War Predictions - Comparative Study of Supervised Learning and Neural Network Approaches

Project Type 2

Group 8

Linh Do, 8682188

Haoyuan Li, 8341378

Github link: https://github.com/tysonlihaoyuan/CSI4106Project

1. Timesheet

Team member	Task	Hours spent	Challenge
Linh Do	Learn and implement DT Classifier	3	Understand algorithm, parameters
	Learn and implement DT Regressor	3	Understand algorithm, parameters
	Learn and implement graphviz	2	Understand parameters
	Implement logistic regression	1	
	Feature selection and data preprocessing	4	Decide which features are relevant and valid, clean data
	Train and evaluation	4	
	Writing report	5	
Li Hao Yuan	Learn and implement Keras package	6	Familiar the function
	Learn and implement deep Learning for classifier regression	6	Understanding loss function and optimizer
	Learn and implement encoding the dataset	2	Advanced with one_hot_encode
	Train and test the data	3	Different methods in evaluate the data
	Graph and analyze the result	2	
	Report	4	

2. Introduction

Problem studied: To compare the different machine learning approaches in predicting wars and conflicts of three types. Predictions include classification (binary and multi-class) and regression on belligerents, outcome, casualty, initiation, intervention, location, duration, etc. The Area project domain is military/defense and international relations.

Context: There are hundreds of war and conflict around the world between 1816 and 2007. Wars have been categorized by whether they primarily take place between/among states, between/among a state and a non-state actor, and within states. We will be using the dataset on these wars and conflicts, as well as different machine learning approaches to analyze the dataset and conduct predictions.

Link to AI: We will use regression and (binary and non-binary) classification methods in supervised machine learning and neural network deep learning to make predictions.

Why it is important: War/conflict predictions are important for policymakers for both preventing conflicts and deciding whether or not to enter a particular conflict. This is also interesting for researchers because there might be some long-term trends in the field of human conflict.

3. Dataset description / Features

Where it comes from

https://www.kaggle.com/umichigan/interstate-wars?fbclid=IwAR008QAZiJ3HeH-P22QxL2pCLPYnIc1T1ci0on68RzYdYMKao8piwHqI-Hk

What does it contain:

1. 'Interstate.csv' - the wars between state and state
2. 'Intrastate.csv' - the wars between non-state and non-state
3. 'Extrastate.csv' - the wars between state and non-state

Size (number of rows and columns)**:**

1. 'extrastate.csv': 198 rows, 27 columns
2. 'interstate.csv': 137 rows , 24 columns
3. 'intrastate.csv': 442 rows , 27 columns

Did it need to be cleaned, did you need to sample it:

Extrastate

	Classification	Regression
Input	'initiation', 'combat_location', 'intervention', 'state_fatalities', 'nonstate_fatalities', 'side1_code', 'side2_code'	'initiation', 'combat_location', 'intervention', 'state_fatalities, 'nonstate_fatalities', 'side1_code', 'side2_code', 'outcome'
Output	'outcome', 'intervention'	'nonstate_fatalities', 'state_fatalities'

Intrastate - using "dropnan" to clean empty cell and invalid values in dataset

	Classification	Regression
Input	'combat_location', 'side1_code', 'side2_code', 'side1_fatalities', side2_fatalities	'combat_location', 'side1_code', 'side2_code',' outcome', 'international_war'
Output	'outcome', ' international_war'	'Side1_fatalities', 'side2_fatalities'

Interstate

	Classification	Regression
Input	'side','initiation','combat_location','combat_fatalities'	''side','initiation','combat_location', 'outcome'
Output	'outcome'	'combat_fatalities'

4. Details of the methods used

A. **Method name and description** (what is it, what does it do)

   a. **Train_model( )**

This function trains a given model and returns the accuracy score. It takes as arguments the untrained model to train, the encoded attributes of the training set, and the target values of the training set.

b. **Create_model( )**
The method creates a sequential model with length of feature in dataset variable input and 10 hidden layer and length of feature in dataset potential output , and uses ‘categorical_crossentropy’ as loss function, ‘adam’ for optimizer. Using the built-in function ’fit to train and test the encoded dataset.

c. **Create_model_fold( )**
The method creates a sequential model with length of feature in dataset variable input and 10 hidden layer and potential output of y, and uses ‘categorical_crossentropy’ as loss function, ‘adam’ for optimizer. Using the built-in ‘kfold’ with 5 fold to train and test the encoded dataset.

d. **Create_model_re( )**
The method is similar to the method create model. The modification in view the mean_square_error and the KerasRegressor function.

B. Why did you choose this method?

a. The train_model method was chosen because of its ability to train for multiple epochs when necessary and output the score.

C. Details of implementation (platform/package used, or re-implementation)

a. The platform and packages use include Keras, TensorFlow, sklearn, pandas, and graphviz.

5. Feature engineering

1. **What features did you use? How did you obtain them** (any transformations needed or did they come from the dataset)?
   a. Only subsets of features were used. Features used: 'Side', 'initiation', 'combat_location', 'combat_fatalities, ‘outcome’, ‘side1_fatalities’, ‘side2_fatalities’, 'intervention', 'side1_code', 'side2_code', 'state_fatalities', 'nonstate_fatalities', ‘international_war’.
   b. One-hot encoding using a one-hot encoder was performed on the selected feature sets.
   c. Labelencoder is use for the output.
2. **Qualitative importance of the features** (your expertise in the domain perhaps tells you what is important)
   a. The chosen features are the most relevant for the predictions. They are the most important factors in wars that military experts look at.
3. **Quantitative importance of the features** (tests like feature ablation - removing one feature at a time to see its impact)
   a. After removing one feature at a time for a couple of times, we observed that no feature has a significant impact on the result.

6. Analysis of Results

1. **Evaluation method**
   Accuracy

Interstate			
Approach	Classcification train/test	Regression train/test	MSE train/test
Logistic R	0.87 / 0.84		
Decision tree	0.98 / 0.95	0.97 / 0.86	
Deep learning	0.96 / 0.82	0.83 / 0.87	0.0012 / 0.0012
Extrastate			
Approach	Classcification train/test	Regression train/test	MSE train/test
Logistic R	0.88 / 0.90		
Decision tree	1.0 / 1.0	0.99 / 0.98	
Deep learning	0.90 / 0.68	0.73 / 0.68	0.0023 / 0.003
Intrastate			
Approach	Classcification train/test	Regression train/test	MSE train/test
Logistic R	0.94 / 0.92		
Decision tree	0.92 / 0.95	0.99 / 0.99	
Deep learning	0.83 / 0.51	0.80 / 0.40	0.0012 / 0.003

Cross-validation

Deep learning (interstate): 0.78

Deep learning (extrastate): 0.62

Deep learning (intrastate): 0.50

2. **Comparative results**

Logistic regression is the baseline approach.

Decision tree vs baseline Approach:

In interstate, decision tree performs better at classification

In extrastate, decision tree performs better at classification

In intrastate, decision tree performs better at classification

Baseline approach vs Deep learning:

In interstate, baseline approach performs better at classification

In extrastate, baseline approach performs better at classification

In intrastate, baseline approach performs better at classification

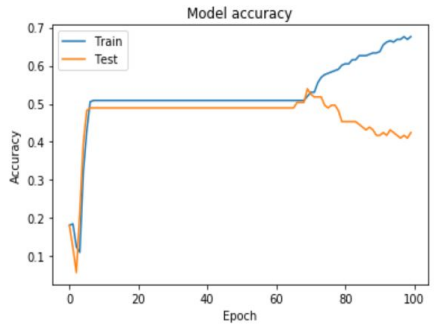
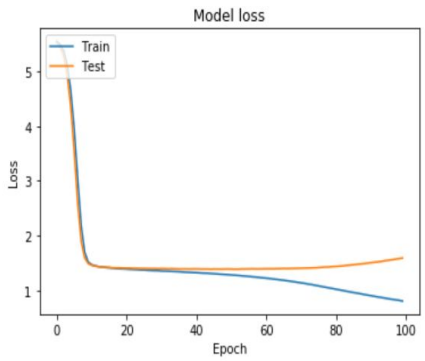
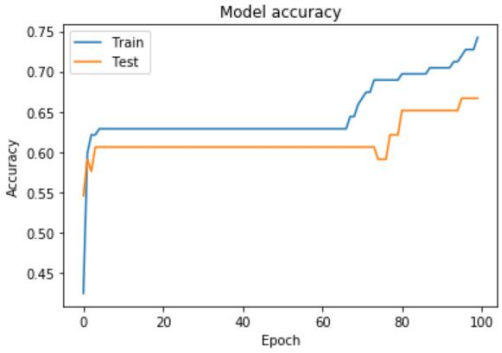
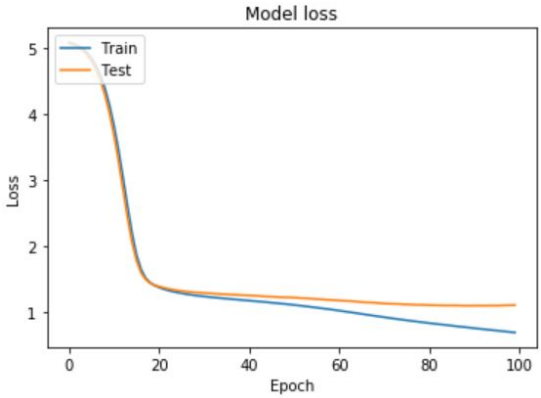
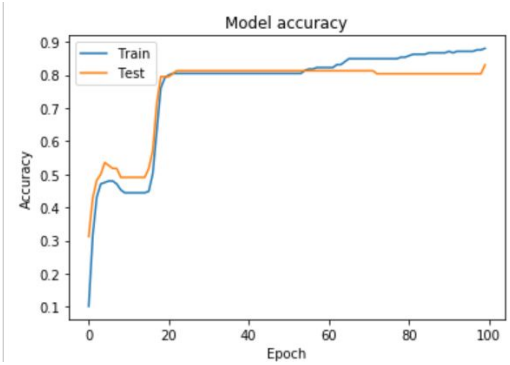
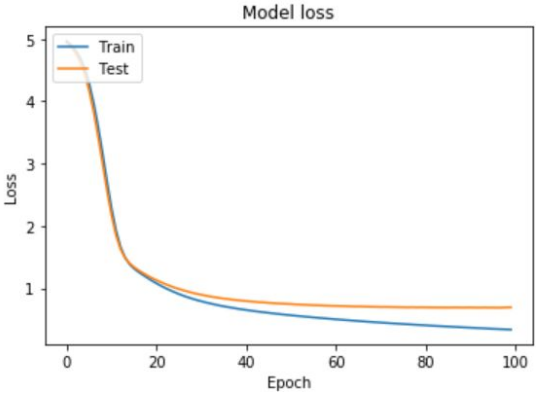
Decision tree vs Deep learning:

In interstate, decision tree performs better at both classification and regression

In extrastate, decision tree performs better at both classification and regression

In intrastate, decision tree performs better at both classification and regression

	Model accuracy	**Model loss**

intrastate		
extrastate		
interstate		

7. Conclusion

After some experimentation, we can see that decision tree performs the best, then logistic regression, and then deep learning. This is probably due to the fact that the Keras neural networks have not been adequately optimized. In the context of the war/conflict problem, the models have been able to make several useful predictions, including outcomes, fatalities on both sides, the existence of intervention, and international war on interstate, extrastate, and intrastate wars. Future projects should build upon these results to improve the accuracy.

References

Code:
https://scikit-learn.org/stable/modules/tree.html
https://machinelearningmastery.com/regression-tutorial-keras-deep-learning-library-python/

https://www.cnblogs.com/peng8098/p/keras_3.html

Other information:
https://en.wikipedia.org/wiki/Correlates_of_War
http://www.correlatesofwar.org/data-sets/COW-war/inter-state-wars-codebook
http://www.correlatesofwar.org/data-sets/COW-war/extra-state-wars-codebook
http://www.correlatesofwar.org/data-sets/COW-war/intra-state-war-data-codebook