Technique overview of the influencing factors of Peugeot Sports' investment in racing cars

Zheng Bin

01/08/2021

Contents

Model 1 (ML): Predict whether a driver will finish the race

Data preprocessing

- Merge data
- Filter the data with 'inner join'

  Due to the 'Abu Dhabi Grand Prix' was included since 2009, hence only keep the data from 2009 onwards.

- Dropping unnecessary columns & fill NA with -1 & transform the data type
- Building 2 new features
  1. Age of each driver per race
  2. History of total finished per driver of each race

Modelling

- One Hot Encoder
  1. 'nationality'
  2. 'circuit'
  3. 'raceId'
- Split at a ratio of 7 to 3
- Transform the tables in a table of label, features format
- Build classification models
  1. Logistic Regression model
     - Hyperparameter tuning
       - 5-folder
       - Best LR model:
         - regParam: 0.01
         - maxIter: 100
     - Predict labels of test set using built model
     - Get model performance on test set
       - AP: 0.6235854122463895
       - AUC: 0.6660521811958343
  2. Random Forest Model
     - 5-folder
     - Get predictions on the test set
       - AP: 0.765786966232127
       - AUC: 0.23367372781707335

Constructing prediction data for the 2018 Abu Dhabi Grand Prix

- Merge the driver data with History of total finished per driver
- Add all the data we need by data processing
- Transform the tables in a table of label, features format
- Choosing the Logistic Regression model to predict on the prediction data
- Result shows 60 of the 834 drivers were predicted to complete the game

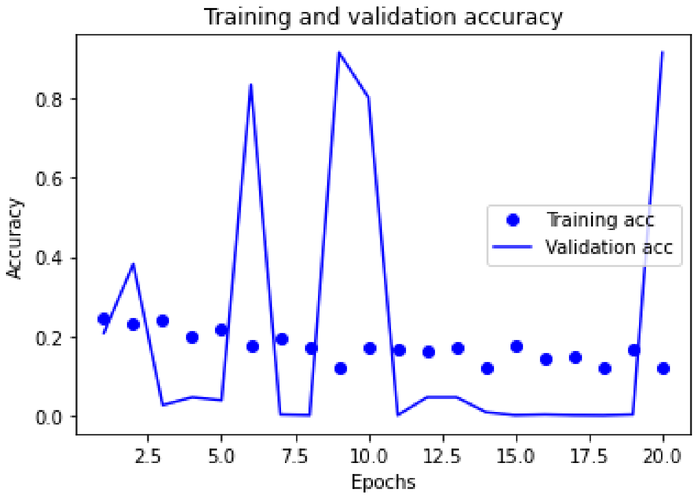
Model 2 (DL): Build a Deep Learning model to predict how many pitstops a driver will need per race

Data preprocessing

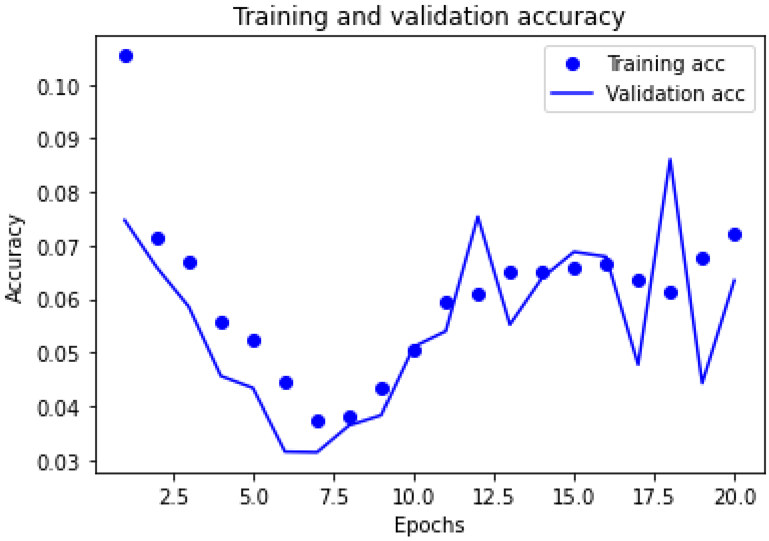
- Merge data
- Keep all the data instead of filter the data
- Dropping unnecessary columns & fill NA with -1 & transform the data type
- Building 3 new features
  1. Age of each driver per race
  2. History of total finished per driver of each race
  3. Create a column of the number of pitstops for each driver in each race
- Drop any row with missing value

Single-label Multiclass Classification of DL

- One-hot encoding of categorical variables
  - "nationality"
  - "raceId"
  - "circuitId"
  - "N_stop"
- Transform to numpy arrays
- Split in a train, val, and test set
- Model of 5 layers
  - Unit of first 4 layers are 64, the last unit is 7 (7 class of the pitstop)
  - Activation : 'softmax'
  - Define an optimizer, loss function, and metric for success
    - Loss function : 'categorical_crossentropy'
    - Optimizer : 'rmsprop'
    - Metrics : 'acc'
  - Fit the model. Use batch_size=512 and epoch=20 as start values.
  - Create plot :



    - Optimization with 5 epochs.
        - Network seems to overfit after 5 epochs (see graphs)
        - Loss : 60.81864547729492
        - Accuracy : 0.9123159050941467
- Model of 3 layers
    - Unit of first 2 layers are 64, the last unit is 7 (7 class of the pitstop)
    - Activation : 'softmax'
    - Define an optimizer, loss function, and metric for success
        - Loss function : 'categorical_crossentropy'
        - Optimizer : 'rmsprop'
        - Metrics : 'acc'
    - Fit the model. Use batch_size=512 and epoch=20 as start values.
    - Create plot :



    - Optimization with 5 epochs.
        - Network seems to overfit after 10 epochs (see graphs)
        - Loss : 0.8983985185623169
        - Accuracy : 0.008701472543179989

Summary

Through data processing and building machine learning model, the K-fold Logistic Regression model and Random Forest Model were built respectively, and it was found that the first model performed better, with an AUC of 0.67. 60 of the 834 drivers were predicted to complete the game.

After establishing the Single-label Multiclass Classification deep learning model, and comparing the performance of the 3 layer and 5 layer models, it is found that the 3 layer model performs better, which with much lower loss and a better fitting accuracy curve.

The machine learning model has a low AUC, and the performance of the model can be improved by adding new variables in the future. The deep learning model shows some overfitting, which can be optimized by Regularization or Dropout later. For the Regularization, the L1 regularization also named lasso: cost is added proportionally to the absolute value of the weight coefficients (L1 norm), due to the weight of some features can be 0, hence the method can be used to select features. L2 regularization also named ridge regression: cost is proportional to the square of the value of the weight coefficients (L2 norm), due to the weight of features can only be close to0, hence the method can't be used to select features, but we can analyze the problem of cancelation multicollinearity.

References

1. *An Introduction to Statistical Learning by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani*

2. *MBD BigData2 course materials*

3. *https://blog.csdn.net/wangcheng666666/article/details/79187703*