which has more number of samples hence if the model correctly classifies only the majority classes overall accuracy remains low.

4 Results

4.1 DATASET DESCRIPTION

The dataset is taken from UCI[3] ML repository. The dataset is divided in training and testing sets. There are around 1M and 25K samples in each set, respectively. This is a 11-dimensional dataset: 10 attributes and 1 label (a.k.a. class or feature). All attributes are categorical. There are no missing values. Each sample represents a 5-cards poker-hand. Each card has two attributes (a.k.a. features): suite and rank.

Encoding

Suite: 1: Hearts, 2: Spades, 3: Diamonds, 4: Clubs

Rank: 1: Ace, 2: 2, ..., 10: Ten, 11: Jack, 12: Queen, 13: King

Label: 0: Nothing, 1: Pair, 2: Two pairs, 3: Three of a kind, 4: Straight, 5: Flush, 6: Full house, 7: Four of a kind 8: Straight Flush 9: Royal Flush

After feature transformation ,we have created a new dataset with 19 feature columns.

Encoding

Attributes 1 through 13: The 13 ranks, i.e. 1: Ace, 2: Two, 3: Three, ..., 10: Ten, 11: Jack, 12: Queen, 13: King. Attributes 14 through 17: The 4 suites, i.e. 14: Hearts, 15: Spades, 16: Diamonds, 17: Clubs

Domain: [0-5]. Each dimension represents the rank or suite cardinality in the hand.

Attribute 18 represents that if,five consecutive cards are present in a hand which means 5 consecutive ones are present in column 1 to 13 then the value of 18th column is 1 otherwise 0.

Attribute 19 represents that if Ranking positions 0(Ace), 9(jack), 10(jack), 11(Queen), 12(king) has 1 (they are present in hand) and all belong to same suit (any of the feature value between 14 to 17 is 5) then value of 19th column is 1 otherwise 0.

Last dimension: Poker hand [0-9]. The dataset is very imbalanced. Class Distribution is mentioned in figure 1.

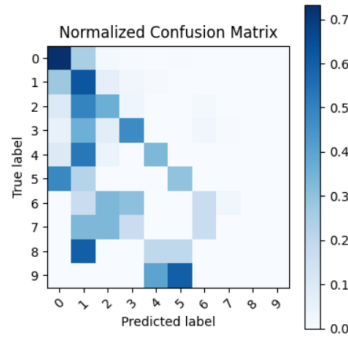
4.2 Experimental Results:

Comparison result of F1 score and Confusion matrix for Base Dataset and Transformed Dataset for different models are described below:

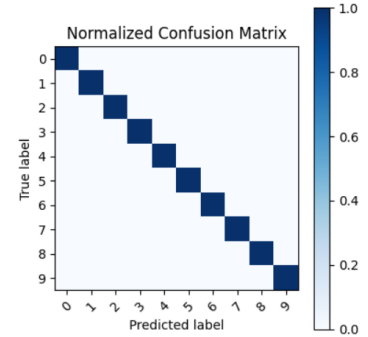
Decision Tree:

Macro F1 score for base dataset is **0.28** and after transformation macro F1 score improves to **1** (All the sample points are correctly classified).

Confusion Matrix for this model is given below:



(a) Confusion Matrix for Base dataset



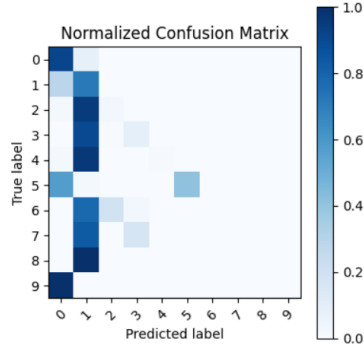
(b) Confusion Matrix for Transformed dataset

Figure 2: Comparison of Confusion Matrices for Decision Tree Model

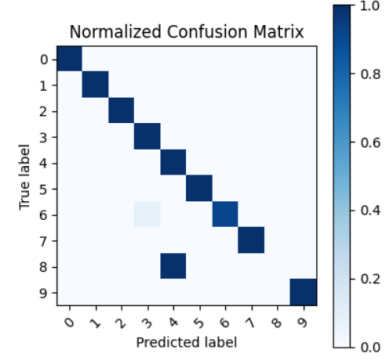
Random Forest:

Macro F1 score for base dataset is **0.24** and after transformation macro F1 score improves to **0.89**.

Confusion Matrix for this model is given below:



(a) Confusion Matrix for Base dataset



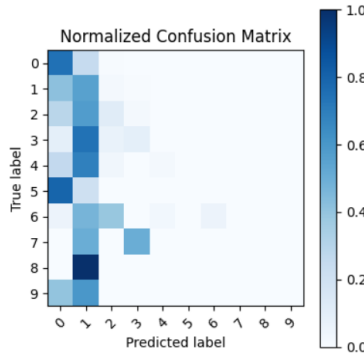
(b) Confusion Matrix for Transformed dataset

Figure 3: Comparison of Confusion Matrices for Random Forest Model

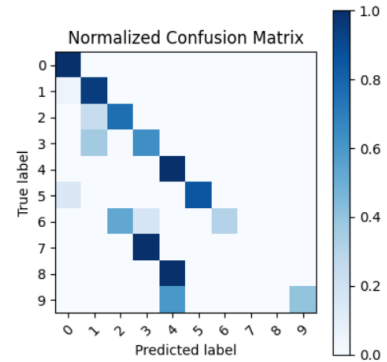
KNN:

Macro F1 score for base dataset is **0.17** and after transformation macro F1 score improves to **0.66**.

Confusion Matrix for this model is given below:



(a) Confusion Matrix for Base dataset



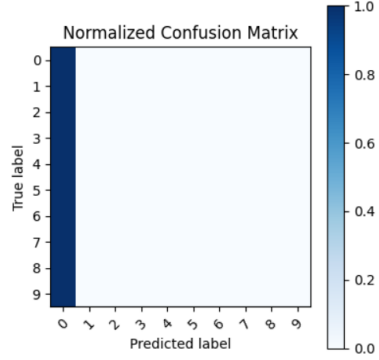
(b) Confusion Matrix for Transformed dataset

Figure 4: Comparison of Confusion Matrices for KNN Model

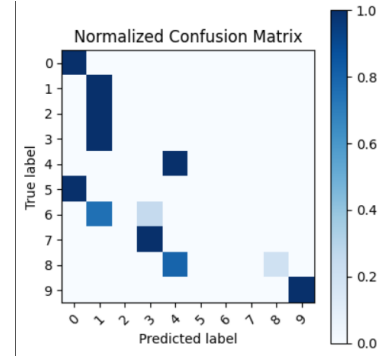
Naive Bayes:

Macro F1 score for base dataset is **0.07** and after transformation macro F1 score improves to **0.42**.

Confusion Matrix for this model is given below:



(a) Confusion Matrix for Base dataset



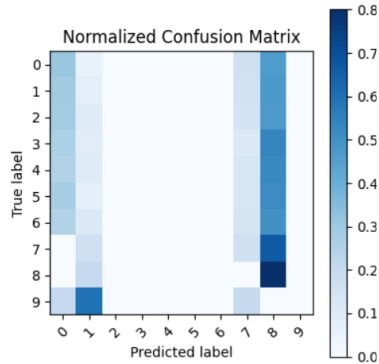
(b) Confusion Matrix for Transformed dataset

Figure 5: Comparison of Confusion Matrices for Naive Bayes Model

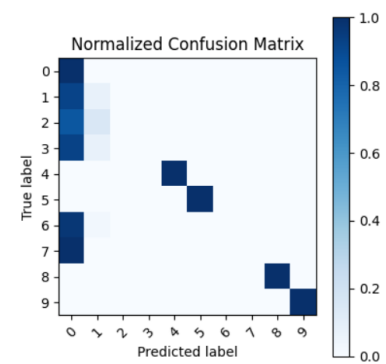
AdaBoost:

Macro F1 score for base dataset is **0.05** and after transformation macro F1 score improves to **0.48**.

Confusion Matrix for this model is given below:



(a) Confusion Matrix for Base dataset

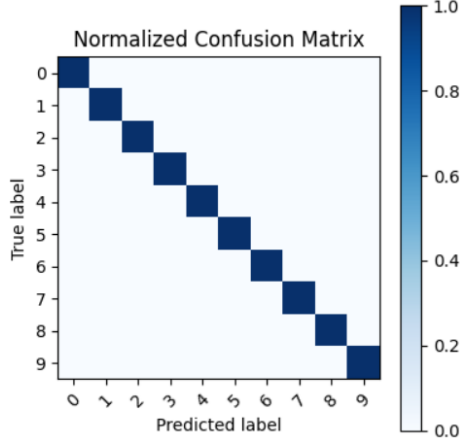


(b) Confusion Matrix for Transformed dataset

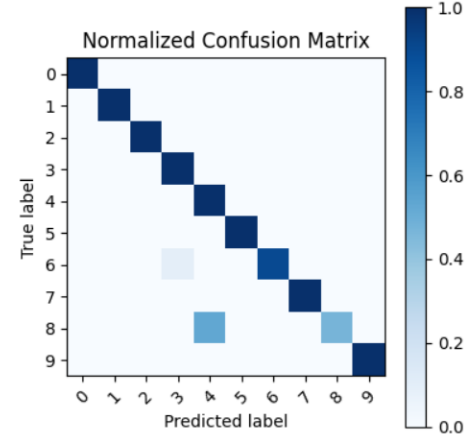
Figure 6: Comparison of Confusion Matrices for AdaBoost Model

Testing Best Models on Exhaustive sample:

Decision Tree and Random forest are the best models. We have tested these models on exhaustive 2.6m sample dataset. For Decision Tree, F1 score is 1 and for Random Forest, F1 score is .94. The confusion Matrix for both models given Below :



(a) Confusion Matrix for Decision Tree



(b) Confusion Matrix for Random Forest

Figure 7: Confusion Matrices for best models

4.3 State of the art comaprison:

The following table shows the comparison of results of different model used by Walinton Cambronerio[2] (17D transformation) and after 19D transformation proposed by us.

Table-1

Models	Accuracy for 17D	Accuracy for 19D
Decision Tree	0.69	1
KNN	0.51	0.66

A significant improvement in accuracy of the models is observed after transforming the feature means adding two more features. Since, Decision tree algorithm is giving the best result to classify the poker hand, we have tested 26 lacs exhaustive sample dataset on that trained model. F1 score for the model is 1 which is actually a good result.

5 Summary

The Poker Hand dataset has two properties that makes it particular challenging for classification algorithms that is it contains only categorical features (suite and rank of a card) and it's extremely imbalanced (2 out of 10 classes constitute 90% of the samples). since the standard data imbalance technique is not suitable here due to constant number of samples per class, for example there are only 480 combinations possible for Royal flush hence we can not increase it to use oversampling technique. Therefore, feature transformation is used to improve classification accuracy. After transforming the base dataset with 10 dimensions into 19 dimensions, data becomes more machine friendly (learning process for different classification model becomes easy) which results in improved classification accuracy for every classifier used in the experiment, particularly, decision tree achieves perfect F1 score of 1, which means 100% samples in test dataset are correctly classified.

References

- [1] Thomas Angeland, Kim-Andre Engebretsen, and Mustafa Numic. Poker hand induction: Multi-class classification of extreme imbalanced data with machine learning. *Machine Learning*, pages 1 – 26, 2019.
- [2] Walinton Cambroner. Poker hand dataset: A machine learning analysis and a practical linear transformation. *Machine Learning*, pages 1 – 11, 2021.
- [3] Robert Catral and Franz Oppacher. Poker hand dataset. UCI Machine Learning Repository, 2007. DOI: <https://doi.org/10.24432/C5KW38>.
- [4] Suraiya Jabin. Poker hand classification. In *International Conference on Computing, Communication and Automation (ICCCA2016)*, pages 1–5. Faculty of Natural Sciences, Jamia Millia Islamia,, 2016.