Machine Learning Nanodegree

Capstone Proposal

This is the proposal document towards the completion of the Machine Learning Nanodegree from Udacity.

Domain Background

Early childhood education research is conducted by many reputed organizations. PBS Kids is one such organization trying to gain deep insights into how modern media helps young children learn important skills needed for success in life. They use unique measuring games and measurement-focused video clips that feature well-known PBS characters to teach the concepts and also reward them as needed. Data Science Bowl (https://datasciencebowl.com/ (https://datasciencebowl.com/)) is conducting a challenge on Kaggle in which participants have to predict certain scores that will aid PBS in devising higher quality educational media and improved learning processes. Of the many challenges presented on Kaggle, I chose this one due to my interest in early childhood education and providing the right kind of tools and resources to kids to help them relaize their potential sooner.

Problem Statement

The Kaggle challenge provides anonymous gameplay data which includes the information about the video clips that were watched by the kids and the games that they played while using the PBS Kids Measure Up! app. In this app the children who are aged between 3 and 5 learn early STEM concepts like length, width, capacity and weight while experiencing an adventure-like gameplay through Treetop City, Magma Peak and Crystal Caves. They collect rewards along the way and can unlock digital toys during their play. The challenge is to predict the scores on the in-game assessments and create an algorithm that will lead to better game design and improve learning outcomes. The challenge has been open for several weeks and they have published a live leaderboard online. The solution is scored using a quadratic weighted kappa, which measures the agreement between two outcomes.

More details about the scoring are available here. https://www.kaggle.com/c/data-science-bowl-2019/overview/evaluation (https://www.kaggle.com/c/data-science-bowl-2019/overview/evaluation)

Datasets and Inputs

The data comes from the PBS Kids Measure Up! app. As the child uses the app to navigate a map while playing games, watching video clips and completing various levels, data is collected and anonymized in accordance with child privacy regulations. This is presented in a tabular form containing the child's interactions within the app, such as the in-app assessment score or their path through the game and certain analytics. The assessments are

designed to test the child's comprehension of a certain set of skills. The game presents five assessments: Bird Measurer, Cart Balancer, Cauldron Filler, Chest Sorter, and Mushroom Sorter.

The challenge provides two main data file (train.csv and test.csv) each of which contain the gameplay events with the following features:

- event_id: Randomly generated unique identifier for the event type.
- game_session: Randomly generated unique identifier grouping events within a single game or video play session.
- timestamp: Client-generated datetime
- event_data: Semi-structured JSON formatted string containing the events arameters. Default fields are: event_count, event_code, and game_time; therwise fields are determined by the event type.
- installation_id: Randomly generated unique identifier grouping game sessions within a single installed application instance.
- event_count: Incremental counter of events within a game session (offset at 1). Extracted from event_data.
- event_code: Identifier of the event 'class'. Unique per game, but may be duplicated across games. E.g. event code '2000' always identifies the 'Start Game' event for all games. Extracted from event_data.
- game_time: Time in milliseconds since the start of the game session. Extracted from event_data.
- title: Title of the game or video.
- type: Media type of the game or video. Possible values are: 'Game', 'Assessment', 'Activity', 'Clip'.
- world: The section of the application the game or video belongs to. Helpful to identify the educational curriculum goals of the media. Possible values are: 'NONE' (at the app's start screen), TREETOPCITY' (Length/Height), 'MAGMAPEAK' (Capacity/Displacement), 'CRYSTALCAVES' (Weight).

The training set contains 11341042(~11.3M) records. It has 11 columns as shown in the figure below.

The testing set contains 1156414(1.1M) records and has 11 columns as can be seen from the image below.

Since the testing set doesnt give the correct answers I will create a validation set and most likely use KFold cross-validation to to evaluate the model's fit using the generated score.

The dataset contains another file **specs.csv** which gives the specification of the various event types.

- event_id: Global unique identifier for the event type. Joins to event_id column in events table.
- info: Description of the event.
- args: JSON formatted string of event arguments. Each argument contains:

- name: Argument name.
- type: Type of the argument (string, int, number, object, array).
- info: Description of the argument.

The dataset also includes a file **train_labels.csv** which demonstates how to compute the ground truth for the assessments in the training set.

Solution Statement

The intent of the Kaggle competition is to use the gameplay data to forecast how many attempts a child will take to pass a given assessment. The outcomes in this competition are grouped into 4 groups (labeled accuracy_group in the data): 3: the assessment was solved on the first attempt 2: the assessment was solved on the second attempt 1: the assessment was solved after 3 or more attempts 0: the assessment was never solved

More detailed information is at the link given below: https://www.kaggle.com/c/data-science-bowl-2019/overview/evaluation (https://www.kaggle.com/c/data-science-bowl-2019/overview/evaluation)

I would like to treat this as suitable for a regression analysis and predict the 0/1/2/3 outcome for each game play in the test set.

Benchmark Model

I will first try to solve the challenge with a naive linear Regression model and generate scores to compare with my real solution. This naive model will be the benchmark and the aim will be to exceed the benchmark with the real solution.

Further, the final result can also be compared on the competition's leaderboard as a measure ofsuccess of the selected models.

Evaluation Metrics

Kaggle scores the submissions based on the quadratic weighted kappa, which measures the agreement between two outcomes. This metric typically varies from 0 (random agreement) to 1 (complete agreement). In the event that there is less agreement than expected by chance, the metric may go below 0. More details on how the metric is calculated is available at the link given below: https://www.kaggle.com/c/data-science-bowl-2019/overview/evaluation (https://www.kaggle.com/c/data-science-bowl-2019/overview/evaluation)

Project Design

The design of the project will follow the following set of steps as outlined below:

- Data Exploration: The data from the dataset will be imported into a Pandas dataframe and various aspect of it willbe analyzed.
- Data Preprocessing: The data needs to be cleaned up as necessary (for example, taking care of NaNs etc.,), formatted and restructured to transform any skewed features. Other scaling and normalization operations can also be performed to make the data suitable

as input to the machine learning algorithm.

- Data Visualization: Visual representation of data using libraries like matplotlib, seaborn will be used to find correlation patterns in the data between the predictors and the target variable.
- Model Selection: The baseline analysis wil be done using a linear regression model to setup a benchmark. After doing a quick comparison of the features and performance of XGBM and LightGBM, I have picked LightGBMRegressor as the model of choice to use for this particular problem. I plan to use Convolutional Neural Networks with multiple hidden layers for training purposes.
- Model Evaluation: Measure the alogirthm's performance against the benchmark created earlier.
- Model Tuning: Fine tune the selected algorithm's hyperparameters to increase performance.
- Feature Optimization: The features of the dataset are analyzed for feature importance, feature relevance and any effects of feature selection. If optimal list of features are found, they willbe inlcuded in the final model.
- Testing: Test the model on testing dataset.