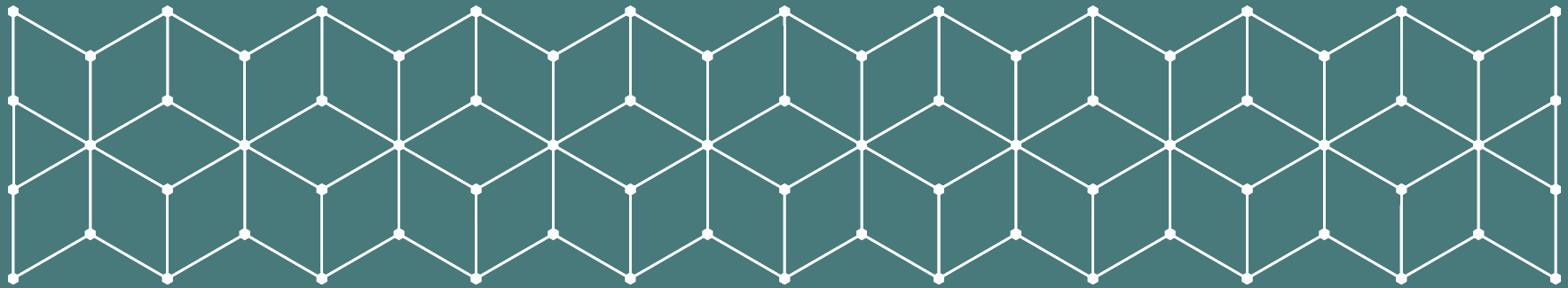
Østfold University College

Poker Hand Induction: Multi-class classification of extreme imbalanced data with neural networks



Thomas Angeland, Kim-Andre Engebretsen & Mustafa Numic | Master in Applied Computer Science

21.03.2019

Problem statement

Train a neural network to accurately assess poker hands
of 5 cards from a 52 card deck.

Østfold University College

Some info

- 52 unique cards in a deck.
- Each hand dealt contains 5 cards.
- 10 distinct, ranked poker hands.
- When order matters, There are 311 875 200 unique poker hands.
- 1 Royal Flush for every 649 740 hand dealt.

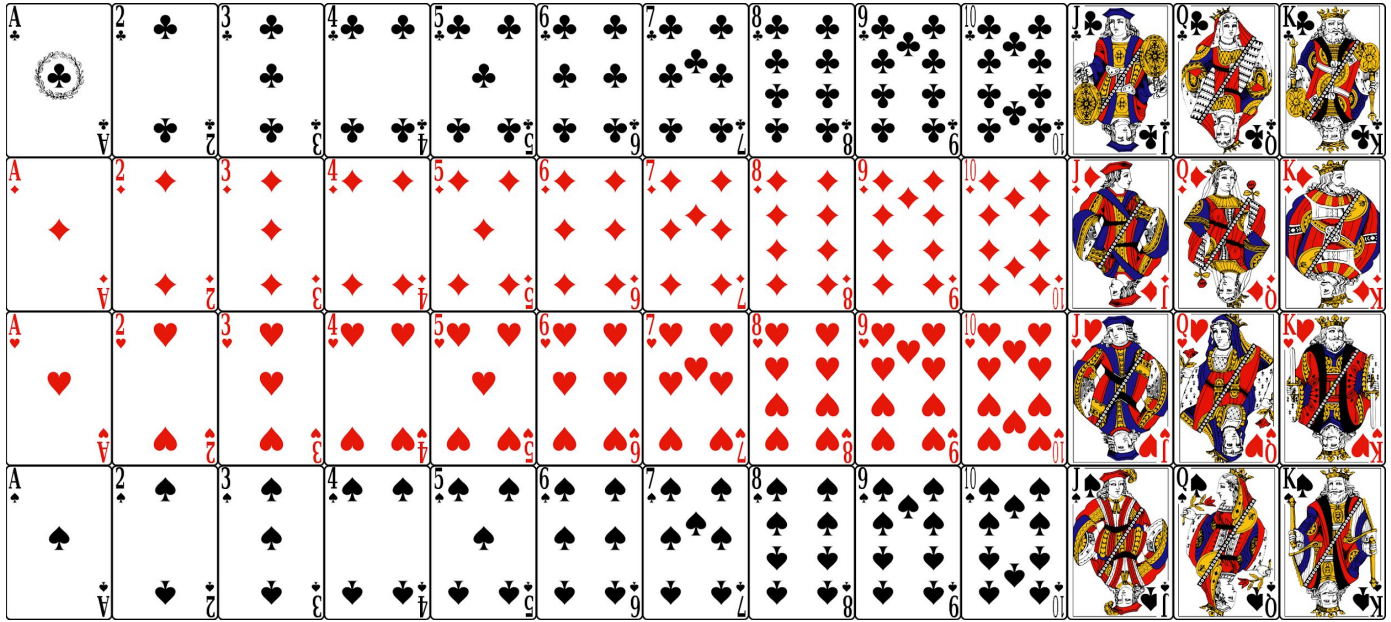
HAND	PROBABILITY	COMBINATIONS
Royal flush	0.00000154	480
Straight flush	0.00001385	4,320
Four of a kind	0.00024010	74,880
Full house	0.00144058	449,280
Flush	0.00196540	612,960
Straight	0.00392465	1,224,000
Three of a kind	0.02112845	6,589,440
Two pair	0.04753902	14,826,240
Pair	0.42256903	131,788,800
Nothing	0.50117739	156,304,800

Østfold University College

Cards (Predictors)

13 ranks

4 suits



Østfold University College

Poker hands (Classes)

9: Royal Flush

8: Straight Flush

7: Four of a kind

6: Full House

5: Flush

4: Straight

3: Three of a kind

2: Two pair

1: One Pair

0: Nothing in hand

Østfold University College

Dataset (derived from)

Title: Poker Hand Data Set

Abstract: Purpose is to predict poker hands

Source: https://archive.ics.uci.edu/ml/datasets/Poker+Hand

Robert Cattral (cattral@gmail.com)
Franz Oppacher (oppacher@scs.carleton.ca)
Carleton University, Department of Computer Science
Intelligent Systems Research Unit
1125 Colonel By Drive, Ottawa, Ontario, Canada, K1S5B6

Data Set Characteristics:	Multivariate
Attribute Characteristics:	Categorical, Integer
Associated Tasks:	Classification
Number of Instances:	1025010
Number of Attributes:	11
Missing Values	0
Date	2007-01-01

Dataset structure

In a hand with *n* cards, each card is denoted by a S[n] (suit, 1-4) value and C[n] (rank, 1-13) value.

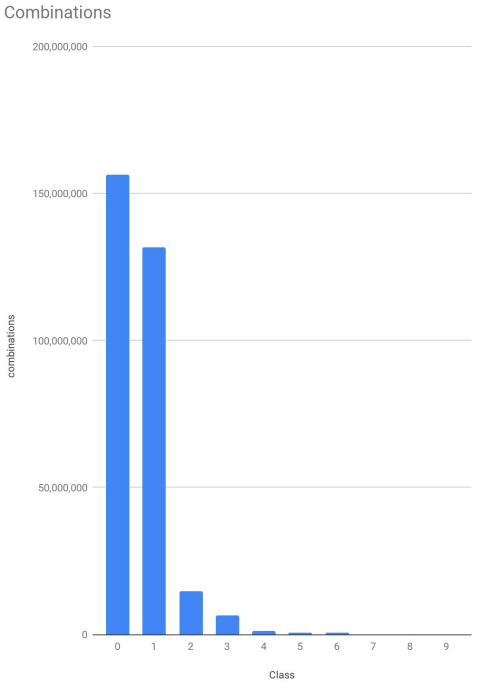
CLASS is assigned a value between 0 to 9 and represents how good the hand is.

Example:

S1	C1	S2	C2	S3	C3	S4	C4	S5	C5	CLASS
3	8	1	7	3	3	2	4	3	11	0
3	9	4	12	4	6	2	10	2	3	0
2	11	2	6	2	5	4	7	3	2	0
4	4	3	13	3	6	2	7	1	12	0
2	12	4	4	2	6	2	2	4	8	0
4	4	4	12	3	2	2	13	4	6	0
3	7	3	13	3	11	1	3	2	2	0
4	11	3	12	3	7	1	7	4	3	1
4	3	4	1	2	2	3	11	2	3	1
2	8	2	1	4	12	4	10	1	12	1
4	10	2	12	4	1	2	1	3	9	1
2	4	1	5	2	10	4	11	4	10	1

Dataset

Problem: Class distribution is very imbalanced.



Attempted to solve this by;

- generating a new dataset programmatically;
- extracting a stratified sample from the generated dataset;
- using under- and over sampling;
- using early stopping, i.e using evaluation set;
- using sample weights; and
- using f1 and geometric mean as evaluation metric.

López, V., Fernández, A., García, S., Palade, V., & Herrera, F. (2013). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Information Sciences, 250, 113–141. https://doi.org/10.1016/J.INS.2013.07.007

Preprocessing

"The SMOTE approach can improve the accuracy of classifiers for a minority class."

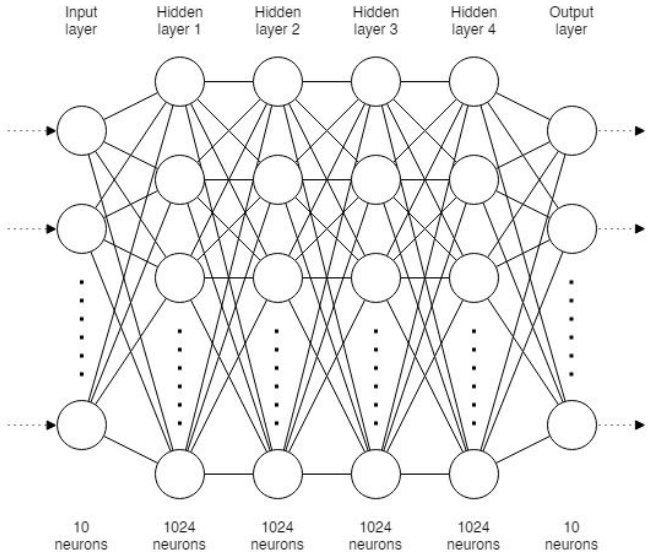
Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2011). SMOTE: Synthetic Minority Over-sampling Technique. Journal Of Artificial Intelligence Research. https://doi.org/10.1613/jair.953

In our testing: SMOTE increased geometric mean for neural networks, but not for decision trees and boosted trees.

SMOTE is used on the last four classes to train our best model.

Classifiers

- TensorFlow Keras (python)
- feedforwardnet (Matlab)



TensorFlow Keras model

- Network structure
  - 1 input layer
  - 4 hidden layers (1024-1024-1024-1024)
  - 1 output layer
- Loss function: Sparse Categorical Cross Entropy
- Metrics: Accuracy and F1
- Optimizer: Adam with default parameters
- Activation function in hidden layers: ReLu
- Activation function in output layer: SoftMax
- Epochs: 1000 or end if no improvement is made after 25 epochs.
- Batch size: 4096
- Callback with Early Stopping and Model Checkpoint.

Evaluation metrics

Accuracy is not a good indicator for multi-classification on imbalanced data. Geometric mean is.

G_{mean} reflects the ability to classify positive samples and negative samples at the same time.

$$G_{mean} = \sqrt{R_{t+} \cdot R_{t-}},$$

where R_{t+} represents true positive rate, which is calculated by $R_{t+} = N_{TP} / N_P$. R_{t-} represents true negative rate, which is formulated as $R_{t-} = N_{TN} / N_N$.

The higher the geometric mean, the better ability of the classifier to recognize positive class and negative class samples at the same time.

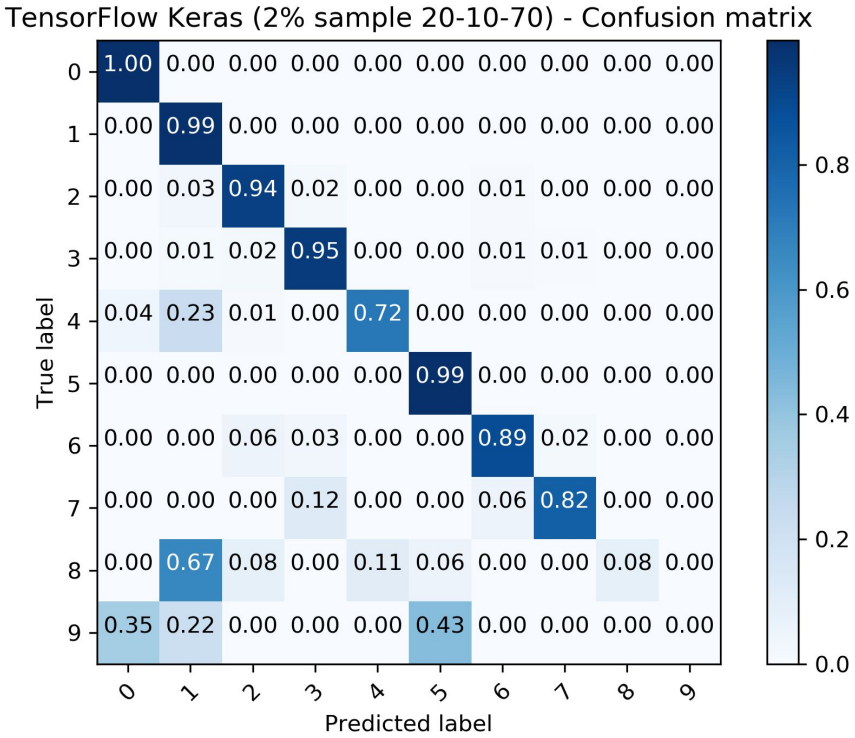
Li, F., Zhang, X., Zhang, X., Du, C., Xu, Y., & Tian, Y.-C. (2018). Cost-sensitive and hybrid-attribute measure multi-decision tree over imbalanced data sets. Information Sciences, 422, 242–256. https://doi.org/10.1016/J.INS.2017.09.013

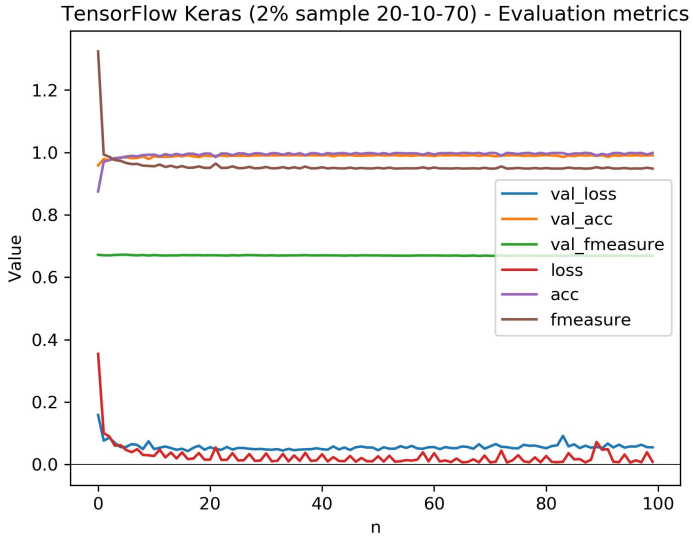
Observations while training...

- Other optimizers where considered, and while testing showed less zig-zagging and loss spikes with AdaDelta and Adamax, plain Adam achieved the highest score overall in our testing.
- Loss spikes during training
  - Accuracy decreases.
  - F-measure increases.
- Using class weights.
  - It yielded bad results in our testing with default Adam learning rate.
  - Decreasing the learning rate improved the score.
- Custom loss function: *Focal loss* for dealing with class imbalance.
  - Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal Loss for Dense Object Detection. Retrieved from http://arxiv.org/abs/1708.02002
  - It yielded bad results in our testing.
- When we increased the number of neurons and layers, more stress was put on the GPU and less stress was put on the CPU.

Østfold University College

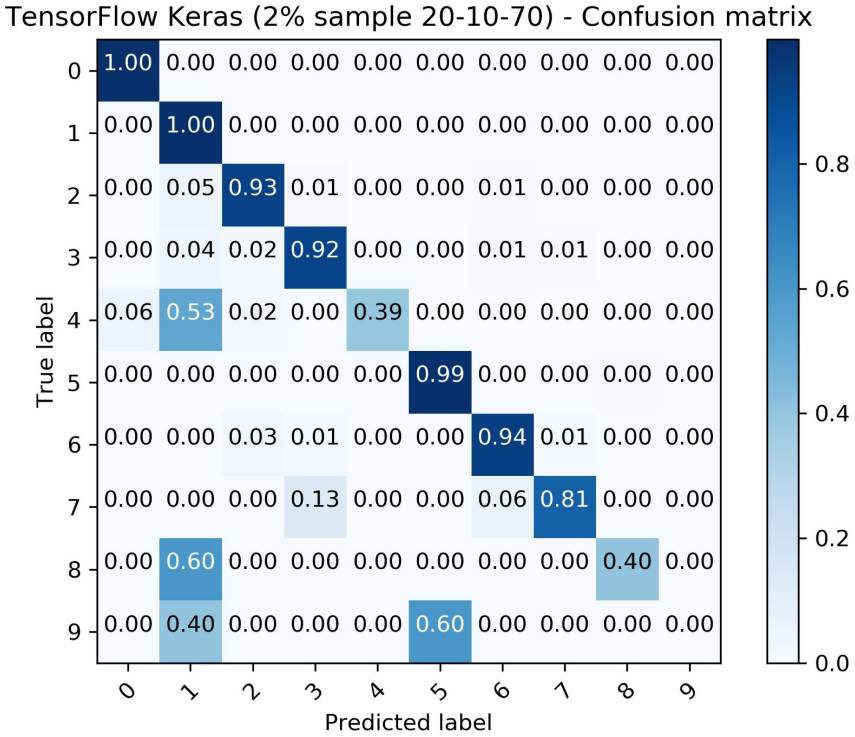
Tensorflow Keras SMOTE (sample to highest) - geometric mean: 0.858669

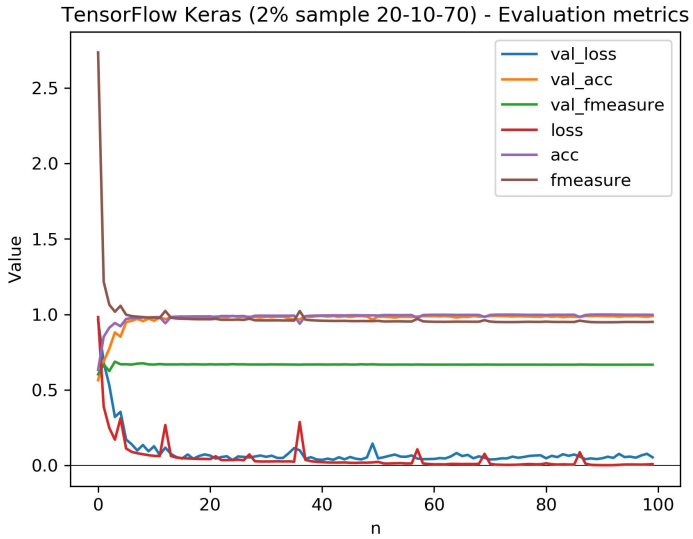




Østfold University College

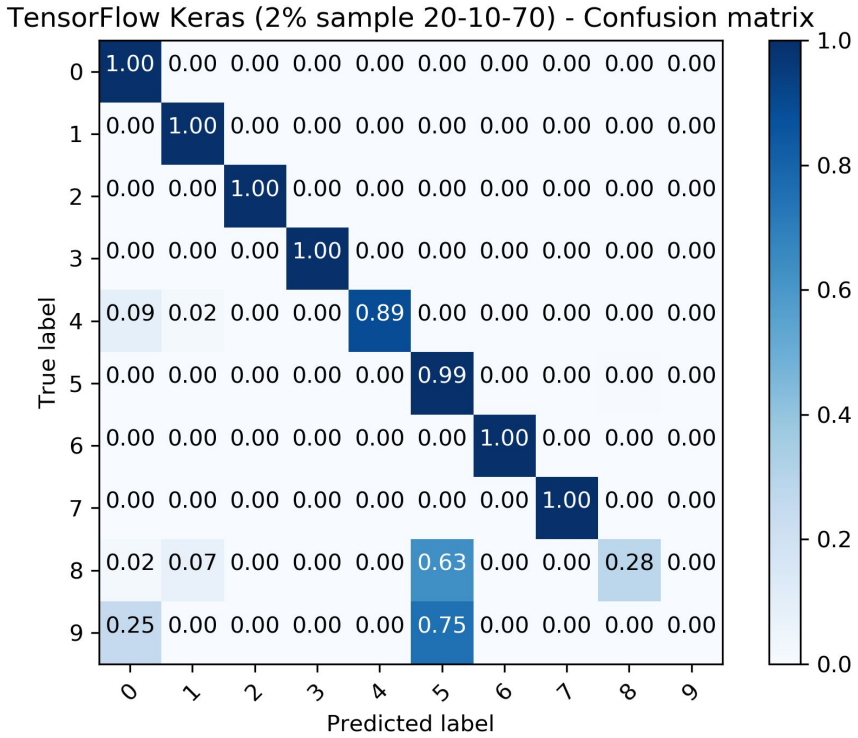
Tensorflow Keras 100k US + 100k OS (SMOTE) - geometric mean: 0.858083

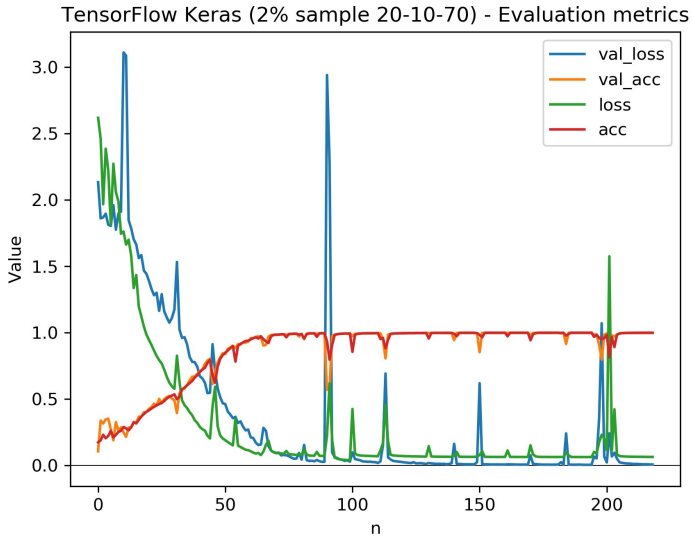




Østfold University College

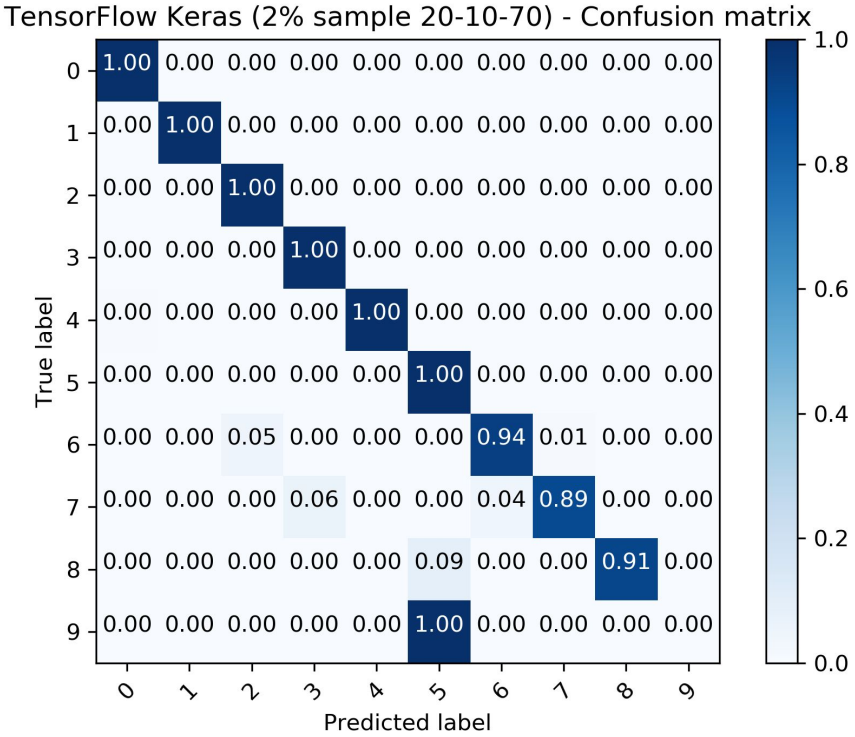
Tensorflow Keras with weights - geometric mean: 0.903304

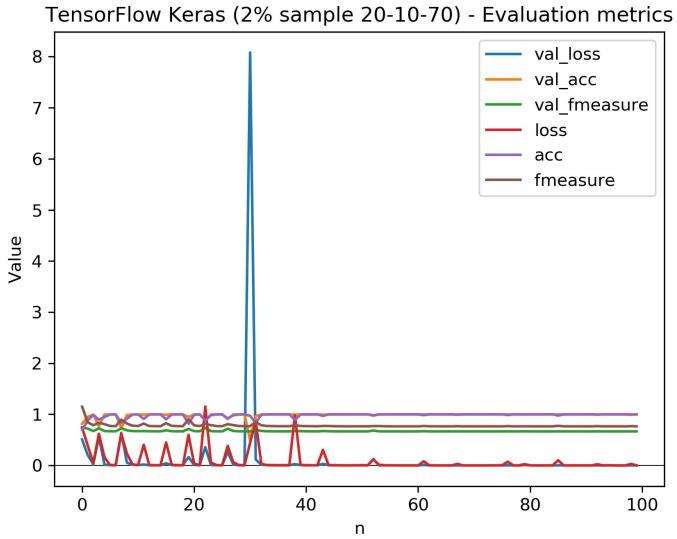




Østfold University College

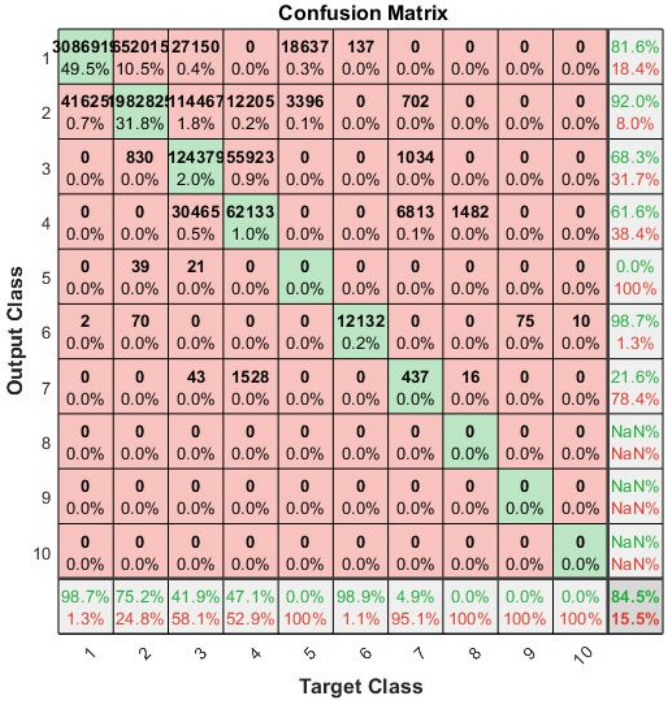
Tensorflow Keras 100k OS on 4 last (SMOTE) - geometric mean: 0.934816





Østfold University College

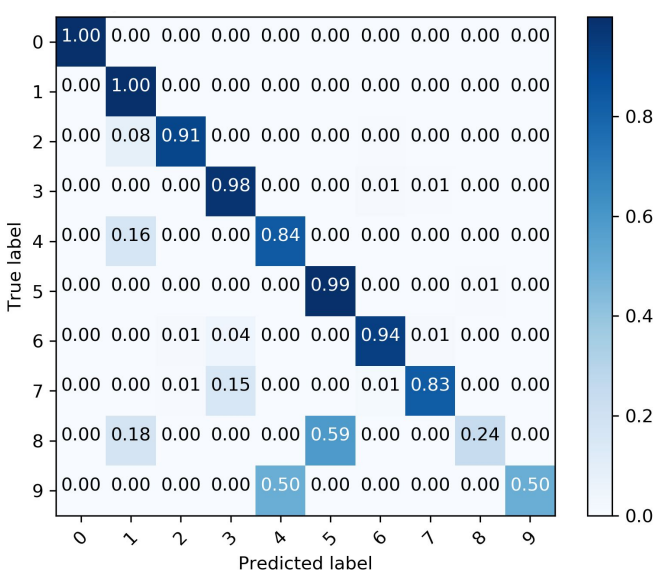
Matlab with standard dataset and 2% sample



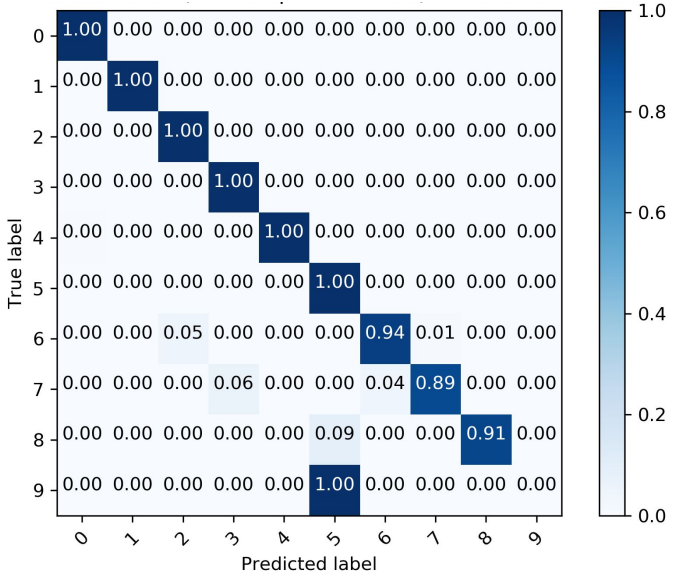
Østfold University College

Comparing trees with neural networks (our results)

Boosted trees: Catboost



Neural networks: TensorFlow Keras



Summary

Model	Sample size (%)	Sample distr. (%)	Accuracy	Geometric mean
AdaBoost	2	20/00/70	0.660446	0.690570
C5.0	2	20/00/70	0.738000	-
RandomForest	2	20/00/70	0.779012	0.826485
Matlab	**2**	**20/10/70**	**84.50000**	**-**
TF Keras (w/ SMOTE)	**2**	**20/10/70**	**0.992779**	**0.858669**
XGBoost	2	20/10/70	0.998940	0.861604
LightGBM	2	20/10/70	0.998926	0.866858
TF Keras (w/ weights)	**2**	**20/10/70**	**0.999373**	**0.903304**
CatBoost	2	20/10/70	0.993507	0.906385
TF Keras (w/ OS & US)	**2**	**20/10/70**	**0.999836**	**0.934816**

Results by others with neural networks

- 92.4% accuracy with 80/10/10 in Matlab.
  - Dişken, G. (2010). Predicting Poker Hands With Artificial Neural Networks. Adana. Retrieved from https://eembdersler.files.wordpress.com/2010/09/2014913024-gc3b6kaydic59fken-ann-proposal.pdf
- 57% accuracy.
  - Sihota, J. S. (2015). Poker Rule Induction. Retrieved February 5, 2019, from http://rstudio-pubs-static.s3.amazonaws.com/59520_f25e40618a3a442ca45b110b57a24a6f.html
- 94% accuracy with two hidden layers with 18 neurons and 10 neurons, respectively.
  - Garg, A. (2016). Predicting Poker hand using neural networks. Retrieved March 20, 2019, from https://medium.com/@akashg/predicting-poker-hand-using-neural-networks-83ed7d0bfc6a

Suggestions for future development

- Use custom classifiers or custom loss functions that focuses primarily on multi-class classification imbalance.
  - Bi, J., & Zhang, C. (2018). An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme. Knowledge-Based Systems, 158, 81–93. https://doi.org/10.1016/J.KNOSYS.2018.05.037
  - Li, F., Zhang, X., Zhang, X., Du, C., Xu, Y., & Tian, Y.-C. (2018). Cost-sensitive and hybrid-attribute measure multi-decision tree over imbalanced data sets. Information Sciences, 422, 242–256. https://doi.org/10.1016/J.INS.2017.09.013
  - Oh, K., Jung, J.-Y., & Kim, B. (2018). Imbalanced classification of manufacturing quality conditions using cost-sensitive decision tree ensembles AU - Kim, Aekyung. International Journal of Computer Integrated Manufacturing, 31(8), 701–717. https://doi.org/10.1080/0951192X.2017.1407447
  - Du, J., Vong, C.-M., Pun, C.-M., Wong, P.-K., & Ip, W.-F. (2017). Post-boosting of classification boundary for imbalanced data using geometric mean. Neural Networks, 96, 101–114. https://doi.org/10.1016/J.NEUNET.2017.09.004
  - Kim, M.-J., Kang, D.-K., & Kim, H. B. (2015). Geometric mean based boosting algorithm with over-sampling to resolve data imbalance problem for bankruptcy prediction. Expert Systems with Applications, 42(3), 1074–1082. https://doi.org/10.1016/J.ESWA.2014.08.025

Code repositories (ours)

Generating new data set (C#):
https://github.com/Alvtron/PokerData

R code:
https://github.com/Alvtron/ITI43210-Machine-Learning

Python code:
https://github.com/Alvtron/PythonMachineLearning

Østfold University College