POPULATION CURATION IN SWARMS: PREDICTING TOP PERFORMERS

A Thesis

presented to

the Faculty of California Polytechnic State University,

California Polytechnic State University, San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Ryan Heller

December 2018

COMMITTEE MEMBERSHIP

TITLE:	Population Curation in Swarms: Predicting Top Performers
AUTHOR:	Ryan Heller
DATE SUBMITTED:	December 2018
COMMITTEE CHAIR:	Franz Kurfess, Ph.D. Professor of Computer Science
COMMITTEE MEMBER:	Lubomir Stanchev, Ph.D. Associate Professor of Computer Science
COMMITTEE MEMBER:	Hisham Assal, Ph.D. Lecturer of Computer Science

ABSTRACT

Population Curation in Swarms: Predicting Top Performers

Ryan Heller

In recent years, new Artificial Intelligence technologies have mimicked examples of collective intelligence occurring in the natural world including flocks of birds, schools of fish, and swarms of bees. One company in particular, Unanimous AI, built a platform (UNU Swarm) that enables a group of humans to make decisions as a single mind by forming a real-time closed loop feedback system for individuals. This platform has proven the ability to amplify the predictive ability of groups of humans in realms including sports, medicine, politics, finance, and entertainment. Previous research has demonstrated it is possible to further enhance knowledge accumulation within a crowd through curation and bias methods applied to individuals in the crowd.

This study explores the efficacy of applying a machine learning pipeline to identify the top performing individuals in the crowd based on a structural profile of survey responses. The ultimate goal is to select these users as Swarm participants to improve the accuracy of the overall system. Unanimous AI provided 24 weeks of survey data collection consisting of 1,139 users from the NHL 2017-2018 season. By applying a machine learning pipeline, this study able to curate a crowd consisting of users that had an average z-score 0.309 and Wisdom of the Crowd accuracy of 61.5%, which is 4.1% higher than a randomly selected crowd and 1.4% lower than Vegas favorite picks.

ACKNOWLEDGMENTS

Thank you to :

- My advisor, Dr. Kurfess for his flexibility, patience, and endless advice over the past two years.
- My committee members, Dr. Stanchev and Dr. Assal for their advice and knowledge in and out of the classroom
- Dr. Kearns and Leanne Fiorentino for their open doors, countless answers to questions about anything and everything, and support throughout this process
- Gregg Wilcox and Dr. Louis Rosenberg from Unanimous AI for the opportunity, continuous feedback, and support throughout this project
- My friends and my family for their constant support and encouragement

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Swarm Intelligence is an integral part to the decision making process of various species in nature. Examples include schools of fish, flocks of birds, and swarms of bees. In 1999, the book, *Swarm Intelligence: From Natural to Artificial Systems*, introduced multiple applications of swarm intelligence to technology to solve real-world optimization problems. [2] Group behaviours including self-organization, robustness, and flexibility were observed in ants, bees, and termites with the goals of identifying optimal routes to food sources, top locations for nest building, and task allocation within a colony. Some of these behaviours were successfully simulated in multi-agent systems and existing algorithms, and other promising approaches were introduced if not implemented. Unanimous AI has created an online platform to enable Swarm Intelligence by developing a single real-time closed-loop feedback system within groups of humans. The collective intelligence of human Swarms has been tested with weekly predictions of the outcome of professional sporting matches in leagues such as the National Football League, National Hockey League, English Premier League, National Basketball Association, and Major League Baseball. Swarms on this platform were given the task to predict the outcome English Premier League matches over a period of five weeks. With this task, the crowd was able to achieve a 72% accuracy compared to an individual average accuracy score of 55% for those games. Furthermore, this platform has been used to accurately predict Super Bowl winners, TIME's Person of the Year, and election results among others. Most impressively, a crowd of users on the UNU platform were able to predict the Kentucky Derby superfecta against 540 to 1 odds, without a single user accurately predicting the winners in a preliminary survey.[3]

1.1 Motivation

Artificial Intelligence is an incredibly powerful tool that is continually making giant leaps in progress. This tool has the power to learn and perform tasks more accurately and efficiently than any human previously has in fields including medicine, automotive, retail support, and more. However, most implementations of these systems leave out one vital element, human intelligence. Great progress can be made by including human intelligence in the learning and decision making component of an intelligent system. Human swarms have managed to outperform other intelligent systems, leading to the next logical step of exploring how to enhance the intelligence of human swarms.

1.2 Goals

As previously mentioned, the UNU Swarm platform has shown promising results to amplify human intelligence. The question remains, how can these swarms be made smarter? Initial tests have shown that Swarms of experts outperform Swarms of randomly selected fans. However it is a hard problem to identify the best performers out of a crowd with little information on each individual. Swarm users answer a survey of questions about the weekly match-ups with multiple confidence measures of their predictions. Currently, these answers are only used to filter out users with no relevant knowledge to the prediction topics if they are unable to answer baseline questions about the sport at hand. The first goal of this project is to determine which features of a user survey provide the most relevant information to a user's performance. Using this information, this project strives to create and apply a machine learning pipeline to identify the best performers out of the crowd based solely on survey answers for each week using a relative performance metric. This process should allow us to know

how a user will perform relative to the average user without any prior information about that individuals previous participation in surveys and Swarms. A selection of strong performers could then be placed into an "expert" crowd to analyze the difference in accuracy between a crowd of experts and a crowd of randomly selected fans. This study hypothesizes that a crowd of experts will produce the most accurate predictions and forecasts seen to date that can then be used to create an "expert" Swarm.

Chapter 2

RELATED WORKS

2.1 Group Decision Making

In fields relevant to forecasting, a myriad of opinions can be seen within a group of random individuals and experts alike. Within these groups, uniformity of opinions does not always translate to the accuracy of the response. Scenarios that contain a larger divergence of answers need to be refined down to a single decision. Face to face interactions tend to influenced by the views of the group member who is most involved in the discussion; however, no significant correlation has been found between the influence within a group and expertise in the problem. To help mitigate these effects in group decision making, the RAND corporation developed the Delphi procedures in October 1967. The Delphi procedures were developed with three characteristics in mind: anonymity, controlled feedback, and statistical response from the group.[4] Anonymity reduced the influence of decisive, powerful, or socially dominant individuals, while controlled feedback was designed to reduce noise within the influence between individuals in the group. Statistical measures were included in the feedback for numerical responses to illustrate an individuals response relative to the group. Individuals first respond to the survey, and can offer additional comments to the survey as feedback. Group members then received the overall statistical responses of the group, before responding to the survey again. This process can be repeated for multiple iterations before conforming to a single group decision. This procedure does not push a group toward unanimity; however the study found that the opinions in the group tended to converge and the median response moved in the direction of the true answer. This method was applied to a group of experts at the Bank of Finland

and Financial Supervising Authority of Finland in the field of short-term economic and financial predictions. [6] Two groups of 20 members were formed to make predictions with the experimental group utilizing Delphi procedures with the benchmark group deciding based on a Face to Face meeting. The Delphi group went through three rounds of surveys, and 2 rounds of feedback with anonymized answers. Figure

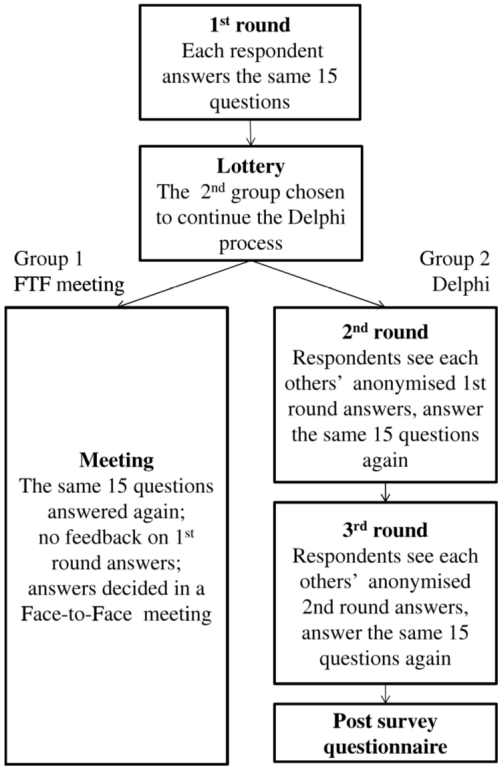

Figure 2.1: Finland Economics and Finance Delphi Process

2.1 illustrates this process. This study found that individuals whose initial predictions were inaccurate tended to adjust toward the correct answer. Furthermore, both methods saw improvement with the resulting group decision relative to the median of initial individual survey responses; however, there was no significant difference in the performance of the group utilizing the Delphi method and the group using face to face discussions. The book, *The Delphi Method, Techniques and Applications*, outlines properties that indicate the need to utilize the Delphi procedures. These

properties include the infeasibility of group meetings based on time, group size, or cost, destructive social influence of individuals within the group, and supplemental group communication to face to face meetings, among others. [8]

2.2 Sports Predictions

Predicting the outcome of matches in a variety of sports is an inherently difficult task due to the uncertainty of events that occur. The article *Sports Forecasting: A Comparison of the Forecast Accuracy of Prediction Markets, Betting Odds, and Tipsters* determined that betting odds are an efficient forecasting instrument based on extensive analyses in economics and business literature. [16] These odds are based on a combination of game outcome probabilities and bet exposure to the bookmakers. This study compared the predictive power of betting odds, prediction markets, and tipster predictions published in newspapers or sports journals. Prediction markets determine the probability of an outcome based on the pricing mechanism of the market, offering an aggregate decision in a competitive market. Based on a dataset consisting of three seasons of predictions and game outcomes in Bundesliga, the German premier soccer league, the difference in accuracy between prediction markets and betting odds was insignificant (0.59%); however, both of these methods outperformed tipster predictions by 11.65% and 11.06%, respectively. [16] Prediction markets demonstrate the ability to crowdsource sports predictions with an accuracy relative to bookmakers odds; however, they do not outperform bookmakers odds. For this reason, bookmakers odds, or Vegas odds, are one of the benchmark accuracies that this study is trying to achieve.

2.3 Artificial Swarm Intelligence

Swarm Intelligence enables numerous species in nature to enhance the intelligence of groups by forming real time closed-loop feedback systems between members. Technologies have been modeled after colonies of ants, flocks of birds, and schools of fish. These ideas has been previously used in the realm of computational intelligence to simulate swarm intelligence with simulated or robotic agents. Two popular methods of this include Ant Colony Optimization and Particle Swarm Optimization.[1] Programs mimicking these types of collective intelligence in nature are commonly referred to as Artificial Swarm Intelligence (ASI). Although these algorithms have offered effective results in various fields, they do not leverage the wealth of knowledge that exists within humans.

However, only recently have platforms become available to effectively combine the power of human intelligence with swarm intelligence.[11] By creating a closed-loop feedback system for humans to interact with, human swarm intelligence tightly connects the individual brains in the group, reducing noise in the group decision making process. Initial research has shown human swarm intelligence predictions have been able to outperform traditional polls and surveys in the financial sector, politics, and sports.[11] Major accomplishments include achieving a 170% Return on Investment (ROI) against Las Vegas bookkeepers over a 20 week period of NHL pick of the week predictions and approximately a 54,000% ROI by accurately predicting the Kentucky Derby Superfecta in 2016.[3][15]

2.4 Crowd Selection

Optimizing the Wisdom of the Crowd effect requires in an depth analysis of the crowd dynamic based on individuals in the crowd. According the Aidan Lyon's paper, The

Wisdom of Crowds: Methods of Human Judgement Aggregation, the organizer should consider the following factors prior to curating a crowd:[9]

- Does your crowd need to consist of experts on some topic? Or can they just all be regular folk?
- How large does your crowd have to be?
- Does your crowd have to be diverse?

These questions relate directly to this experiment. The overall goal aligns directly with the first question to determine how a Swarm of experts compares to a randomly selected Swarm. Although this experiment does not perform any analysis in regards to the size of a Swarm, Wisdom of the Crowd accuracy measures in this experiment compare the accuracy of a group of the top performers to the whole group demonstrating some analysis into the required size of the crowd. Finally, determining whether the solution presented selects individuals with similar betting profiles would decrease the diversity of the crowd. However, the effect of decreasing diversity on the overall accuracy of the crowd is currently unknown. In addition to these questions, one major concern of crowds is that low performers will bring down the average prediction ability of the rest of the crowd.[5] Previous research has shown that the predictions of a whole crowd tend to beat an individual expert; however, randomly selected smaller crowds perform just as well as the whole crowd. Furthermore, the same study demonstrated when experts can be identified and drawn to form a smaller crowd of approximately five members from a pool of twenty-five members show significantly better predictive ability than using the predictive ability of the whole crowd or a randomly selected crowd.[5] Research conducted in this study is based off these principles.

2.5 Individual Bias

Previous studies have achieved success in improving the crowd performance in a variety of tasks in areas including general knowledge, event predictions, and collective estimations with an analysis of individual answers and user profiles.[7][10] A group of researchers at the Department of Cognitive Sciences, University of California Irvine created an experiment in which individuals completed a set of general knowledge and prediction tasks followed by a survey to report their previous experience and knowledge of the topics covered in these tasks.[7] The results of this study demonstrated that inferences about the structure of individual's answers effectively measured the relative expertise of individuals without having the correct answers to the knowledge and predictions tasks. This measure of relative expertise was then used to apply a bias to each individual response that resulted in an improvement in the overall Wisdom of the Crowd. Additionally, two researchers at the oldest neuroscience research center in Spain, Cajal Institute, proved that weighting users based on social influence without historical data outperformed a simple crowd majority for collective estimation tasks.[10] Social influence had previously had a perverse effect on the Wisdom of the Crowd by leading to a more biased and innacurate crowd estimation. However, by building a profile for each user based on participation in social interactions of the crowd and applying a bias that has a negative correlation with social influence improved the estimation power of the crowd as a whole. These two experiments provided great value to this study with respect to creating models that represent individuals, that can then be analyzed to determine the relative expertise of each individual in the crowd.

Chapter 3

BACKGROUND

3.1 Unanimous AI

Unanimous AI is a company based out of San Luis Obispo, California. The overall mission of the company is to amplify intelligence by leveraging both the collective knowledge of a group of humans and advancements in artificial intelligence through the creation of a real-time feedback loop between individuals. Two of their flagship products, Swarm AI and Swarm Insight, mimic examples of Swarm intelligence found in nature to create a platform for human swarming. Groups participating in the Swarm AI platform have predicted sports outcomes, financial trends, and political outcomes with more accuracy than tradional survey methods including polls and prediction markets.

3.1.1 What is a Swarm?

In 2015, Unanimous AI developed an online platform to create a closed feedback loop between a distributed group of users. This platform was originally called UNU, but has been re-branded to Swarm AI according to the Unanimous AI Products page (link). This platform allows all users to work in parallel to weigh competing alternatives and converge on a single solution. An example of this platform at the beginning of a live swarm is shown in Figure 3.1. A group on the Swarm AI platform is given a set of options to make a prediction. Each user on the platform is represented as a magnet that pulls a central puck with a force relative to the distance between each magnet and puck. The sum of all force vectors created by the magnets determines the overall magnitude and direction of the puck, eventually settling over a single

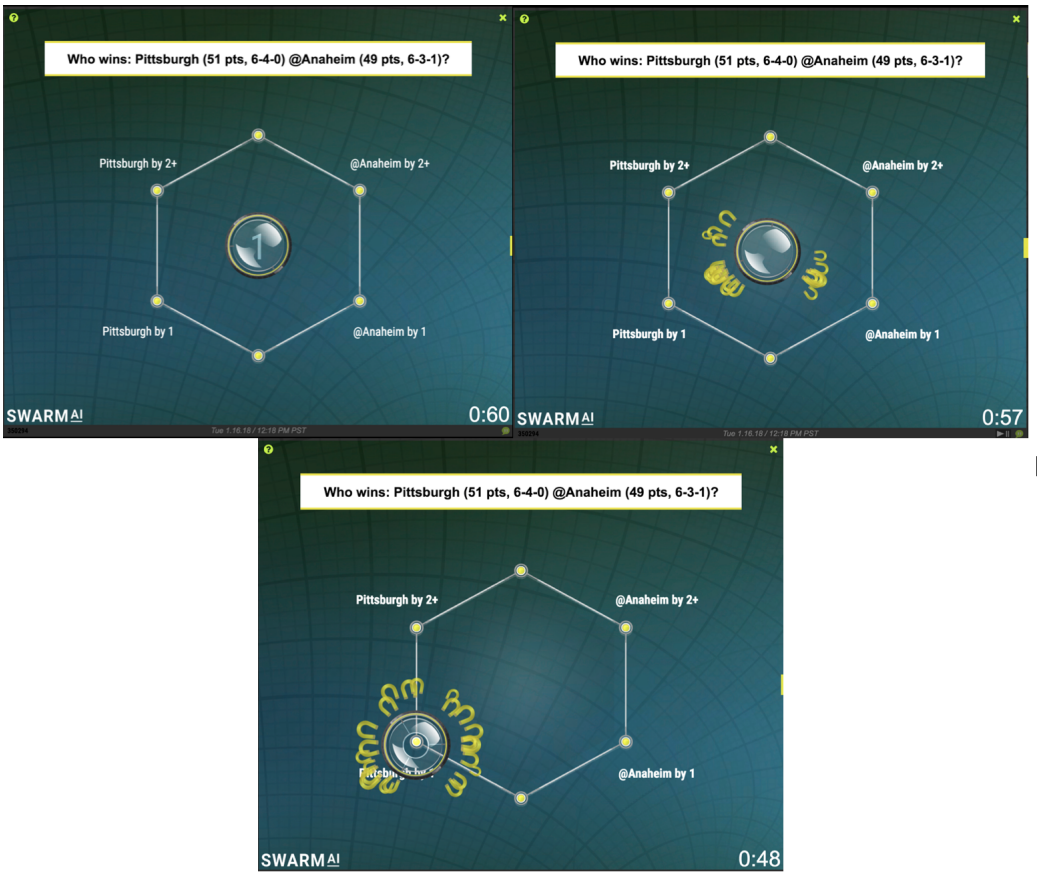

Figure 3.1: Step by Step Live Swarm

selection that represents the groups decision. Members of the group are free to move their magnet at any point throughout the swarm, enabling unspoken negotiations and compromise between all members of the Swarm. Users must actively participate in the swarm as the applied force of each magnet decreases as the distance between the magnet and the central puck increase.

3.1.2 Swarm Selection

Currently, Swarm AI users are selected from Unanimous AI followers and Amazon Mechanical Turk users. All members of each of these groups are asked to take a filter quiz that asks very basic questions about the topic of choice that a majority of fans

would know, such as "Who won the last Super Bowl?" for NFL prediction Swarms or "Who won the last Stanley Cup?" for NHL prediction Swarms. Users who are able to answer these basic questions are invited to take a survey to make the same predictions individually as the group will make later in the Swarm. Approximately 25-35 users that complete this step are invited to participate in the Swarm. Figure 3.2 illustrates the previous process for Swarm user selection.

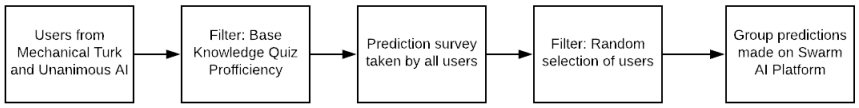

Figure 3.2: Previous Process for Swarm User Selection

This project proposes another step to filter users before entering the Swarm. Based on the predictions and confidence levels of survey responses, the structure and trends of a users survey responses will be analyzed to determine whether or not they are likely to be a top performer. From this predictive analysis, this study proposes that it would be beneficial to select only a percentage of the predicted top users to join the Swarm to improve the accuracy of the groups decision as a whole.

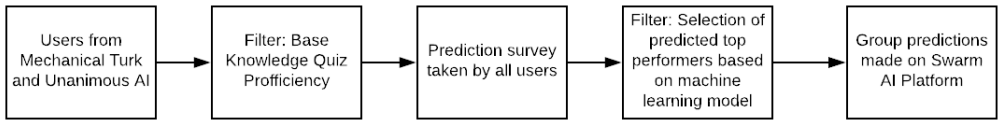

Figure 3.3: Proposed Process for Swarm User Selection

Figure 3.3 illustrates the new proposed process for selecting users to participate in the swarm.

3.2 Artificial Neural Network

Artificial Neural Networks are computational models derived from the functional processing of neural networks within a human brain. The smallest unit of a neural network is referred to as a neuron, or a node. A node takes in a number of inputs with weights from nodes in previous layers, and outputs the result of an activation function applied to the weighted sum of inputs. In a fully connected neural network, a neuron sends its output to all of the neurons in the next layer of the neural network. A model of a single node can be seen in the image below.

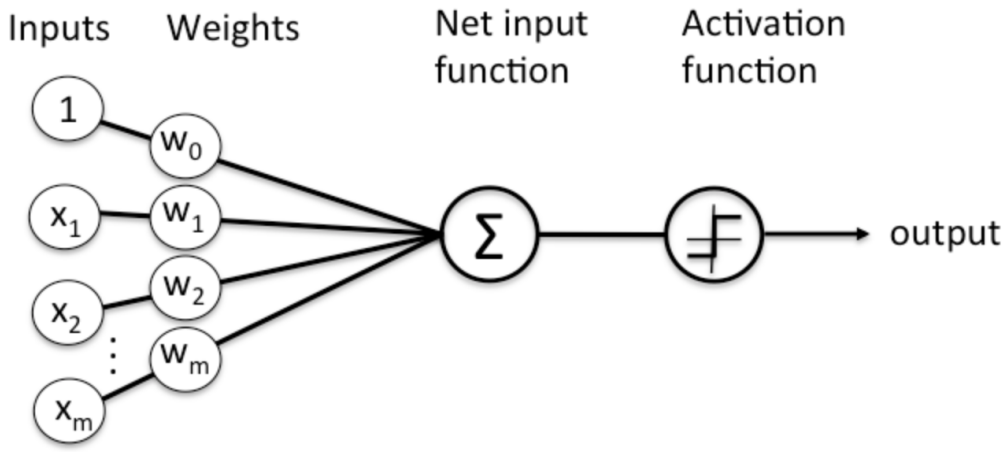

Figure 3.4: Diagram of a Single Node

Activation functions perform a certain mathematical operation on the sum of weighted inputs to derive the output of each node. Most activation functions in practice are non-linear, as neurons need to be trained to recognize patterns most commonly seen in real world data. Example activation functions can be seen in 3.5. Tanh, ReLU, Linear, and Leaky ReLU were the most common activation functions used in this experiment.

Artificial Neural Networks are composed of three types of layers: input, hidden, and output. Each layer is composed of a number of nodes. Nodes in the input layer

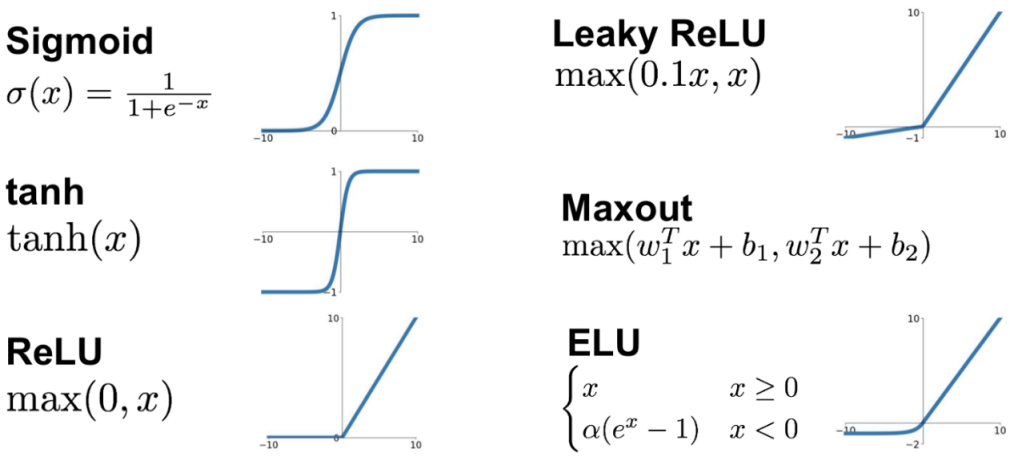

Figure 3.5: Commonly Used Activation Functions

pass real world data to nodes in the first hidden layer without performing a computation on that data. Nodes in the hidden layer take information from the input nodes or previous hidden layer nodes, perform a computation on that data, and send those computations to the next hidden layer or the output layer. Output layer nodes perform computations to transform the data from the network back to the context of the real world data. Only one input layer and one output layer may exist in the Artificial Neural Networks used in this experiment; however, the number of hidden layers may vary. This is the basic network architecture that goes by the names feedforward neural networks or multi-layer perceptrons due to the unidirectional information flow. Neural networks learn through a feedback process called backpropagation. This compares the output of the neural network with the actual value it should have produced. A loss function is used to measure the error of the output, and this error is fed backwards through the neural network from the output layer through hidden layers and back to the input layer to modify the weights of connections between nodes in the neural network. Weights will be adjusted during each iteration of backpropagation. This should improve the accuracy of the predictive model, unless the model succumbs to overfitting. Supervised learning is training a machine learning model with data

that has a label, or actual output attached to each data point. This makes backpropagation useful because there is an expected outcome that can be measured against the predicted outcome of the model.

3.3 Cross-Validation

Cross-validation is a model validation technique that improves the estimation of how accurate a machine learning model is predicted to perform in practice by mitigating the effects of overfitting. For each round of cross validation, data is split into two sets - a training set and a test set. Packages generally exist in machine learning libraries to support cross-validation; however, this module was manually implemented due to the data structures of the weekly surveys in a season. Each week represents the test set for one round of cross validation in this experiment, while the other 24 weeks are used for training. With 25-fold cross-validation, the assumption is made that the results offer an accurate estimate of the machine learning model performance.

3.4 Tools

3.4.1 SciPy

SciPy is an open-source software library providing functions for mathematics, science, and engineering. SciPy.stats is the main module used in this library for feature preprocessing with Gaussian Kernel Density Estimation and statistical testing to analyze the results of each of our experiments.

3.4.2 Scikit-Learn

scikit-learn is an open-source machine learning library in Python. For a large part of the feature engineering and feature selection process, the preprocessing module within

scikit-learn was used. Preprocessing provides functions to normalize, scale, binarize, and encode categorical data. Additionally scikit-learn provides a module with various feature selection algorithms that includes the SelectKBest method utilized in this experiment.

3.4.3 Gaussian Kernel Density Estimation

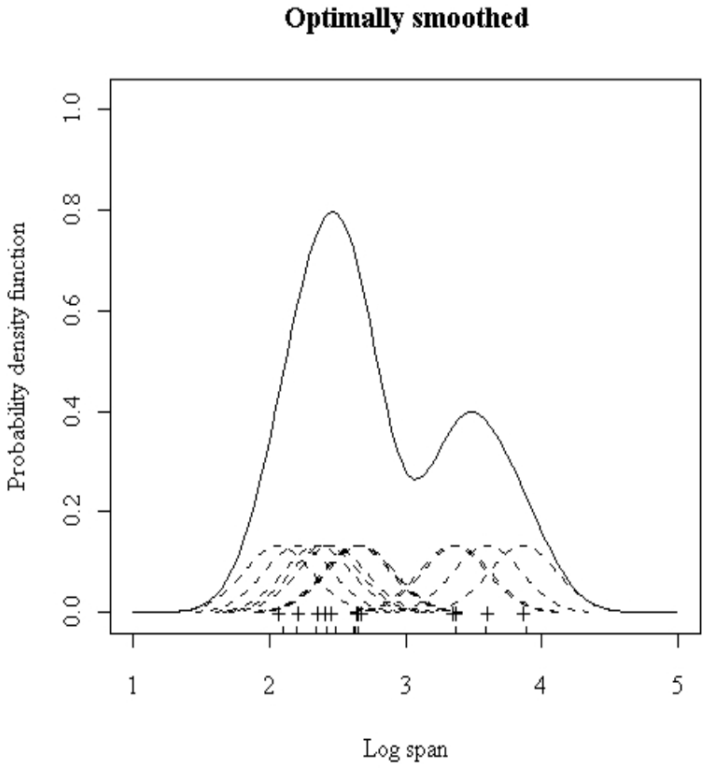

Figure 3.6: Example of Gaussian Kernel Density Estimation

Kernel density estimations (KDE) are methods to estimate a probability density function of a variable. A Gaussian Kernel Density Estimation captures a smoother distribution as each point on the graph contributes a Gaussian curve to the overall kernel. Figure 3.6 illustrates this. Data points are represented as a "plus" symbol, The Gaussian contribution for each point to the overall kernel is represented by the dashed curves, and the resulting kernel is displayed by the solid curve. This method

offered a better representation of the data used in this experiment than more rigid kernel density estimations, such as a tophat kernel density estimation illustrated in Figure 3.7.

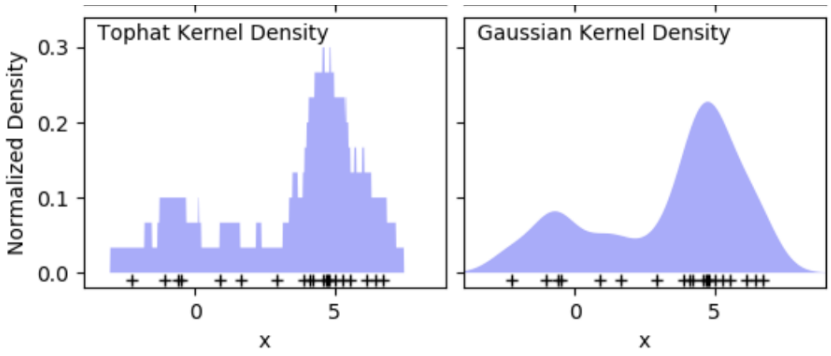

Figure 3.7: Kernel Density Estimation Comparison

3.4.4 Keras

Keras is an open-source Python library designed to enable quick prototypes of deep neural networks. It has the capability to be run on popular deep-learning libraries including Tensorflow, Microsoft Cognitive Toolkit, and Theano. Tensorflow was used as the Keras backend library for this experiment. Outside of the ability for rapid prototyping, Keras was selected for the ability to support GPU-accelerated computing, as well as the modularity with cost functions, optimizers, and activation functions for the neural network architecture.

3.4.5 Hyperopt

Hyperopt is an open-source Python library to perform distributed asynchronous hyper-parameter optimization over search spaces with real-valued, discrete, and conditional dimensions. Users define an objective function to minimize, a search space for parameters, and the search algorithm to do. Search spaces may include num-

ber of nodes per layer, number of layers, activation function, epochs, learning rate, optimization algorithm, cost function, among others for machine learning problems similar to this. Currently, hyperopt only implements two search algorithms, Tree of Parzen Estimators (TPE) and Random Search, but can accommodate other Bayesian optimization algorithms. Random search was the hyper-paremeter search algorithm for this experiment.

3.4.6 Amazon Web Services

Amazon Elastic Compute Cloud is a service within Amazon Web Services that offers customizeable cloud compute instances. This service includes GPU-compute instances in the P2 tier well-suited for machine learning. Amazon EC2 P2 instances contain 1 NVIDIA K80 GPU, 4 virtual CPUs, and 61 GiB of RAM. This infrastructure allowed multiple features and models to be trained simultaneously at a faster average training time then a personal computer. Each Amazon EC2 P2.xlarge instance is priced at $0.90 per Hour; however, spot instances for unused infrastructure were rented for an average price of $0.271per Hour. This component allowed for faster iterations in this project.

Chapter 4

EXPERIMENTAL DESIGN

This chapter explains the experimental design for this experiment. Major components of this design include preprocessing of data, feature engineering, neural network model creation, training and validation, and evaluation.

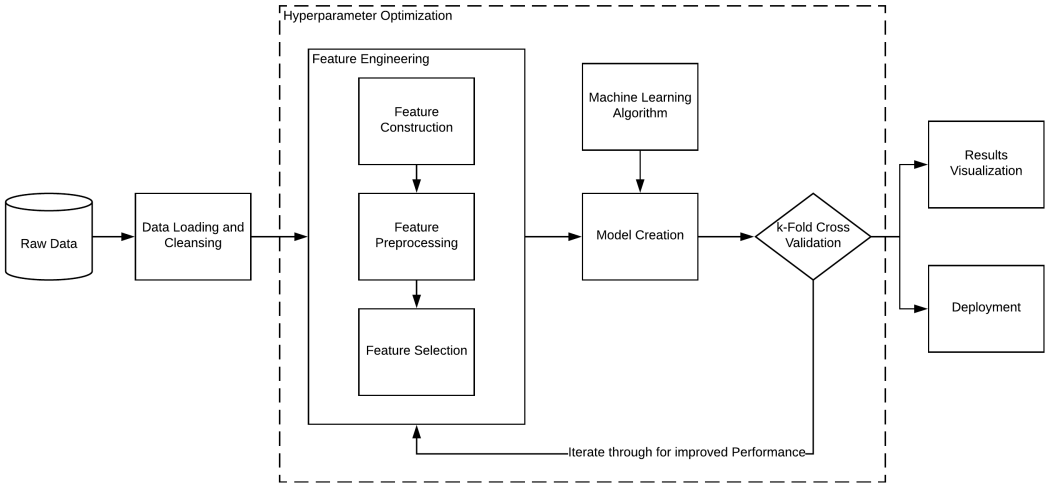

Figure 4.1: Machine Learning Pipeline

4.1 Data Cleansing

4.1.1 Data Extraction

The data was formatted in a folder of Microsoft Excel files representing each week. Each file consists of two Microsoft Excel worksheets - one containing individual users' survey data and the other containing group swarm data. The survey data worksheet contains all user data in this experiment. Vegas Odds for each game were gathered from the Swarm results worksheets. Each row represents a single user's survey response in the "Scored" or "Survey" spreadsheet. The data in Table 4.1 was collected

Column Names and Features	
Worker ID	Agent Id
For each game in a week:	
Who will win: Team 1 @ Team 2?	Predicted Winner,Goal Confidence
What percentage of the people taking this survey will pick the same winner you did?	Predicted Support
You can bet $0 to $100 that the team you selected will win (regardless of number of goals). How much would you bet?	Dollar Confidence
Actual Result	Correct
Earned/Lost	ROI
Aggregate statistics for the week	
Total Score	Accuracy
Total Earned/Lost	Total ROI
ROI	Total Percent ROI

Table 4.1: Microsoft Excel Worksheet Column Names and Corresponding Features

for each user.

4.1.2 Data Organization

Data from each week is represented with a Week class. Two key parts to this class are the list of agents represented by "Worker ID" and a dictionary of dictionaries named "responses" to represent all survey responses for each week. The following diagram illustrates the relationship between Seasons, Weeks, and "responses". Each

dictionary in the responses class maps an agent to the value of each specified feature for that user.

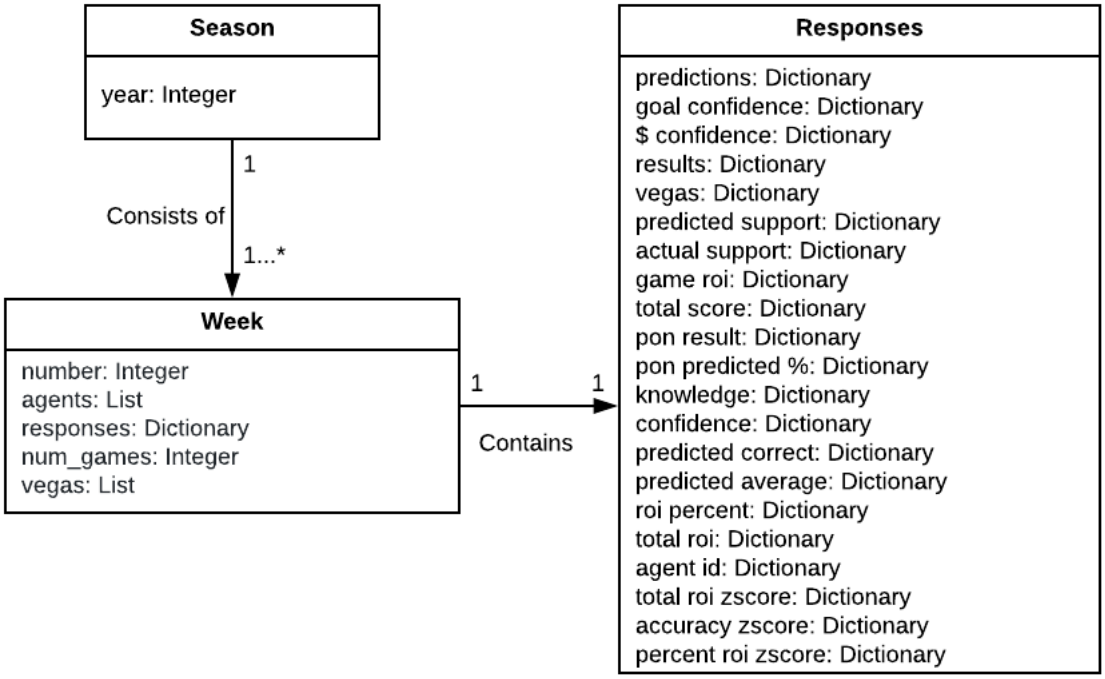

Figure 4.2: Data Structure Diagram

All features in this vector are normalized to values between 0 and 1 to ensure that no single feature inadvertently has a stronger weight on the model. There are three different labels given to each agent at this stage: Return on Investment Percentage Z-Score, Raw Return on Investment Z-Score, and Wisdom of the Crowd Z-Score. A Z-score defines the performance of a single agent relative to the average agent in the crowd for each week. The formula for a Z-score is:

$$z = \frac{x + \mu}{\sigma}$$

4.2 Feature Engineering

4.2.1 Feature Construction and Preprocessing

The goal for this component of the experiment is to identify relationships between multiple variables to create new features that increase the accuracy of the prediction of an agents performance that week. The raw data for each agent consists of answers to survey data including goal confidence, dollar confidence, and predicted support. To add more value, the "Feature Construction" step is utilized to build out new features that add value to our model. Features introduced in this process include actual support, vegas odds, and various correlations between existing features of this data. At this point, the feature preprocessing engine manipulates the data in multiple ways. For both single feature and feature correlation data vectors that consist of a betting strategy for a whole week, a Gaussian Kernel Density Estimation (KDE) is applied to a range between minimum and maximum of a feature. Each Gaussian KDE is sampled at standard intervals to create a new feature vector. This process has two advantages: the length of feature vectors are standardized week to week as the number of predicted games varies between 8 and 12, and the kernel density estimation represents the agents holistic betting trend better than the original raw feature vectors. At this point, the feature vector is normalized between 0 and 1 to ensure that no features have superior weighting at the initiation of the model.

4.2.2 Feature Selection

Feature selection reduces the dimensions of the feature to decrease the noise of data fed to the ML model. Two strategies were tested in this process: Principal Component Analysis (PCA) and Select K Best. PCA is a feature extraction technique that creates a new reduced feature vector with independent variables created with correlations

from the existing variables. Select K Best is a feature elimination technique that selects the features that have the highest correlation with the target variable. Both of these transformation techniques are created based on training data only, and applied to validation data. Select K Best was ultimately chosen as the final technique for experimentation as it handled extensive noisy data better than PCA; however, this decision could also be included in the hyperparameter optimization wrapper of our program.

4.3 Neural Network Model Creation

To experiment with our feature selection process, a basic Sequential neural network was used. This network consisted of an input layer containing 15 nodes, 3 hidden layers with tanh and relu activation functions, and an output layer with a linear activation function. A mean average error loss function correlated directly with the goal of selecting the best users without skewing the model due to incorrect outlier predictions. After getting basic results with this network, hyperparameter optimization was used to create an architecture that minimized error within the model. This network is illustrated in Chapter 5.

4.4 Training and Validation

To validate the results of each iteration through the machine learning pipeline process, k-fold cross validation is used. This process partitions the data into k different sets, in this case 25 sets with each week representing a set. There will be k iterations of training and testing the machine learning model. In each iteration, the training set consists of k-1 sets and the leftover set is used as a test set to validate the accuracy of the model. The test set is rotated for each iteration. The main advantage of using k-fold cross validation to test the accuracy of the model is to mitigate the effects of

over fitting based on training data, and reduce the possibility of selecting a training and testing set that has better than average accuracy.

4.5 Evaluation Criteria

The evaluation criteria for this project were created to evaluate the efficacy of the features and trained model specifically relevant to the goals of this project of selecting a crowd of top performers and determining if that crowd will perform better than a randomly selected crowd. The first is a comparison of the average z-score of top users curated based on the model to the average z-score of randomly selected users. This criteria demonstrates that our features and models are indeed selecting users that perform better than average in general. The next step is to compare the Wisdom of the Crowd accuracy for a curated crowd against the accuracy of a randomly selected crowd. The Wisdom of the Crowd Accuracy is based on the percentage of correct predictions that the majority of the crowd voted for in the survey. Although this does not necessarily prove that this crowd will perform better in a Swarm, this does prove that the curated crowd has better performance prior to entering the Swarm, on average. Along with this, a betting strategy is created for each crowd based on the bets made by individual users in each crowd. This develops a Wisdom of the Crowd Return on Investment criteria that enables us to calculate and compare the average ROI per game for a curated swarm against a randomly selected swarm.

4.6 Hyperparameter Optimization

Hyperparameter optimization acts as a wrapper for our feature engineering and model creation process. This component tunes the hyperparemeters of the overall learning model to minimize the predefined loss function for the package. Hyperparemeters included in this optimization consist of the size of the feature vector, number of

layers, number of nodes for each layer, activation functions for each layer, number of epochs, and the optimization function of the model. The loss function used for this component was the mean average error of the predicted z-score to the actual z-score of users. This loss function was selected over evaluation criteria relating to Wisdom of the Crowd because it aligns directly with the primary goal of identifying the top performers of a crowd.

4.7 Command Line Interface

Testing different features in the early stages of this project was time consuming and troublesome. The script to run the program was manually changed, which seemed to be ineffective as the evaluation criteria and resulting visualizations morphed throughout the project. For this reason, a simple command line interface was created to run different feature combinations with ease. Steps for this interface are shown in the figures below.

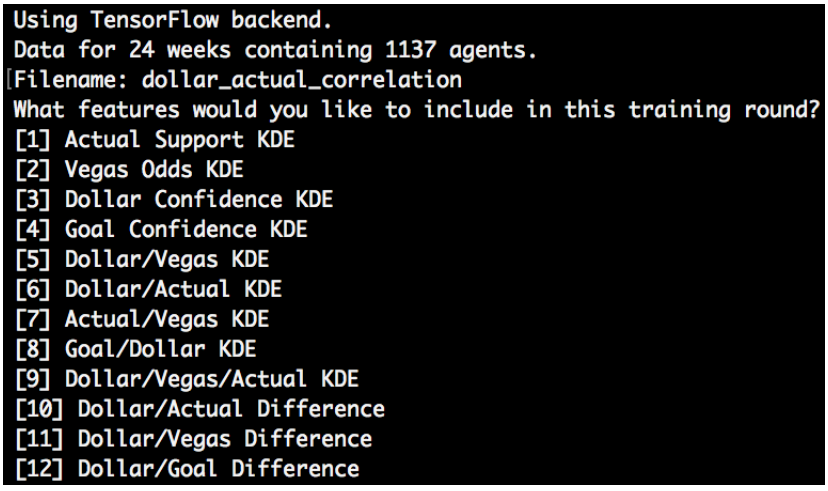

Figure 4.3: Command Line Interface Options

Starting the Python script displays the number of weeks and total number of agents to the user. The user is prompted for a filename that serves as a prefix to

all result visualizations and statistic log files. Features available in the program are displayed to the user before prompting the user for input as to which features the user would like to train a model on.

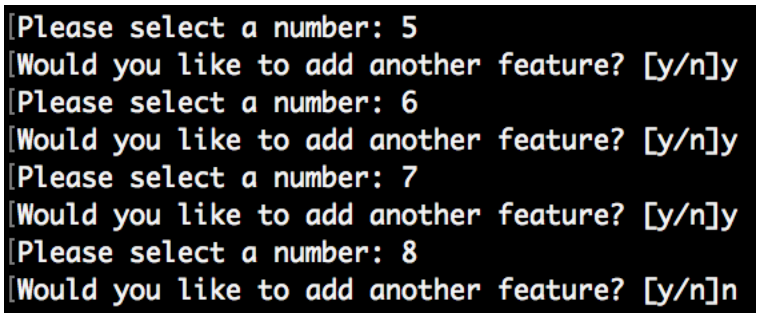

Figure 4.4: Selecting Features with Command Line Interface

The user is able to select anywhere from one feature to all features available. When the user is done with their selection, the feature extraction, model training, and validation begins. Resulting visualizations and statistics are saved to a local directory with names including the prefix filename inputted by the user. Currently, the command line interface only supports the filename and feature selection, but could simply be extended for model parameters including learning rate, epochs, batch size, and more.

Chapter 5

EVALUATION

5.1 Single Feature KDE

Models trained using Gaussian kernel density estimations of single features were the foundational step for this project. Each model gave insight as to how useful each component alone might be with respect to predicting the overall performance of a user; however, if a feature was not valuable at this stage, this does not mean that the feature will not prove to be valuable when correlated with other features. Kernel density estimations demonstrated users' trends for a single week rather than simply using raw data collected from the surveys. Additionally, this standardizes the length of the feature vector from week to week, as the number of predicted games per week varied between 8 and 12.

5.1.1 Actual Support

Actual Support is the calculated percentage of users that predicted the same team to win as that individual for each game.

Legend Description	
Ideal	Actual Average Z-Score based on data
Predicted	Average Z-Score based on predicted top performers
Random	Average Z-Score for randomly selected users
Curated WOC	Z-Score for an agent representing the Crowd

Table 5.1: Average Z-Score Graph Legend Descriptions

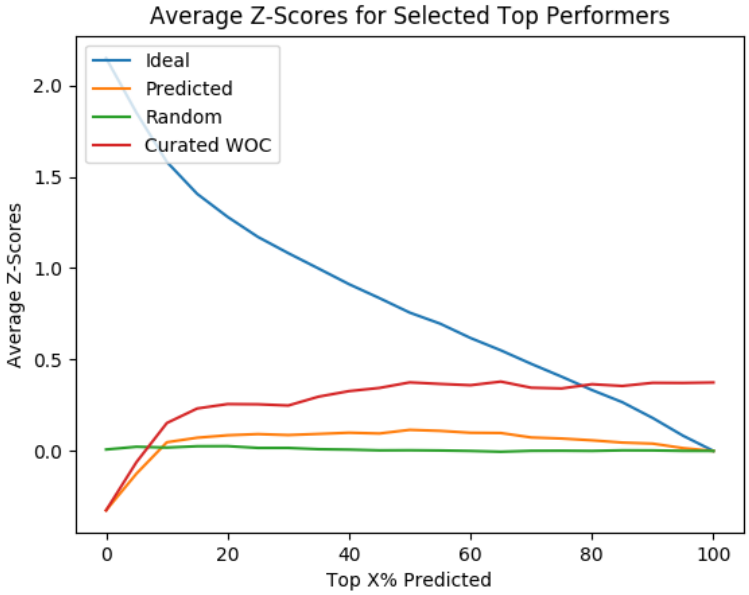

Figure 5.1: Average Z-Score for Top Performing Agents Curated by Actual Support

Figure 5.1 illustrates the average z-score (based on ROI Percentage) of the top percentile of users displayed on the x-axis value. Four different categories are illustrated with descriptions for each line in Table 5.1.

As can be seen in Figure 5.1, there is minimal differentiation between the group of randomly selected agents and agents curated based on their Actual Support score profile for each week. It is notable that the line depicting the average z-score for

predicted top performers starts at a z-score of -0.32. In other words, the top predicted user has an average z-score of about -0.32 demonstrating that the basic model is exceptionally bad at predicting the top user. Selecting the predicted top 50 percent of the crowd gives us an average z-score of 0.11 better than a crowd of randomly selected users of the same size. Previously, Actual Support score has shown to be a great predictor of agent success in Unanimous AI data sets for other sports; however, the results for the NHL 2017-2018 season show us that the Actual Support score is not as valuable for this group of agents. It is worth noting that the dynamics of the crowd have shown to vary from sport to sport, and this result may not hold true for all groups entering a sports prediction swarm.

5.1.2 Goal Confidence

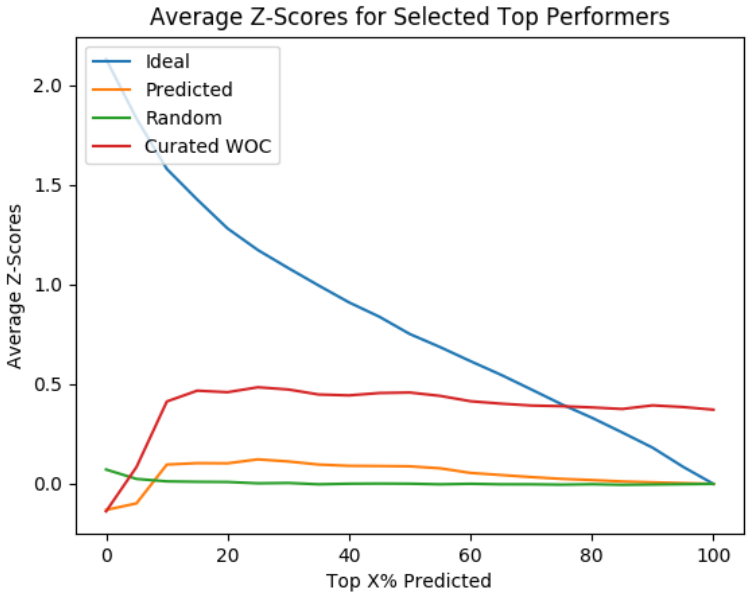

Figure 5.2: Average Z-Score for Top Performing Agents Curated by Goal Confidence

Goal Confidence alone has proven to be a mostly ineffective metric in predicting the success of an agent when used alone as seen in the results in Figure 5.2. This model

had a similar issue with respect to unexpectedly picking a poor top performer. Based on this model, taking the predicted top 25% of users yields an average z-score of 0.12 better than a randomly selected crowd of the same size. These results are very similar to the Actual Support Metric, and may provide more value when applied in correlation with other features. A likely reason for the poor performance for this as a single feature is that a user predicts between 10-12 games, with each game having two possibilities for Goal Confidence (1 or 2+). This strictly limits the information that can be interpreted solely based on their Goal Confidence for matches that week. In NHL game predictions, increasing the number of possibilities for this category will likely not be effective as most games end with a small goal differential; however, implementing a larger number of possibilities in leagues such as the NBA or the NFL where point differentials are larger is likely to be more effective. Making the assumption that Goal Confidence does not give us any information would be invalid, as correlations between Goal Confidence and other features are likely to prove to be more valuable.

5.1.3 Dollar Confidence

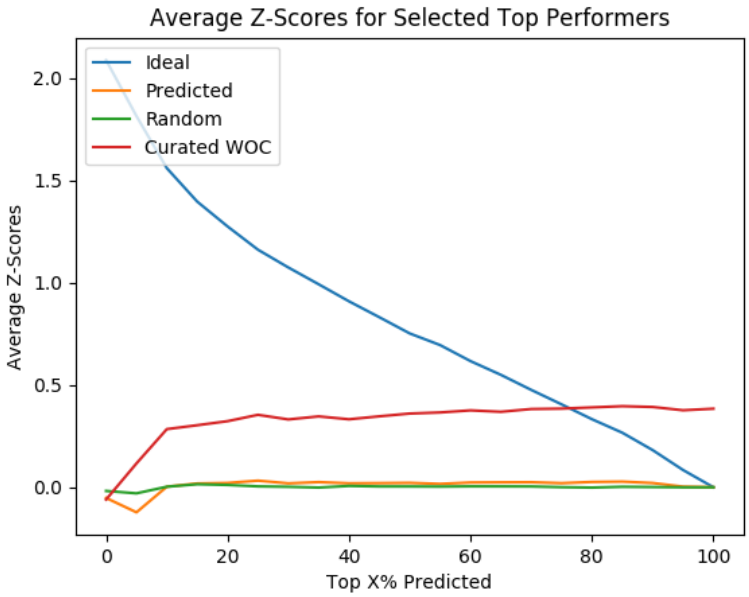

Figure 5.3: Average Z-Score for Top Performing Agents Curated by Dollar Confidence

Using heuristics early in this experiment, Dollar Confidence was predicted to be the second most valuable feature in predicting a users performance. This proved not to be true when examining Figure 5.3. Unlike Actual Support and Goal Confidence, Dollar Confidence does not succumb to the same issue of selecting top performers with an average z-score of below 0. Predicted users average a z-score of around 0 regardless of the size of the crowd, aligning with the trend for a randomly selected crowd. Dollar Confidence data tends to have the highest variability from one user to the next, likely correlating with a high volume of noise within that data. Dollar Confidence alone does not provide great value in selecting a crowd.

5.1.4 Vegas Odds

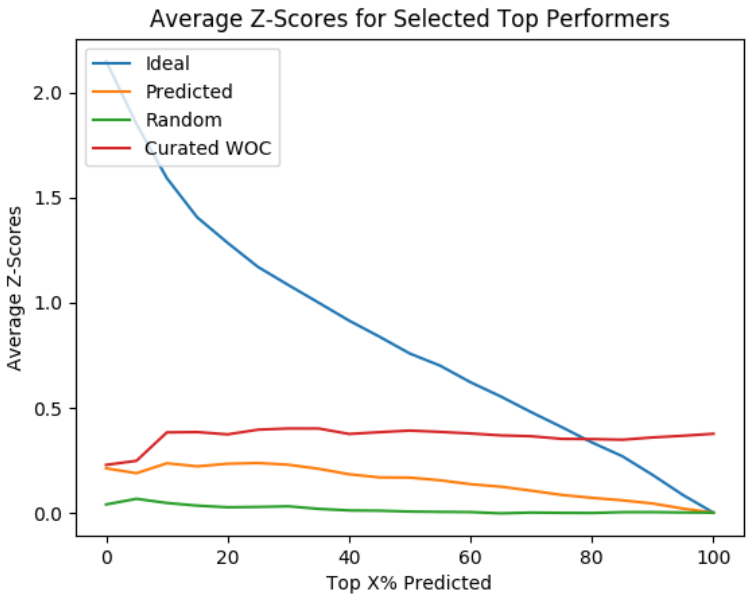

Figure 5.4: Average Z-Score for Top Performing Agents Curated by Vegas Odds of Predicted Winners

Vegas Odds were not a raw feature of the survey data. Swarm data worksheets contained the Vegas Odds formatted as moneyline odds for the value of a potential bet, and these were translated into implied probabilities as percentages using the following equations.

If moneyline odds are negative:

$$VegasOdds = \frac{-MoneylineOdds}{-MoneylineOdds + 100}$$

If moneyline odds are positve:

$$VegasOdds = \frac{100}{MoneylineOdds + 100}$$

Vegas bookkeepers release among the best probabilities for sports predictions, leading

to the belief that correlating the Vegas Odds of each user's predicted winners will be valuable in selecting the top performers of a group. Vegas Odds proved to be the best solo feature as seen in Figure 5.4. The average z-score for users in the 20th percentile shattered other solo features with an average z-score of 0.238. This illustrates that correlations between Vegas Odds and the users predictions is a substantial predictor of success for users. Moving forward, analyzing betting strategies relative to Vegas Odds is demonstrated to be effective later in this chapter.

5.1.5 Combination of All Single Features

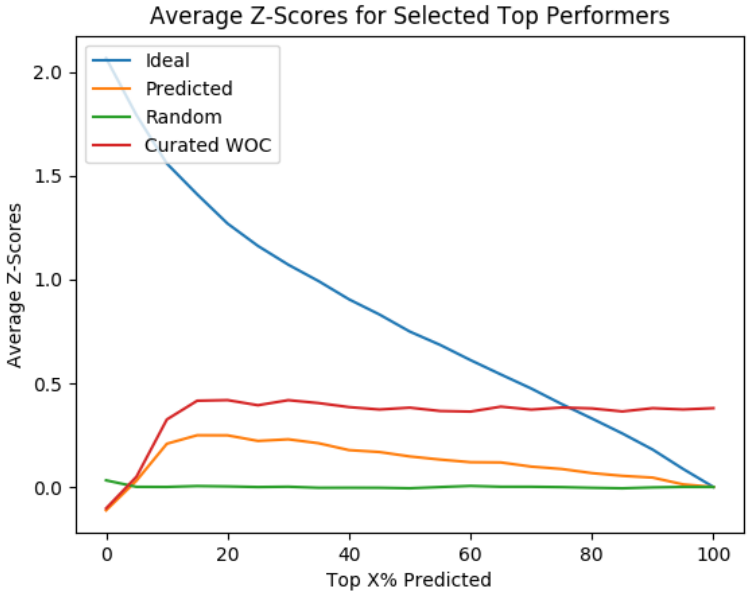

Figure 5.5: Average Z-Score for Top Performing Agents Curated by Combination of All Single Feature

Creating a feature vector that consisted of all four solo feature kernel density estimations offered the results above. As expected, the combination of all single features outperformed each of the single features. At the peak average z-score, the top 15% of users averaged a z-score of approximately 0.250, beating the Vegas Odds model by only 0.012. An interesting note with this model follows the trends of the Actual

Support and Goal Confidence, with the top user selected generally performing worse than average in the crowd with an average z-score of -0.112. This attribute drastically affects the overall performance for the low percentiles, drastically reducing the efficacy of this model.

5.1.6 Dollar Confidence and Vegas Odds

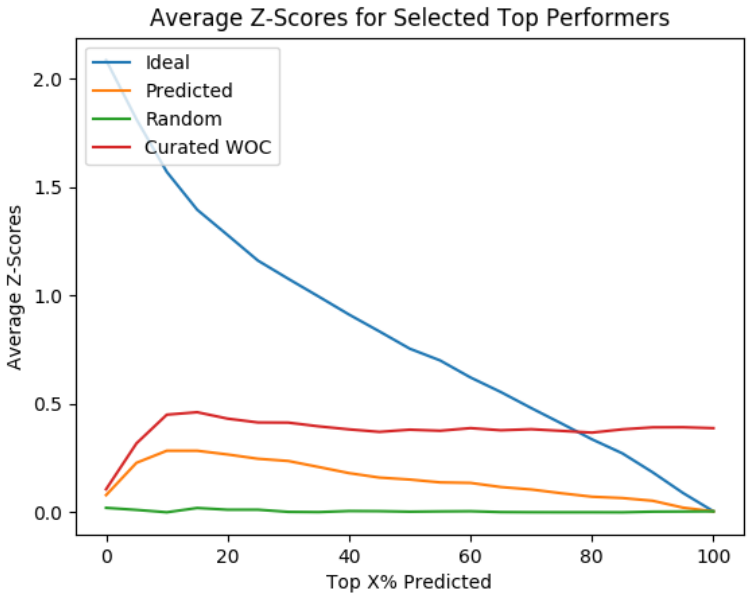

Figure 5.6: Average Z-Score for Top Performing Agents Curated by Combination of Vegas Odds and Dollar Confidence

Section 5.1.5 proposed that a combination of all single feature kernel density estimations had the top performance to that point, with the observation that this model still succumbed to the pitfalls of poor top predicted performers in the Goal Confidence and Actual Support models. Dollar Confidence combined with Vegas Odds outperformed the combination of all single features without falling to issues of selecting poor users as top performers. The average z-score for users in the top 15th percentile was approximately 0.283, or 0.033 higher than a combination of all single feature kernel density estimations.

Summary of Single Feature Results		
Feature	Top Percentile	Average Z-Score
Actual Support	50	0.115
Goal Confidence	25	0.121
Dollar Confidence	25	0.033
Vegas Odds	25	0.238
Combination of All Single	15	0.250
Dollar Confidence and Vegas Odds	15	0.283

Table 5.2: Summary of Single Feature Results

5.1.7 Summary Single Feature Findings

Table 5.2 shows a summary of the highest average z-scores for the top predicted performers based on models trained on each of the single feature kernel density estimations. Vegas Odds significantly outperformed every other solo feature, with the combination of all single feature kernel density estimations performing only slightly better than Vegas Odds alone. Dollar Confidence was the most surprising, only offering a slightly better average than random selection. These findings give a good foundational knowledge of feature efficacy, paving way for features engineered and tested later in this chapter.

5.2 Feature Correlations

Section 5.1 stated that some features alone did not have the predictive power desired for this experiment, but that correlations between these features may provide more value. To achieve this, the feature vector consisted of a two-dimensional Gaussian

Kernel Density sampled at every 0.01 by 0.01 block for each feature correlation.

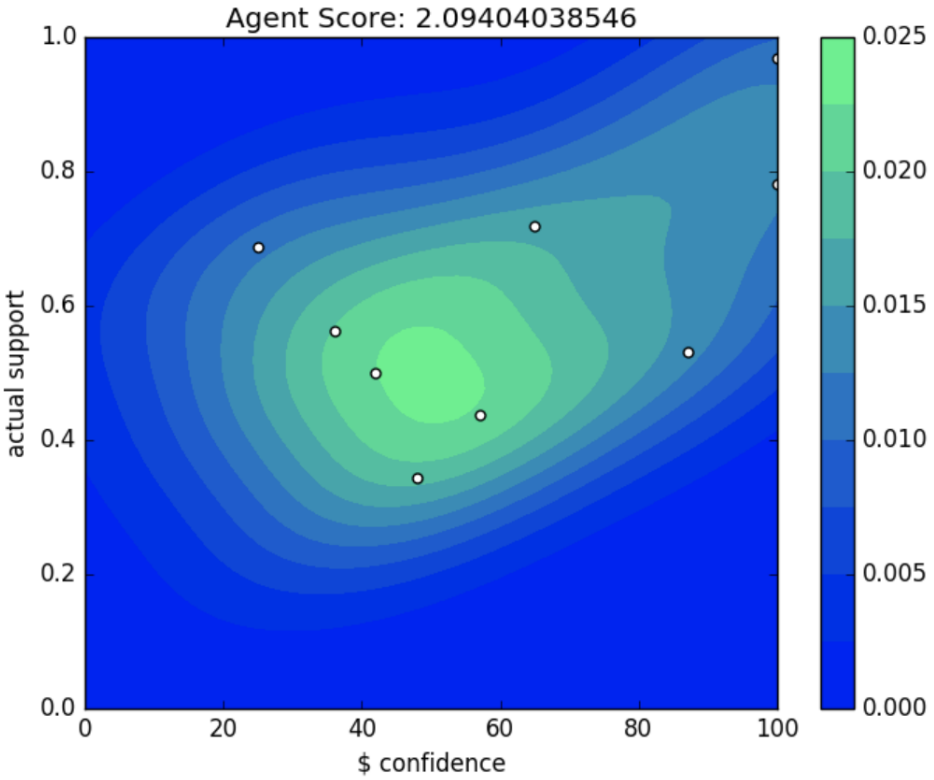

Figure 5.7: Example of Two-Dimensional Gaussian Kernel Density between Dollar Confidence and Actual Support

Figure 5.7 is an example of an agents Gaussian kernel density estimation for a correlation between Actual Support and Dollar Confidence. This specific agent had a z-score of 2.094, performing in the 95th percentile of the population that week. Each point reflects a single game prediction and the side color bar represents the relative frequency color scale for this graph. In the early stages of this project, visual trends in these graphs were used for heuristic functions in selecting top performers. The performance of this metric was slightly better than random selection but did not achieve the same results of applied machine learning, offering a good temporary solution. This section will explore the efficacy of using correlated features to train the machine learning model.

5.2.1 Dollar Confidence and Vegas Odds Correlation

The combination of the single feature kernel density estimations for these two features provided the best results for Section 5.1 leading to the hypothesis that a two-dimensional kernel density estimation containing the correlation of these two features might provide more value to this model.

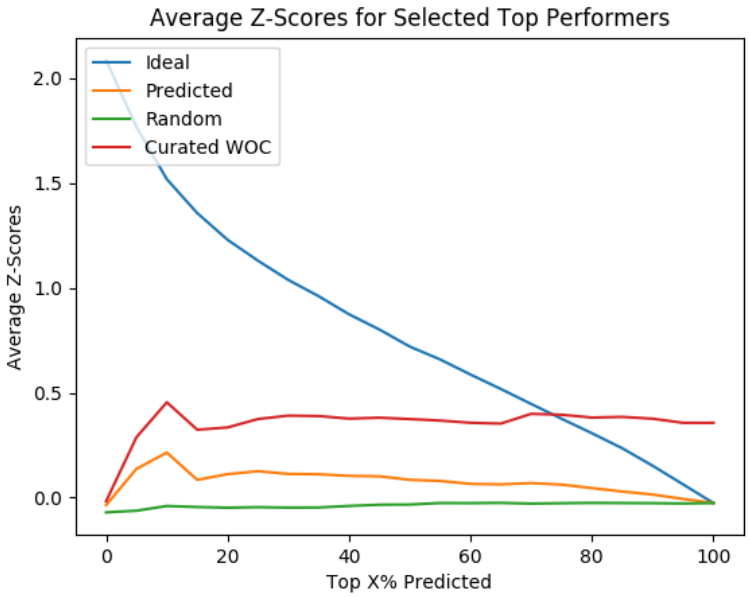

Figure 5.8: Average Z-Score for Top Performing Agents Curated by Dollar Confidence-Vegas Odds Correlation

Unfortunately, Figure 5.8 illustrates that this correlation actually resulted in a lower average predicted z-score, with the maximum at 0.215 for the top 10 percent of users compared to the maximum average z-score of 0.283 for the top 15 percent of users in the previous section. Two possible causes for this are that the trends for each individual feature are more valuable than the correlation between the two features or the sampling of the two-dimensional kernel density estimation has more noise and variation than a one dimensional kernel density estimation.

5.2.2 Actual Support and Vegas Odds Correlation

In Section 5.1.1, it was observed that a model trained on Actual Support alone performed worse than a random selection for selecting approximately the top 9 percent of users. This led to the thought that Actual Support alone may not be an effective feature; however may be more valuable when correlated with other features. The next two sections explore the results of this hypothesis.

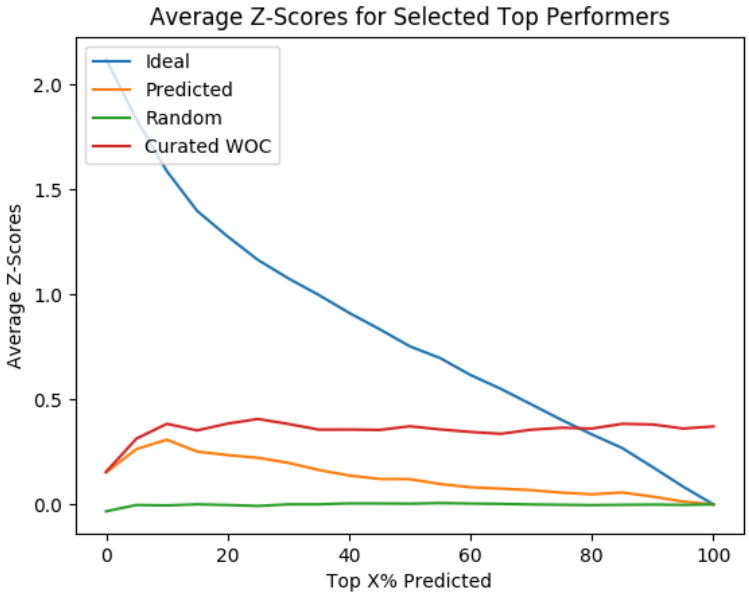

Figure 5.9: Average Z-Score for Top Performing Agents Curated by Actual Support-Vegas Odds Correlation

It is evident in Figure 5.9 that a model trained on the sampling of a two-dimensional kernel density estimation between Actual Support and Vegas Odds curates a group of top performers with an average z-score higher than random selection. Training on this feature shows ample value of the Actual Support metric when paired with Vegas Odds, as the average z-score for the top 10 percent of users is 0.309, beating every model to this point. Even with anticipated noise included in the sampling of a two-dimensional kernel density estimation, strong predictive power is exemplified in

this model.

5.2.3 Actual Support and Dollar Confidence Correlation

Unanimous AI has tested their Swarm AI platform on sports, entertainment, and political event predictions. Bookmakers have released Vegas Odds for almost all of these events; however, Unanimous AI does not want to rely completely on these odds as the ultimate goal is to beat these odds. For this reason, it is important to test features and combinations that do not contain any relation to Vegas Odds. As Actual Support performed extremely well with Vegas Odds, it seemed reasonable that Dollar Confidence and Actual Support might offer similar results.

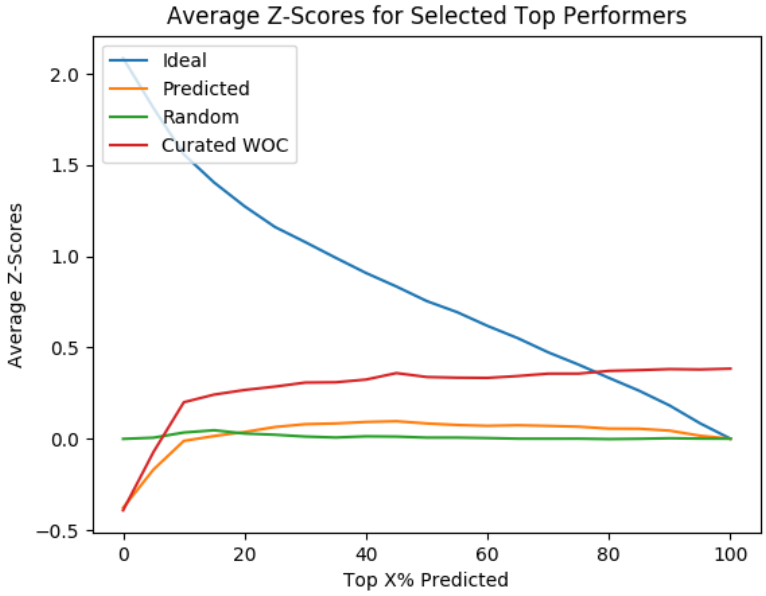

Figure 5.10: Average Z-Score for Top Performing Agents Curated by Actual Support-Dollar Confidence Correlation

This was evidently not the case. The two-dimensional kernel density estimation between Actual Support and Dollar Confidence performed horribly for the top 20 percent of users, with the predicted top users achieving an average z-score of -0.378. These results quickly improved to an average z-score of almost 0 within the top 10

percent of users; however the remaining percentiles did not deviate strongly from random. Excluding Vegas Odds from any model has proven to be a challenge up to and including this point.

5.2.4 Goal Confidence and Dollar Confidence

Goal Confidence has not proven to be a great predictor of success in this experiment, likely as it is a binary variable for NHL survey combined with the fact that betting trends tend to have high variation between users. Both confidence variables, Goal Confidence and Dollar Confidence, theoretically should have some correlation for a rational user. If a user is willing to bet more money on one team, it is logical that user will think that team has a higher probability of winning by a larger goal margin than a team they would place a low bet on. For this reason, a two-dimensional kernel density estimation was created for the correlation between Dollar Confidence and Goal Confidence.

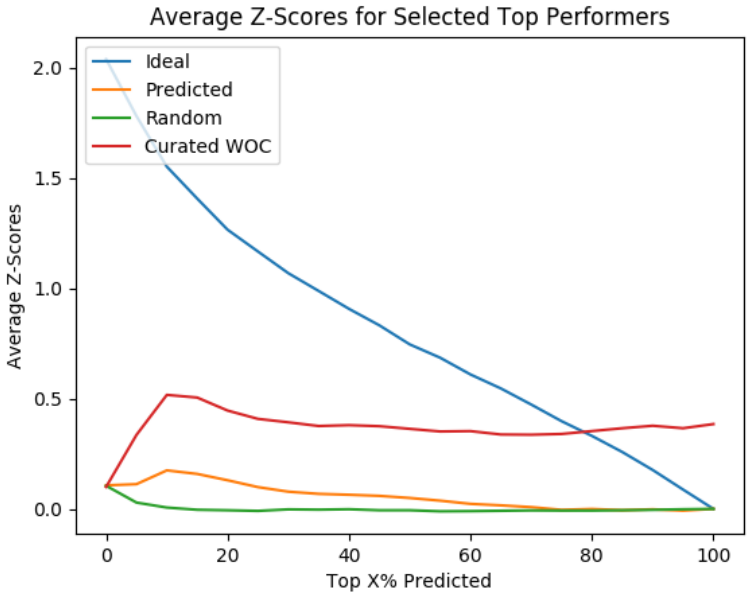

Figure 5.11: Average Z-Score for Top Performing Agents Curated by Goal Confidence-Dollar Confidence Correlation

This feature showed the top performance for any features or combination thereof that did not contain Vegas Odds, which is a significant finding for Swarm predictions that bookmakers have not assigned Vegas Odds to. The maximum average z-score of the predicted top 10 percent of users was 0.175, coming in about 0.134 lower than the top performing model and 0.054 higher than the top performing model that did not include Vegas Odds.

5.3 Feature Combinations

5.3.1 Differences between Single Features

Noise is an immense issue in machine learning. Samplings of two-dimensional kernel density estimations contained plenty of noise, but still proved to be powerful predictors of success for users. One idea to eliminate this source of noise while still capturing the correlation between features is calculating the kernel density estimation for each feature, then training a model based on the difference between the two kernel density estimations. This reduced the original feature vector size by a factor of 100. The three correlation differences that were gathered were Dollar Confidence and Actual Support, Dollar Confidence and Vegas Odds, and Dollar Confidence and Goal Confidence. A model was trained on all three of these feature kernel density estimation differences.

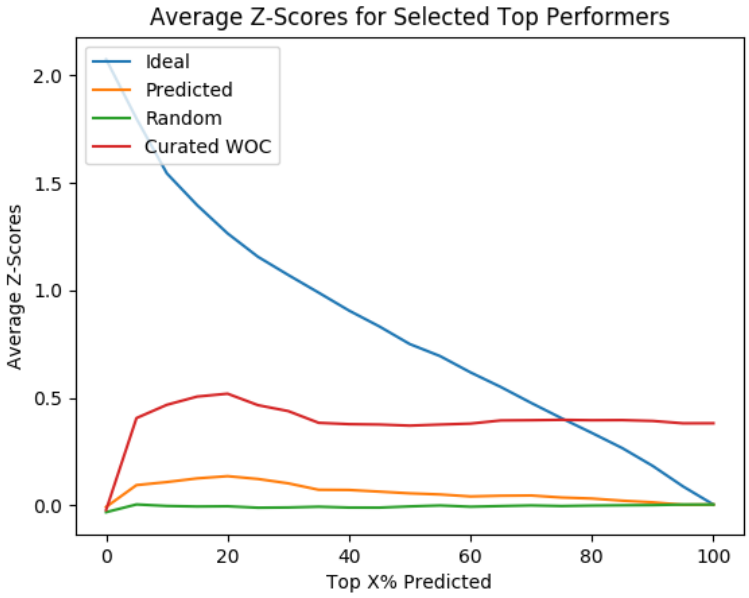

Figure 5.12: Average Z-Score for Top Performing Agents Curated by Combination of Single Feature Differences

The maximum average z-score for predicted top performers of 0.136 is not jaw dropping by any means relative to other successful metrics described previously in this experiment. The interesting finding here lies in the z-score of the Curated WOC, or the crowds z-score for the top 20 percent of users. This tells us that if the crowd of the top 20 percent of users was represented as a single user, its predictions and betting strategy would give it a z-score of 0.520 for that week. The crowd is expected to perform significantly better than the average user due to the Wisdom of the Crowd effect, but this feature combination stands out as 0.520 was the highest curated Wisdom of the Crowd z-score achieved by any model when selecting less than half of a crowd. In simpler terms, this may imply that although a group of users curated based on this metric may not have the highest average z-score, the collective intelligence of this crowd appears to be the most powerful.

5.3.2 Top Performers from Previous Sections

Sections 5.1 and 5.2 revealed that the top two performing models were Dollar Confidence kernel density estimation combined with the Vegas Odds kernel density estimation and the two-dimensional kernel density estimation between Vegas Odds and Actual Support. This led to the idea of training a model with the combination of all features included in both of these models.

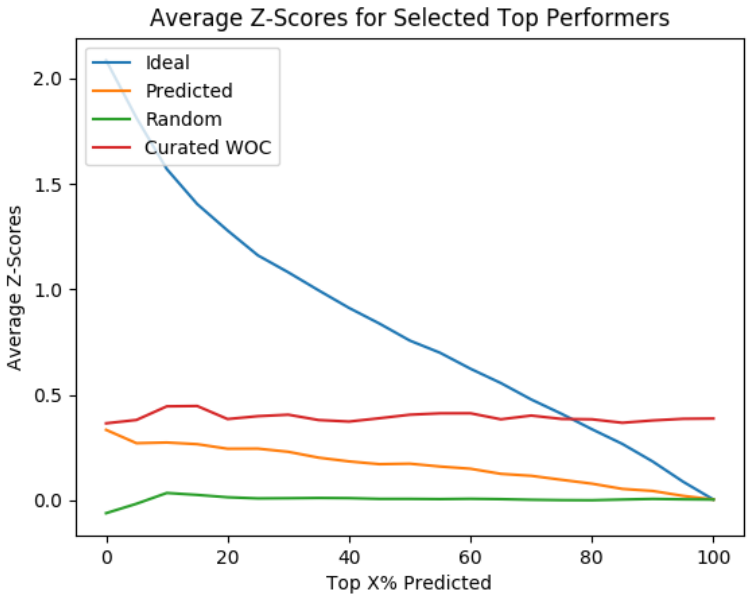

Figure 5.13: Average Z-Score for Top Performing Agents Curated by Combination of Dollar Confidence, Vegas Odds, and Vegas Odds-Actual Support Correlation

The combination of a sampling from these features led to the best z-score for the predicted top performer of the crowd of 0.333. This was the highest average z-score for the top predicted performers in any model; however, this only holds true for the top percentile, or top performer in this case. Moving to the top 10 percent of users holds an average z-score of 0.273 for the predicted top performers. Although 0.273 is above average and outperforms many other models, it is below the 0.309 apex for the 10th percentile set by the Vegas Odds and Actual Support correlation.

Legend Description	
Ideal	Average WOC Accuracy for crowd of actual top performers
Predicted	Average WOC Accuracy for crowd of predicted top performers
Random	Average WOC Accuracy for crowd of randomly selected users
Vegas	Average Accuracy for Vegas Favorite

Table 5.3: Wisdom of the Crowd Accuracy Graph Legend Descriptions

5.4 Wisdom of the Crowd Analysis

Developing methods to make swarm smarter is the overarching goal and motivation behind this project. Unfortunately, curating crowds and developing swarms for each model tested in this experiment is infeasible. To measure the performance of a crowd, a Wisdom of the Crowd metric was created. Wisdom of the Crowd accuracy is the accuracy of a crowds decisions based on a majority vote for each prediction in the survey responses. Previous sections analyzed results on how well each predicted who the top performers in a crowd are, whereas this metric assesses how a crowd of our top performers might perform relative to a randomly selected crowd of the same size in a swarm. Table 5.3 explains the legend for Wisdom of the Crowd evaluation criteria graphs used in this section.

5.4.1 Actual Support and Vegas Odds Correlation

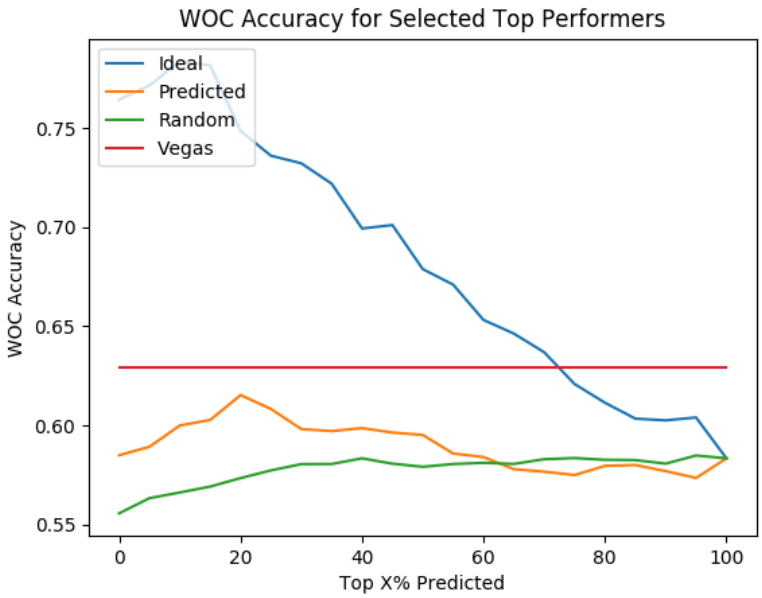

Figure 5.14: Wisdom of the Crowd Accuracy for Crowd of Top Performing Agents Curated by Actual Support and Vegas Odds Two-Dimensional Kernel Density Estimation

Section 5.2.2 explains why the model trained using the correlation Actual Support and Vegas Odds arguably had the most predictive power for selecting top performers. High performing users are measured on individual performance only. Exploring whether a crowd of high individual performers will perform better than a randomly selected crowd was the larger motivation for this project. Vegas favorites are one of the most accurate predictions for winners with an 62.9% accuracy. Using the Wisdom of the Crowd predictions for a crowd curated based on the correlation between Actual Support and Vegas Odds offers a maximum average accuracy of 61.5% for a crowd consisting of the top 20 percent of users. A random crowd of the same size produces an average accuracy of 57.3%. Discovering a crowd that performs 4.1% higher than a randomly selected crowd and only 1.4% lower than the Vegas favorites using survey responses is a significant finding. Theoretically the same set of users would achieve a

higher accuracy on the Swarm platform than in surveys based on past performance. This leads to the possibility of forming a crowd that produces predictions that can compete with or even beat Vegas favorite picks.

5.4.2 Dollar Confidence and Vegas Odds

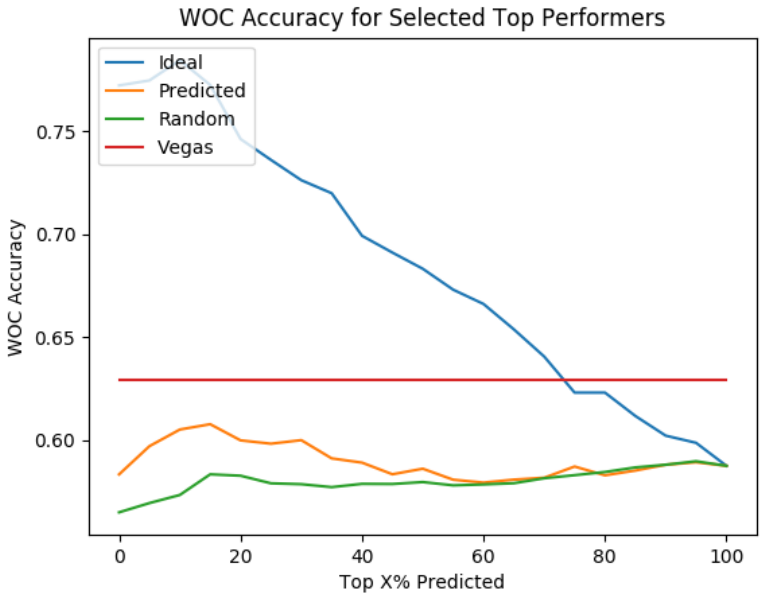

Figure 5.15: Wisdom of the Crowd Accuracy for Crowd of Top Performing Agents Curated by Combination of Dollar Confidence and Vegas Odds

Using only single feature kernel density estimations, a model trained with the combination of Vegas Odds and Dollar Confidence produced the best crowds based on the top average actual z-score. Looking at Figure 5.15, it is evident that this metric also produces a high performing overall crowd. With a maximum Wisdom of the Crowd accuracy of 60.8%, this model only sits 0.7% below the Actual Support and Vegas Odds correlation model. To put this in perspective, this model could achieve the same accuracy by predicting only 1 more of 143 games correctly illustrating how close these two crowds are linked to in performance. Additionally, the accuracy of this crowd is comfortably larger than the accuracy of the randomly selected crowd

with a 2.8% difference between the two.

5.4.3 Difference Between Dollar Confidence KDE and Vegas Odds KDE

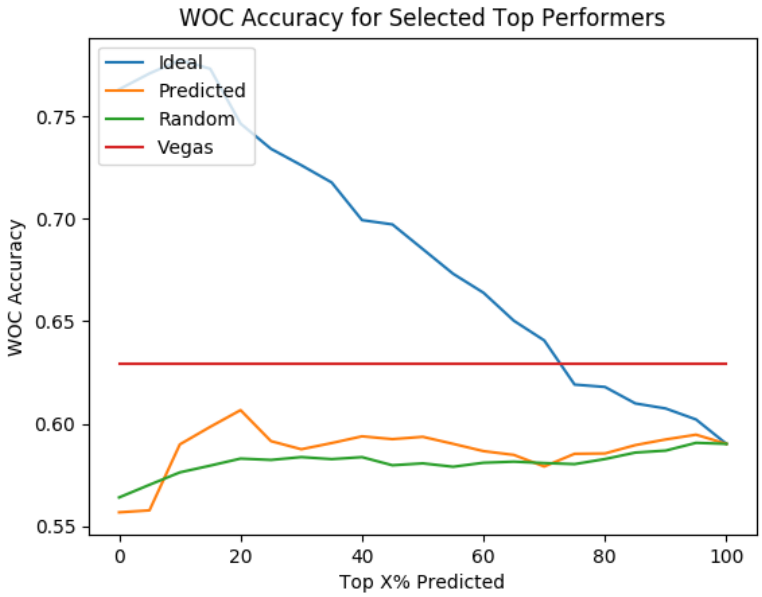

Figure 5.16: Wisdom of the Crowd Accuracy for Crowd of Top Performing Agents Curated by Difference Between Dollar Confidence KDE and Vegas Odds KDE

Section 5.3.1 describes an approach to creating new features based on the difference between the single kernel density estimations of two features. The previous section only explored the combination of various feature differences; however, this section dives into a crowd created based only on the difference between the Dollar Confidence and Vegas Odds kernel density estimations. Although this model did not produce significant results looking at the average z-score of top predicted performers, the Wisdom of the Crowd graph illustrates that this model can select a crowd that performs better than a randomly selected crowd. This crowd produced a maximum accuracy of 60.7%, only 0.1% less than the combination of the same two kernel density estimations. The major significance of this finding is that a model that does not ac-

curately select the top actual performers may still curate a crowd that is significantly more accurate than a randomly selected crowd.

5.5 Hyperparameter Optimization

As illustrated in Figure 4.1 depicting the overall machine learning pipeline for this project, a hyperparameter optimization wrapper covers the feature engineering and model creation processes. In order to limit the scope of this hyperparameter optimization run, the focus was placed on model architecture and creation. The variables included for hyperparameter optimization included input dimension for the model, number of layers, number of nodes for each layer, activation function for each layer, number of epochs, and optimizer for the model. The function to maximize was the distance between the average z-score of predicted top users and the average z-score of random users at 5% population intervals of the whole crowd. The models were trained using the best feature combination found in previous sections - Vegas Odds KDE, Dollar Confidence KDE, and Vegas Odds and Actual Support Correlation KDE. Due to time and compute restraints, feature selection was not included as a parameter; however, this feature is supported in the program. Additionally, only 50 models were evaluated, meaning there is plenty of room for improvement with a more extensive hyperparameter optimization run.

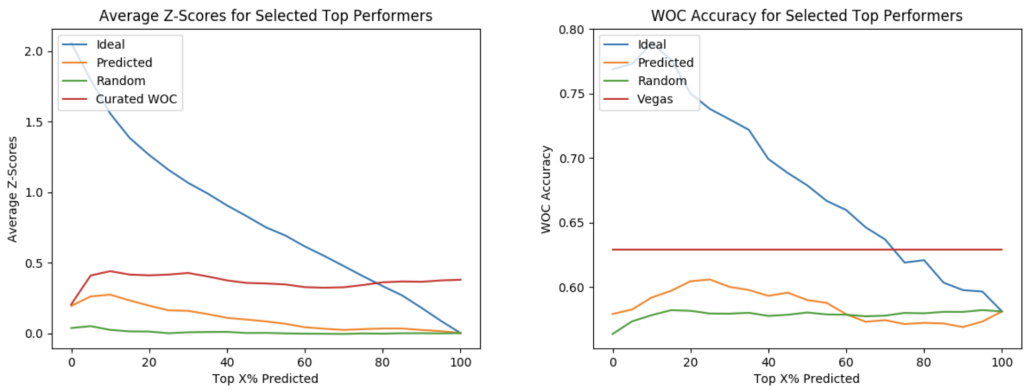

Figure 5.17: Average Z-Score and Wisdom of the Crowd Accuracy for Crowd of Top Performing Agents Curated with Optimized Model

Figure 5.17 illustrates the results of the optimized model trained using the best feature combination found in previous sections - Vegas Odds, Dollar Confidence, and the Vegas Odds-Actual Support Correlation. The maximum average z-score of users was 0.274 when selecting 10% of users, which is not the top result of all models; however, is near the top. Similarly, the maximum Wisdom of the Crowd accuracy score is 60.6% selecting the top 25% of the crowd. Again, this is a strong Wisdom of the Crowd accuracy score, but not the best. However, the notable feature of this graph is the consistency of the Wisdom of the Crowd accuracy relative to other feature combinations and models. This model demonstrated the ability to curate a crowd that consistently had a higher Wisdom of the Crowd accuracy, which aligns with the function that was maximized in the hyperparameter optimization. To account for this, the maximized function in hyperopt could be modified to maximize the top average z-score or Wisdom of the Crowd accuracy for a curated crowd or the average distance between the average z-scores or Wisdom of the Crowd accuracies for a curated crowd and randomly selected crowd, effectively including the ability to change the ultimate goal for the program.

Chapter 6

FUTURE WORKS

6.1 Cross Sport Analysis

Unanimous AI has tested the Swarm AI platform with a variety of sports leagues including Major League Baseball, National Basketball League, National Football League, English Premier League, National Hockey League, and more. This experiment only included participants from National Hockey League surveys and swarms. This significantly reduced the amount of data available and limited the scope of this experiment. Participants across these leagues are often different; however, exploring similarities in performance across leagues will provide more data to each experiment, fostering a stronger machine learning environment. There are two facets to this claim. The first is goal confidence will vary to point spread and run spread, which drastically changes sport to sport. Additionally, more data does not always mean good data. Machine learning requires relevant data, therefore requiring further experimentation to reveal if cross sport training is effective with respect to selecting the top performers from survey data.

6.2 Feature Engineering

Feature engineering is generally the most important aspect of a machine learning project. As seen in Chapter 5, using different features to train the same machine learning model produced drastically different results. With this being said, the most important aspect of this project to focus on moving forward is feature engineering. There are two proposals for this. Features included in the survey could be further manipulated and explored. Creating new training models based on modifications these

features could increase the predictive power of the features included in this dataset. The second recommendation is to extend each survey to include additional information about each participant. Overall, feature engineering is an iterative process. It will likely never achieve the optimal point, and requires extensive experimentation to improve on the current program.

6.3 Command Line Interface

The command line interface for this project produced more than required for this project, but could be improved. Currently, the command line interface only supports single neural network architecture with user-inputted features. Moving toward future iterations, it is recommended that this interface is extended to include machine learning parameter options, hyper-parameter optimization, and stronger input validation within the script.

6.4 Model Optimization

Implementing hyperopt was the first step toward model optimization for this project. For first steps, it provided great value in determining a good base architecture for this project. This base architecture simply included the depth of the neural network, the number of layers, the number of input dimensions, and the activation functions for each layer. Extending this architecture to support larger models, different activation functions, different loss functions and evaluation criteria, and finally the overall evaluation of the model may drastically change the outcome of this experiment. With this being said, my recommendation is to modularize each of these components to further improve the performance of this experiment and process.

BIBLIOGRAPHY

[1] C. Blum and X. Li. *Swarm Intelligence in Optimization*. Springer Verlag, 2008.

[2] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, 1999.

[3] A. Cuthbertson. Artificial Intelligence Turns $20 Into $11,000 in Kentucky Derby Bet. *Newsweek*, 2018.

[4] N. C. Dalkey. Delphi. *RAND Corporation*, 30, October 1967.

[5] J. G. Joseph P. Simmons, Leif D. Nelson and S. Frederick. Intuitive Biases in Choice versus Estimation: Implications for the Wisdom of Crowds. *Journal of Consumer Research*, 2011.

[6] K. Kauko and P. Palmroos. The Delphi method in forecasting financial markets-An experimental study. *International Journal of Forecasting*, 30, April-June 2014.

[7] M. D. Lee, M. Steyvers, M. de Young, and B. Miller. Inferring Expertise in Knowledge and Prediction Ranking Tasks. *Topics in Cognitive Science*, 4(1):151–163, January 2012.

[8] H. A. Linstone and M. Turoff. *The Delphi Method, Techniques and Applications*. Wesley Publishing Company, 1977.

[9] A. Lyon and E. Pacuit. The Wisdom of Crowds: Methods of Human Judgement Aggregation. *Handbook of Human Computation*, pages 599–614, November 2013.

[10] G. Madirolas and G. G. De Polavieja. Improving Collective Estimations Using Resistance to Social Influence. *PLOS Computational Biology*, 2015.

[11] L. Rosenberg. Artificial Swarm Intelligence vs Human Experts. *International Joint Conference on Neural Networks (IJCNN)*, July 2016.

[12] L. Rosenberg, D. Baltaxe, and N. Pescetelli. *Crowds vs Swarms, a Comparison of Intelligence.* IEEE 2016 Swarm/Human Blended Intelligence Workshop (SHBI), 2016.

[13] L. Rosenberg and N. Pescetelli. Crowds vs. Swarms, a Comparison of Intelligence. *IEEE Swarm/Human Blended Intelligence Workshop*, 2016.

[14] L. Rosenberg and N. Pescetelli. Amplifying Prediction Accuracy using Swarm A.I. *Science and Information Conferences - Intelligent Systems Conference (Intellisys)*, 2017.

[15] L. Rosenberg and G. Wilcox. *Artificial Swarm Intelligence vs Vegas Betting Markets.* IEEE Developments in eSystems Engineering, 2018.

[16] M. Spann and B. Skiera. Sports Forecasting: A Comparison of the Forecast Accuracy of Prediction Markets, Betting Odds and Tipsters. *Journal of Forecasting*, 2008.

[17] J. Whitehill, T. fan Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 2035–2043. Curran Associates, Inc., 2009.