Timing the Bluff: Predicting Deception in Poker through Timing

Cynthia Du, Ryan Tolone, Ivan Karp, Minjae Kim, GyeongMin Kwon

Introduction

Poker is a game of skill, strategy, and deception. One of the most critical aspects of the game is the ability to bluff - convincing opponents that a weak hand is strong or vice versa. But, would it be possible to use data and machine learning to detect when a player is bluffing? This project explores whether or not it is possible to detect if a player is bluffing based on various in-game factors, such as decision time, player position, and bet sizing, with a primary focus on whether the time it takes to bet can uncover deception. The ability to identify bluffs in poker has significant implications not only for poker strategy, but also for broader fields like behavioral economics, game theory, and artificial intelligence.

Primary Research Question: Is there a statistically significant difference between betting times and whether the hand is a bluff? If so, can we estimate a bluff based on betting timing?

Background Research

Much of the analyses on Poker revolves around psychological behaviors on bluffs and timings, but not a lot of statistical analyses is done based on real-life poker data. Statistical analysis that had already been done are very one-dimensional, predicting purely based on your opening hand. Our research aims to provide a new perspective on Poker analyses using unique variables like blinds, betting, raising, folding, and showdown results.

Poker Terminology

Hand Cards: 2 Cards held by the Player

Community Cards: Cards on the table, shared by all participating players

Position: Playing order, rotates every round

- **Small Blind**: player that automatically bets half the blind size
- **Big Blind**: player that automatically bets the full blind size
- **Dealer**: last player to perform an action

Showdown: Poker rounds that play until the participating players reveal their cards

Buyin: How much $ a player sits down at the table with

Actions: An action performed by a player during their turn

- **Bet/Raise**: Player raises the minimum pot amount to play
- **Call/Check**: Player matches a bet/raise deemed by another player
- **All In**: Player bets their entire stack

Street: A sequence of chronological stages within a round of poker

- **Pre-Flop**: hand cards of all players are distributed and individually checked, no community cards
- **Flop**: 3 community cards are revealed on the table
- **Turn**: 4th community card is revealed on the table
- **River**: 5th community card is revealed on the table

Data Collection and Engineering

In this study, we collected hand histories from private, real-money cash games hosted on the PokerNow.club platform. These games featured blinds ranging from $0.25/$0.50 up to $2/$5 ($50 to $500 buy-in). After each session, we exported the detailed game logs—encompassing actions such as posting blinds, betting, raising, folding, and showdown results—into CSV files.

We then processed and cleaned these logs, filtering out administrative lines and any incomplete or irrelevant records. The resulting dataset retained all pertinent gameplay actions, including each player's decisions (e.g., bets, calls, raises, folds) and timing information. This allowed us to label each hand as a bluff or value scenario and engineer various features (such as bet size ratios, decision times, and positional context) for subsequent analyses.

Data Collection and Engineering

Given the abundance in data values, we decided to discard any anomalies from standard Texas Hold'em 2 hand poker. From these data values, we extracted hands that went to showdown for verification of our dependent variable bluff/value.

We also extracted information of community/hand cards pertaining to the round, engineering variables such as winning_hand and board_evaluation. The variable board_evaluation was then used to engineer variables analyzing the current state of the community cards. Any variables that was directly related to the outcome were only used to verify the validity of the dependent variable.

Dataset

handType: response variable (value or bluff, determined by whether the villain or hero won the pot)

Player_type: Hero and Villain (villain: the last aggressor in the hand, hero the person that calls the aggressor)

Position: IP and OOP (IP: take action after your opponent, OOP: take action before your opponent)

Street: The stage in the betting sequence of a round (preflop, flop, turn, river)

Decision_time: The time it took to make a decision

Action: What type of action the player took (check, raise, bet, call, fold)

PIP: “Players in Pot” the # of players that have not folded yet

betRatio: Bet ratio compared to the original pot

Board: The cards that are revealed on the board

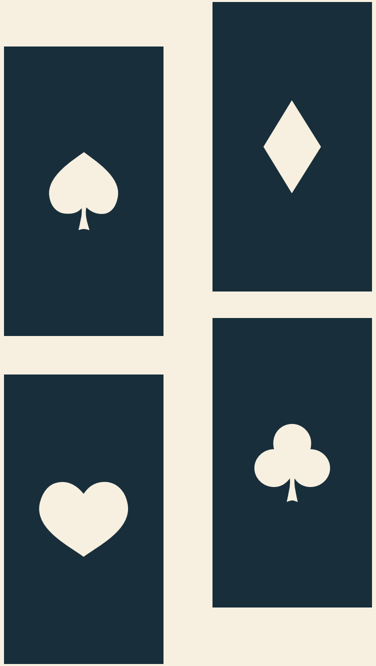
Player_hand: The pair of cards we are taking perspective from

Winning_hand: Combinations available based on our hand and the board

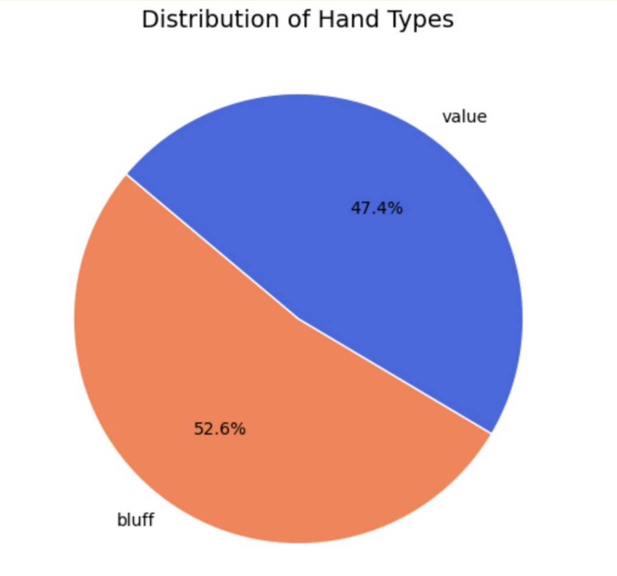
Hand_ranking: The combination ranking of our winning_hand

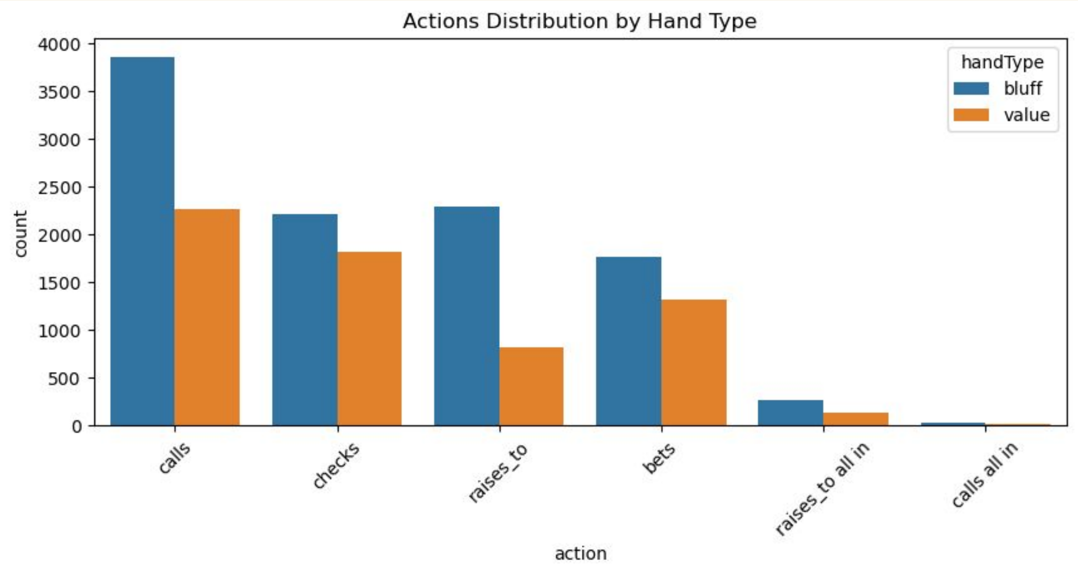
Board_evaluation: Combinations available on the board + potential combinations that are at risk

* The variable we are interested in is **handType**, predicting whether a hand was a “bluff” or “value”

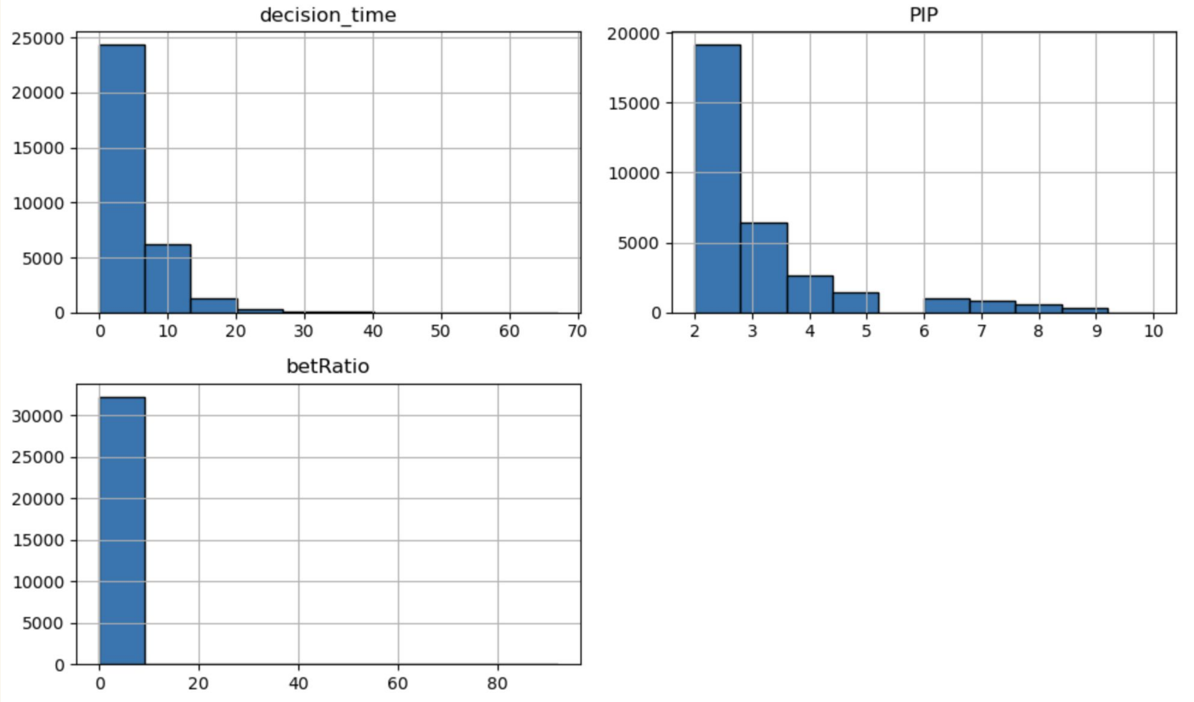


Exploratory Data Analysis

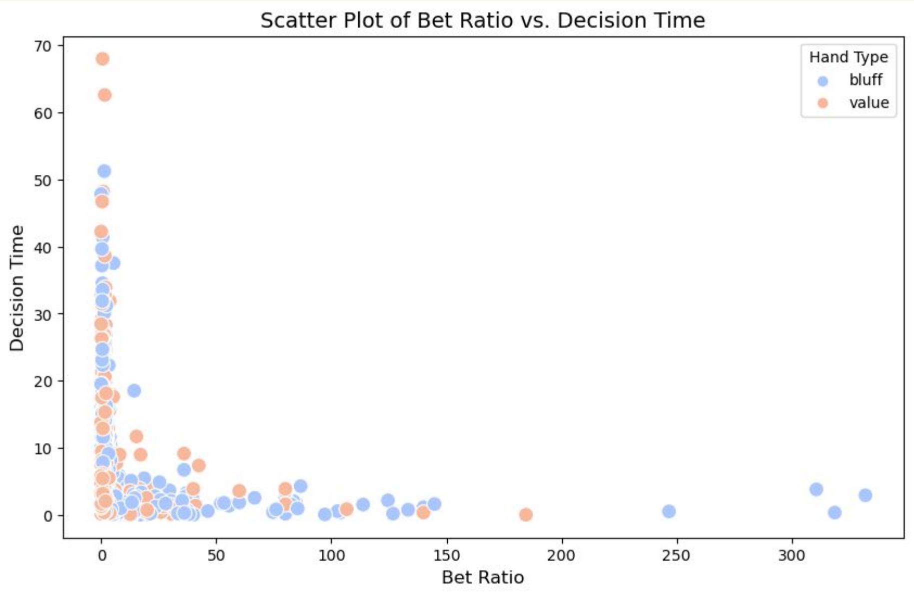


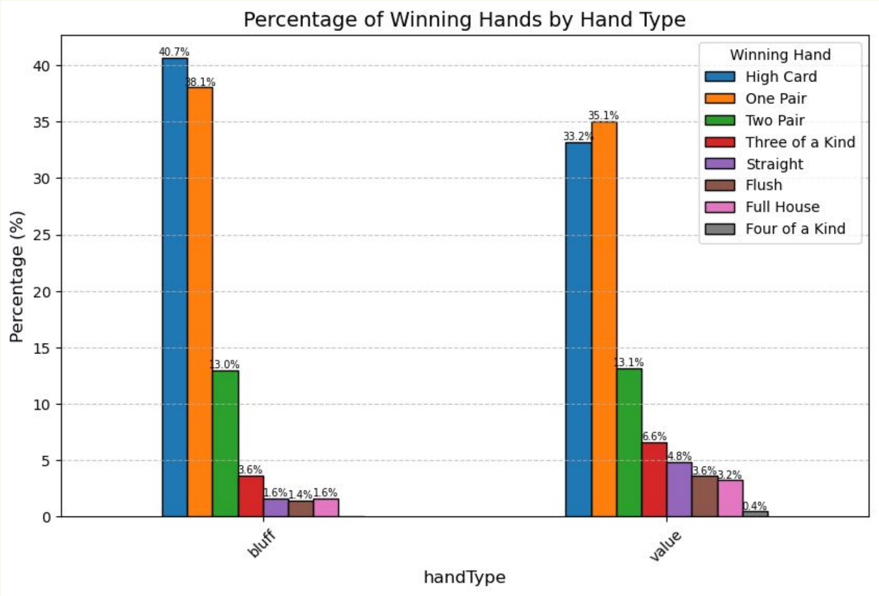


Exploratory Data Analysis - Numeric Columns

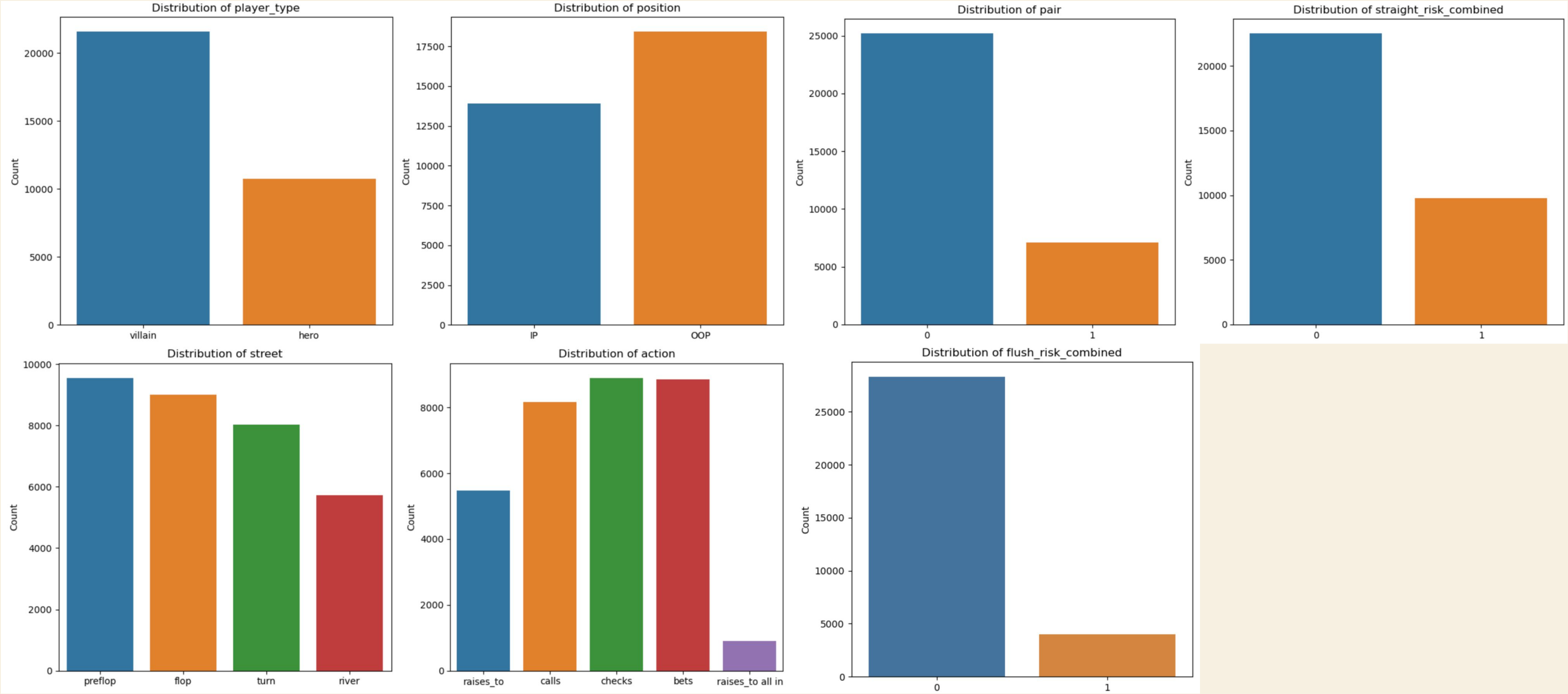


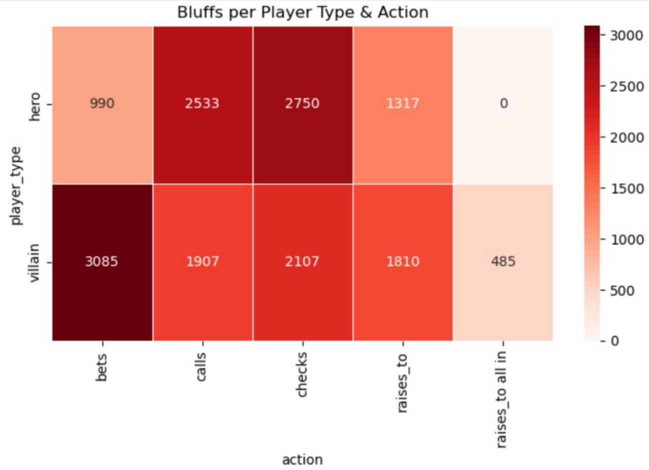
Exploratory Data Analysis - handType

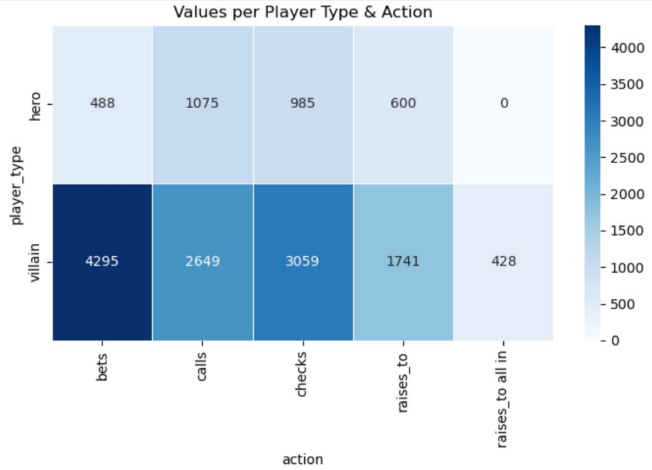


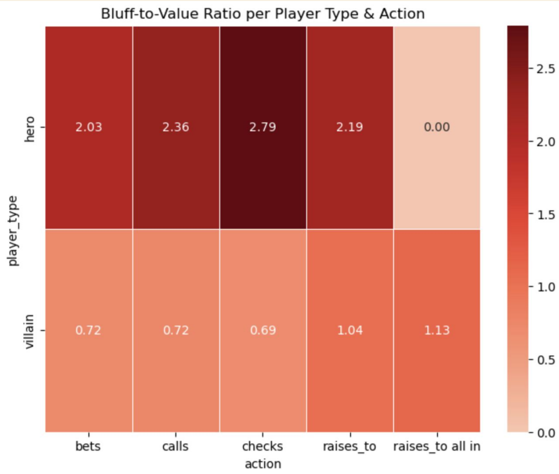


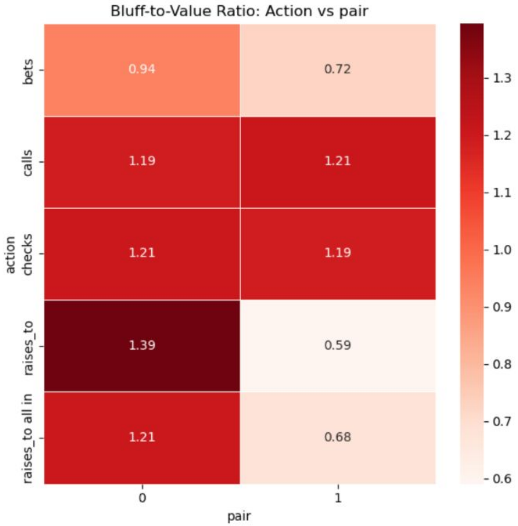
Exploratory Data Analysis - Discrete Columns

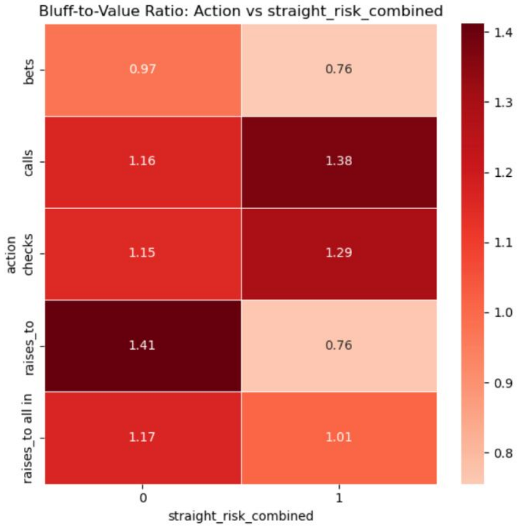


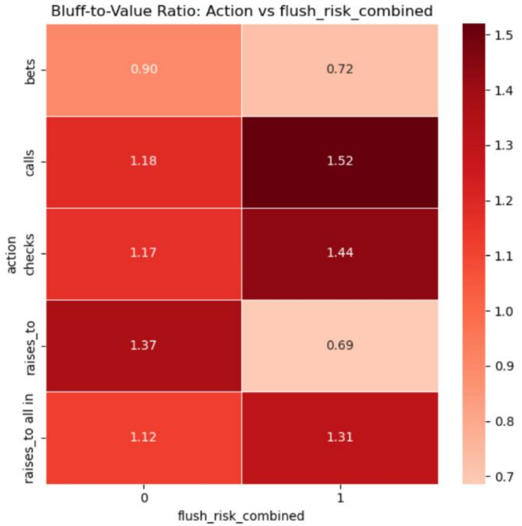

Bluffs per Player Type & Action
player_type
hero
villain
990
2533
2750
1317
0
3085
1907
2107
1810
485
bets
calls
checks
raises_to
raises_to all in
action
3000
2500
2000
1500
1000
500
0

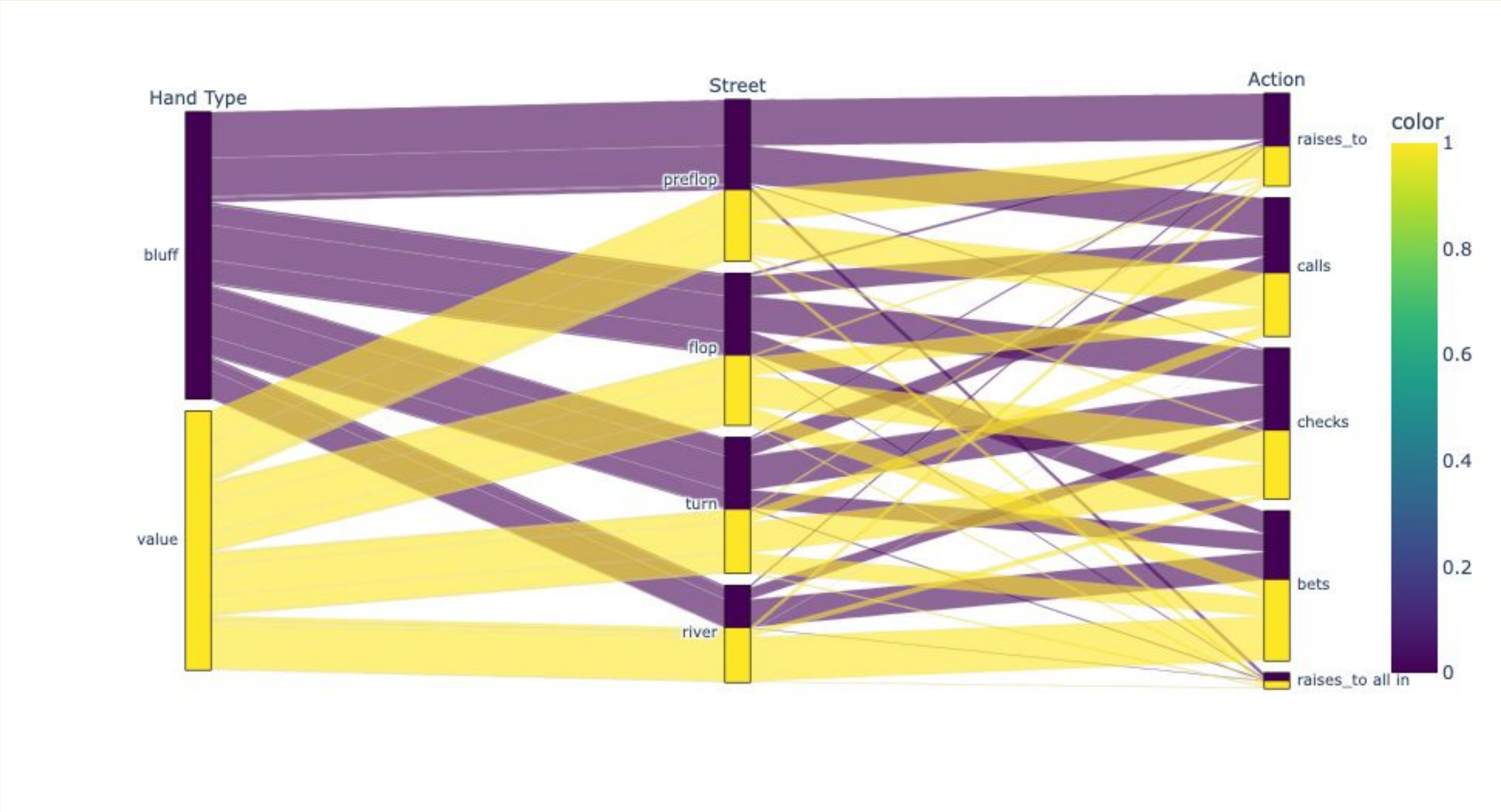

Values per Player Type & Action
player_type
hero
villain
488
1075
985
600
0
4295
2649
3059
1741
428
bets
calls
checks
raises_to
raises_to all in
action
4000
3500
3000
2500
2000
1500
1000
500
0


Bluff-to-Value Ratio per Player Type & Action
player_type
hero
villain
2.03
2.36
2.79
2.19
0.00
0.72
0.72
0.69
1.04
1.13
bets
calls
checks
raises_to
raises_to all in
action
2.5
2.0
1.5
1.0
0.5
0.0


Bluff-to-Value Ratio: Action vs pair
action
bets
calls
checks
raises_to
raises_to all in
0.94
0.72
1.19
1.21
1.21
1.19
1.39
0.59
1.21
0.68
0
1
pair
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6

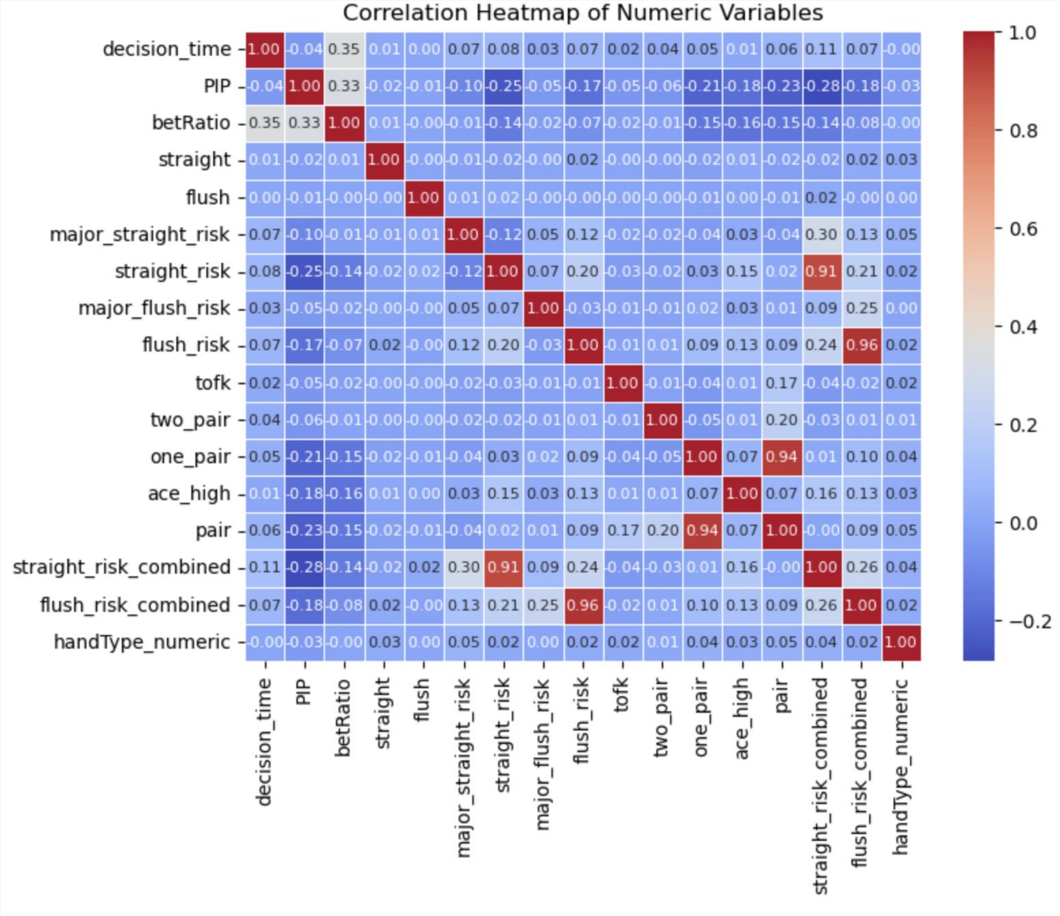

Bluff-to-Value Ratio: Action vs straight_risk_combined
action
bets
calls
checks
raises_to
raises_to all in
0.97
0.76
1.16
1.38
1.15
1.29
1.41
0.76
1.17
1.01
0
1
straight_risk_combined
1.4
1.3
1.2
1.1
1.0
0.9
0.8


Bluff-to-Value Ratio: Action vs flush_risk_combined
action
bets
calls
checks
raises_to
raises_to all in
0.90
0.72
1.18
1.52
1.17
1.44
1.37
0.69
1.12
1.31
0
1
flush_risk_combined
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0.7



Hand Type
Street
Action
bluff
value
preflop
flop
turn
river
raises_to
calls
checks
bets
raises_to all in
color
1
0.8
0.6
0.4
0.2
0

Feature Selection



Models

MLP

Multi-layer Perceptron. Feedforward neural network with multiple hidden layers.

LSTM

Long Short Term Memory. Type of Recurrent Neural Network

SVM

Support Vector Machine. Supervised learning, good at binary classification

Random Forest

ML model using multiple decision trees

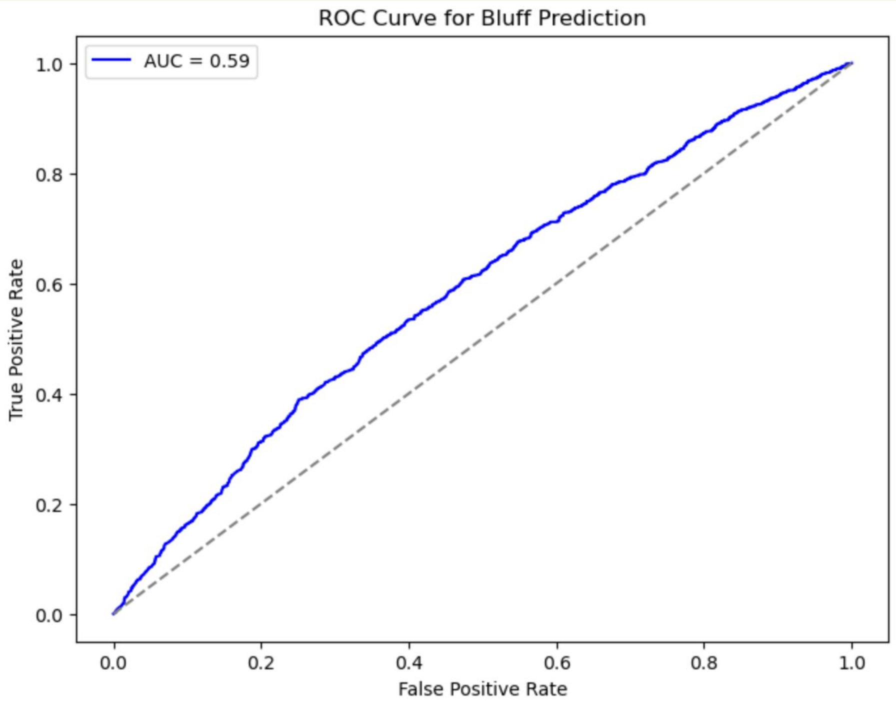
XGBoost

Uses gradient-boosted decision trees

Logistic Regression

Supervised learning, uses statistical model to predict categorical vars

Logistic Regression Model



Accuracy: 0.63

[[100 1136]
 [105 2024]]

	precision	recall	f1-score	support
0	0.49	0.08	0.14	1236
1	0.64	0.95	0.77	2129
accuracy			0.63	3365
macro avg	0.56	0.52	0.45	3365
weighted avg	0.58	0.63	0.54	3365

AUC Score: 0.59

ACCURACY OF ALTERNATIVE MODELS WITH RELATIVE VARIABLE IMPORTANCE.

	Random Forest	xgBoost	SVM	Log. Reg.
Test Accuracy (%)	62.4	61.0	61.4	61.6
1st Variable	Player Type	**Decision Time**	Bet Ratio	Player Type
2nd Variable	**Decision Time**	Bet Ratio	Player Type	Flush Risk
3rd Variable	Bet Ratio	Player Type	PIP	Position
4th Variable	PIP	Ace is High	All In	**Decision Time**

Key Results

- The baseline models are insufficient for accurately predicting a bluff
- Decision time proves highly important in all but the SVM model

MLP Model

What is a MLP Model?

- A type of model that uses multiple layers (input layer, hidden layer, output layer) to specialize in classification and regression.

Why it might fit this dataset:

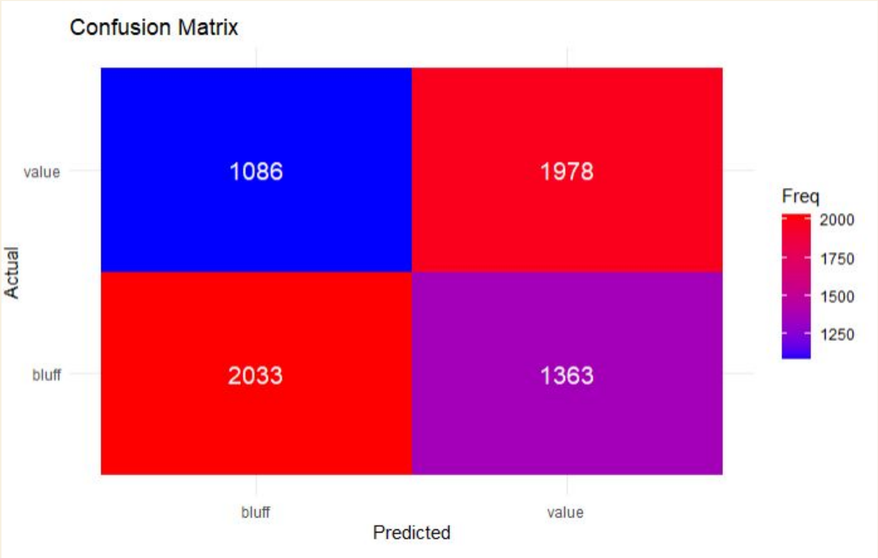
- Since predicting hand_type as response variable is categorical, we can use MLP model for classification. It can take in complex patterns between different variables to provide a more accurate prediction.
- Opposed to other traditional linear models, MLP can use hidden layers to provide more insight into interactions between different variables.

MLP Results

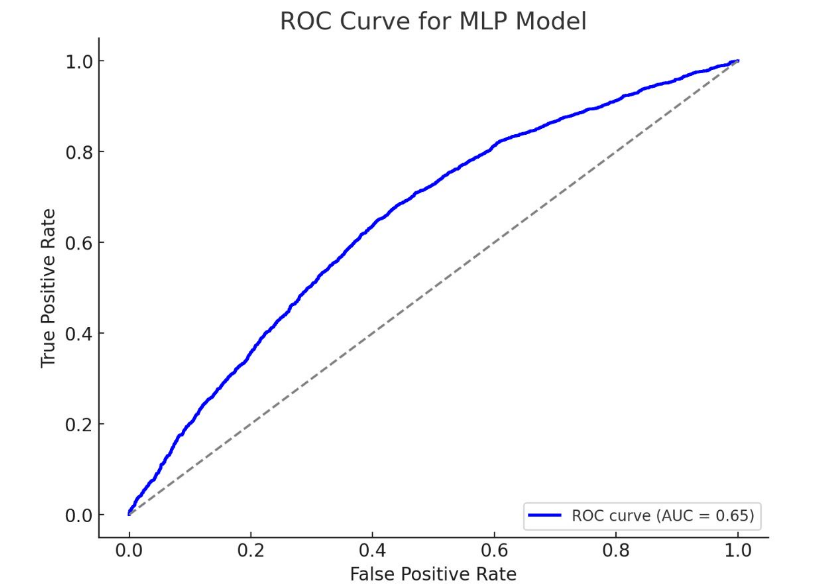
Test Accuracy: 0.6209

Mcnemar's Test P-Value : 2.444e-08

Sensitivity : 0.5986
Specificity : 0.6456
Pos Pred Value : 0.6518
Neg Pred Value : 0.5920
Prevalence : 0.5257
Detection Rate : 0.3147
Detection Prevalence : 0.4828
Balanced Accuracy : 0.6221
‘Positive' Class : bluff



MLP Model



- **0.65** indicates a relative separation between positive and negative classes, but is not highly accurate.
- The model would only be suitable in specific cases.

Long Short Term Memory Model

What is an LSTM?

- A type of recurrent neural network (RNN) designed to capture long-term dependencies in sequential data.
- LSTM units store and update a memory cell over time, allowing the network to retain relevant information from earlier in the sequence and effectively learn long-term dependencies.

Why it might fit this dataset:

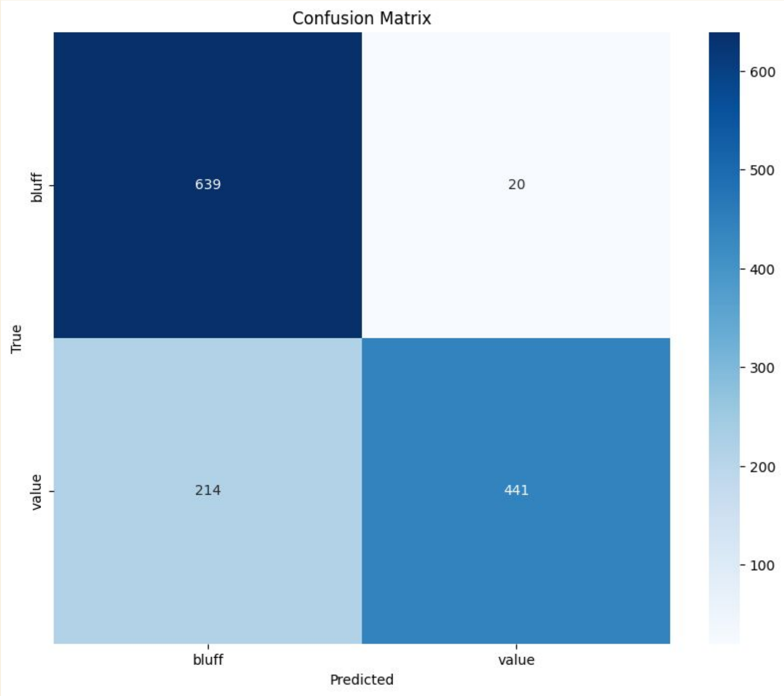
- Naturally handles variable-length sequences, matching the sequential nature of poker hand actions.
- Captures long-term dependencies in decision-making steps, improving classification of complex, time-dependent behaviors.
- Can learn complex patterns from the sequential nature of game actions and outcomes.

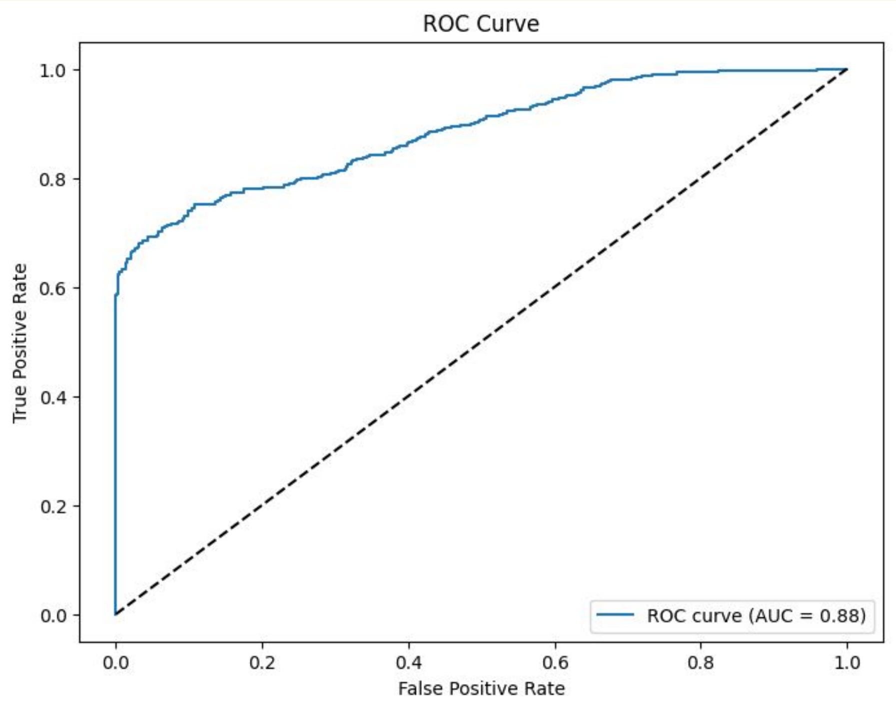
LSTM Results

Test Accuracy: 0.8250

Evaluating model on test set:
Classification Report:

	precision	recall	f1-score	support
bluff	0.75	0.97	0.85	659
value	0.96	0.67	0.79	655
accuracy			0.82	1314
macro avg	0.85	0.82	0.82	1314
weighted avg	0.85	0.82	0.82	1314

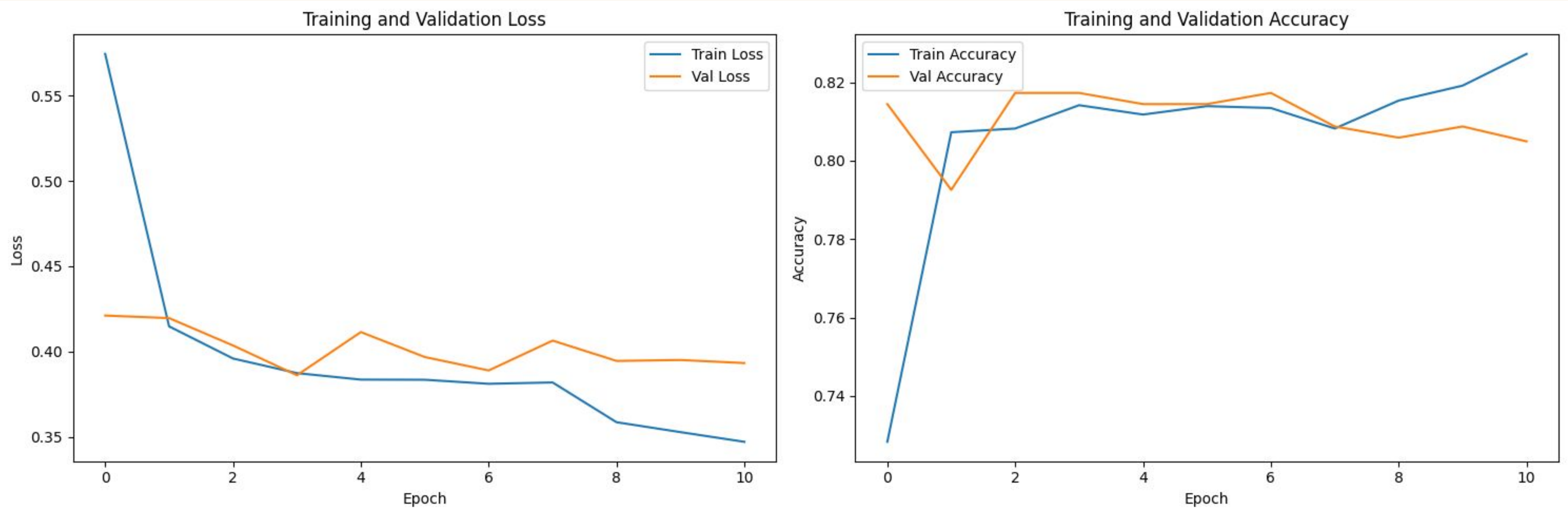




ROC Curve

- **High AUC (0.88)** indicates strong separation between positive and negative classes.
- Model reliably distinguishes “bluff” vs. “value” (or similar classes) in most scenarios.

Loss And Accuracy Progression



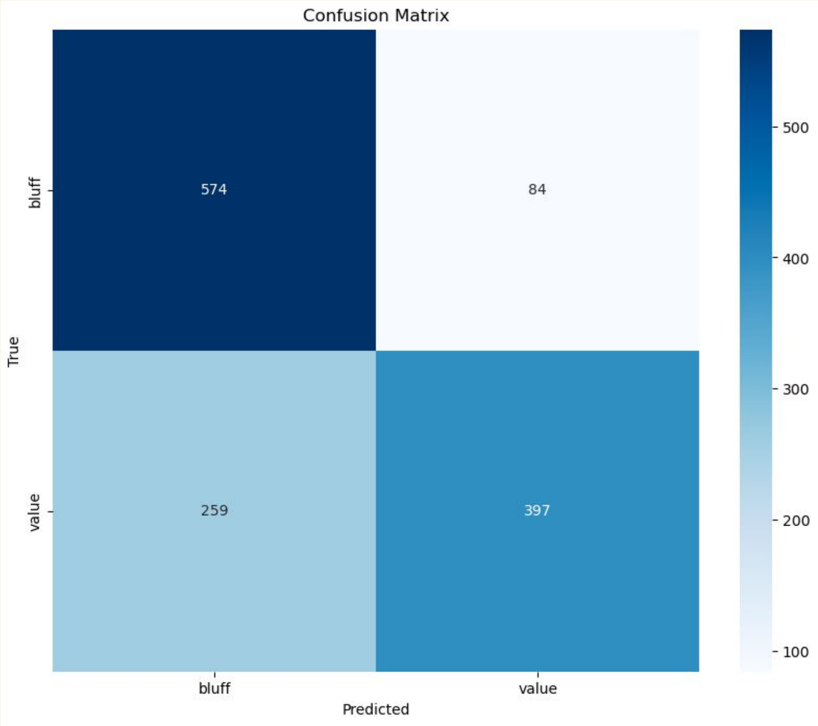
Timing- Only LSTM Results

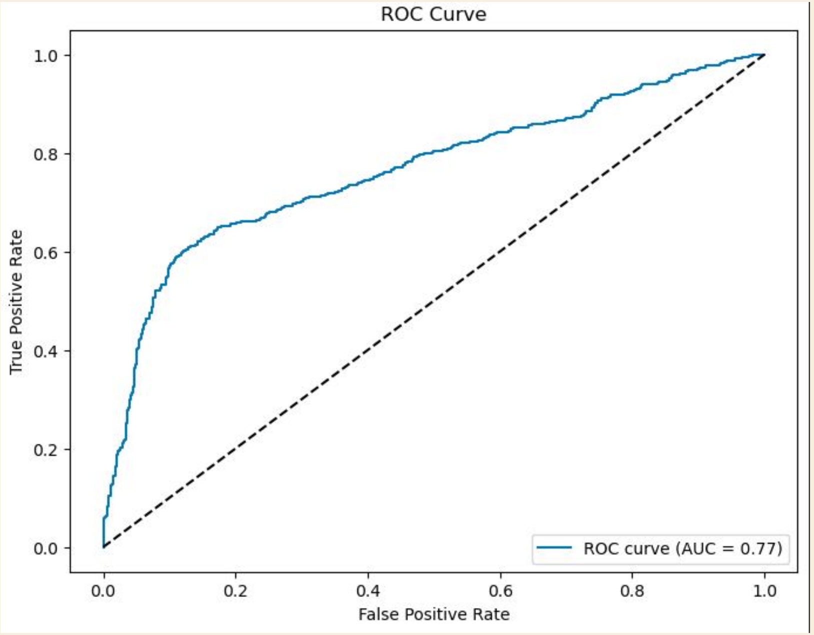
To dive deeper into our primary research question, we evaluated a model based only on timing (using only the decision_time variable)

Test Accuracy: 0.7390

Classification Report:

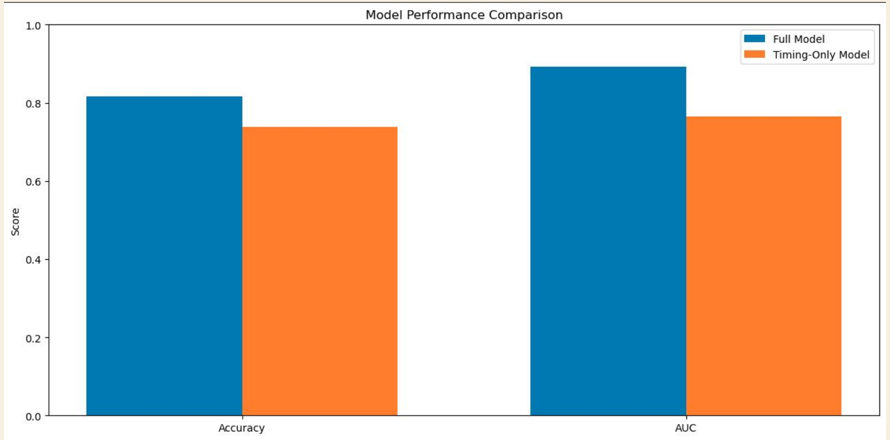
	precision	recall	f1-score	support
bluff	0.69	0.87	0.77	658
value	0.83	0.61	0.70	656
accuracy			0.74	1314
macro avg	0.76	0.74	0.73	1314
weighted avg	0.76	0.74	0.73	1314



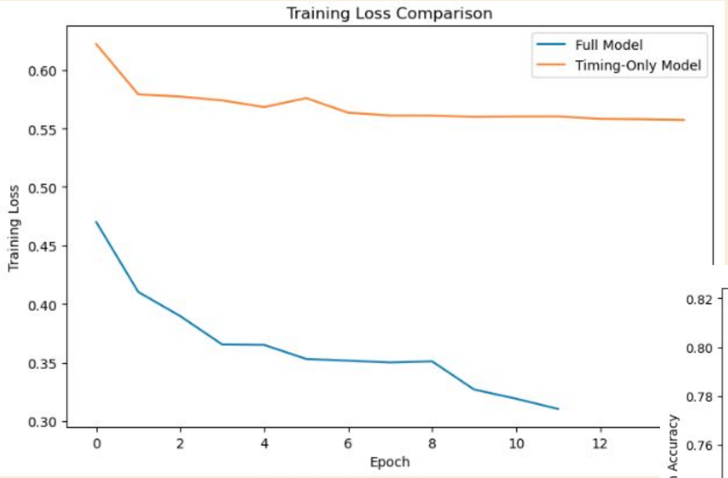


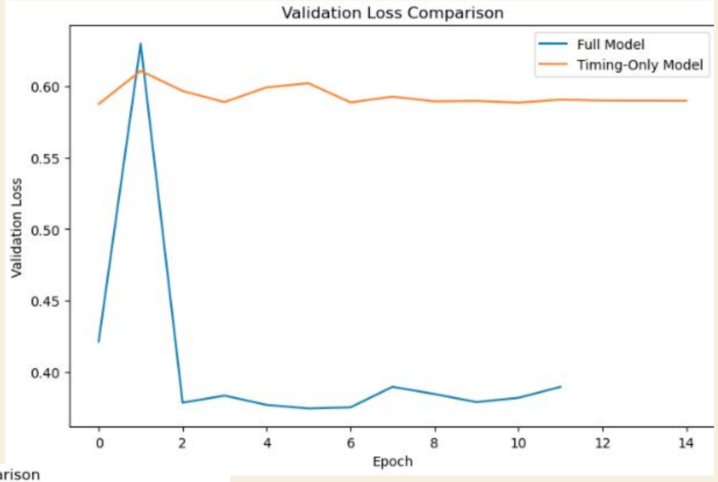
ROC Curve

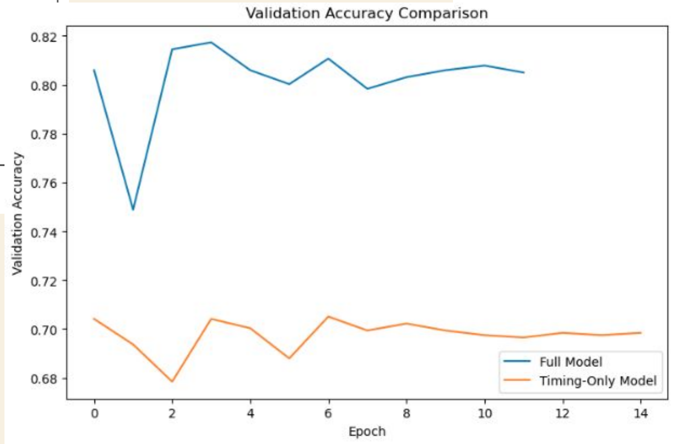
- **Acceptable AUC (0.77)** indicates good separation between positive and negative classes.
- While the full model achieves 0.88 AUC using all features, the timing-only model reaches 0.77 AUC, suggesting that betting timing is a good bluff indicator
- However, incorporating additional poker context is crucial



Comparison


Training Loss Comparison
Full Model
Timing-Only Model
Training Loss
Epoch
0.30
0.35
0.40
0.45
0.50
0.55
0.60
0
2
4
6
8
10
12


Validation Loss Comparison
Full Model
Timing-Only Model
Validation Loss
Epoch
0.40
0.45
0.50
0.55
0.60
0
2
4
6
8
10
12
14


Validation Accuracy Comparison
Full Model
Timing-Only Model
Validation Accuracy
Epoch
0.68
0.70
0.72
0.74
0.76
0.78
0.80
0.82
0
2
4
6
8
10
12
14

Feature Importance Analysis

=== COMPARISON SUMMARY ===
Full model accuracy: 0.8158, AUC: 0.8929
Timing-only model accuracy: 0.7390, AUC: 0.7650

=== ANALYZING TIMING FEATURE IMPORTANCE ===
Timing feature importance: 0.0068

The low timing feature importance score (0.0068) indicates that shuffling the timing features alone had minimal impact on the full model's predictions

This suggests that the full model isn't heavily dependent on timing data in isolation, but rather on **how timing interacts with other features**.

Conclusion

- Overall, the Long Short Term Memory model has proven highly effective at predicting bluffing where other models failed
- Across multiple models, decision time has proven useful in predicting whether of not a player is bluffing
- These result should prove useful both in justifying the inclusion of decision time in future statistical models, and poker simulations.

Thank you!