The solution.py file contains feature engineering, model pipeline (OneHotEncoding, Standard Scaling and Logistic regression) fit and prediction on the test dataset.

To run the code, one must provide the path to next files: train_matches.json, test_matches.json (given by the competition).

The output is the submission csv file.

In addition to the features initially provided by the competition organizers, we extract from the .json files : ‘ability_level’, ‘max_hero_hit’, ‘purchase_count’, ‘count_ability_use’, ‘damage_dealt’, ‘damage_received’. Majority of the features were used in the aggregated form among the members of the same team (Radiant or Dire), providing an effective dimensionality reduction. Coordinates were converted to the indicator functions for each player of being in their own home base, in the opponent’s base or neither of them. ‘hero_id’ s were converted into success rate of winning based on the training data. Logarithmic transformations of some skewed features were also added.

In the model: some categorical features were converted to binary using OneHotEncoder(), numerical features were standardized and logistic regression (C=10, LASSO) was applied.

<u>Not included in the execution file:</u>

Feature selection was done using RFECV (recursive feature elimination with cross-validation) and step-wise selection using test auc-score.

Logistic regression was selected among other models based on cross-validation auc-scores. Regularization parameter is selected using grid search.

References (feature extraction from .json files):

https://www.kaggle.com/artgor/dota-eda-fe-and-models