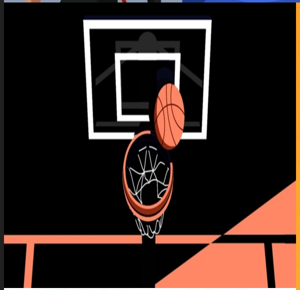
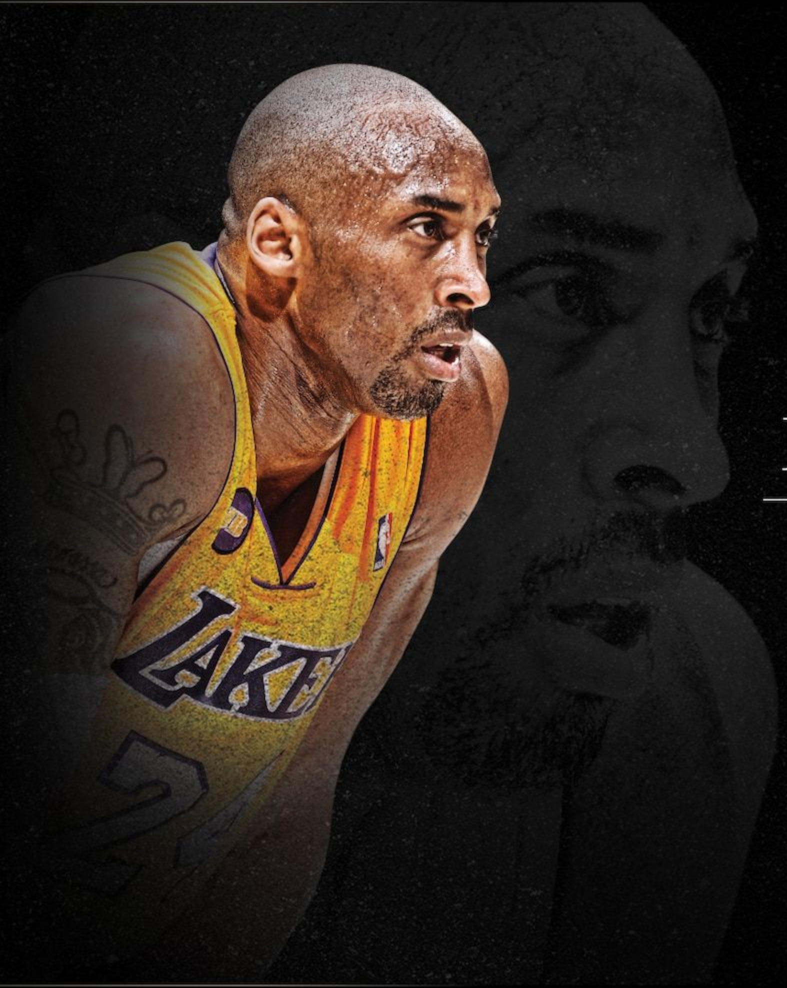
SHOT PREDICTION - KOBE BRYANT

TEAM 14: EILEANOR LAROCCO,
ERIK ZWICKLBAUER,
JUNGHEE MUN,
SEBASTIAN RINCON

Contents





KOBE BRYANT

1978-2020

“Black Mamba”

- LA Lakers (1996 - 2016)
  - 5 NBA Championships
  - 2 NBA Finals MVP
  - NBA MVP
  - 18 NBA All Star selections
  - 11 All-NBA First Team selections
  - + many more
- Team USA
  - 2008 & 2012 Olympics Gold

MAMBA MENTALITY

Objective

Problem: Using 20 years of data on Kobe Bryant's shots, create a ML model that predicts which shots were made or missed, while minimizing the log loss

Log loss

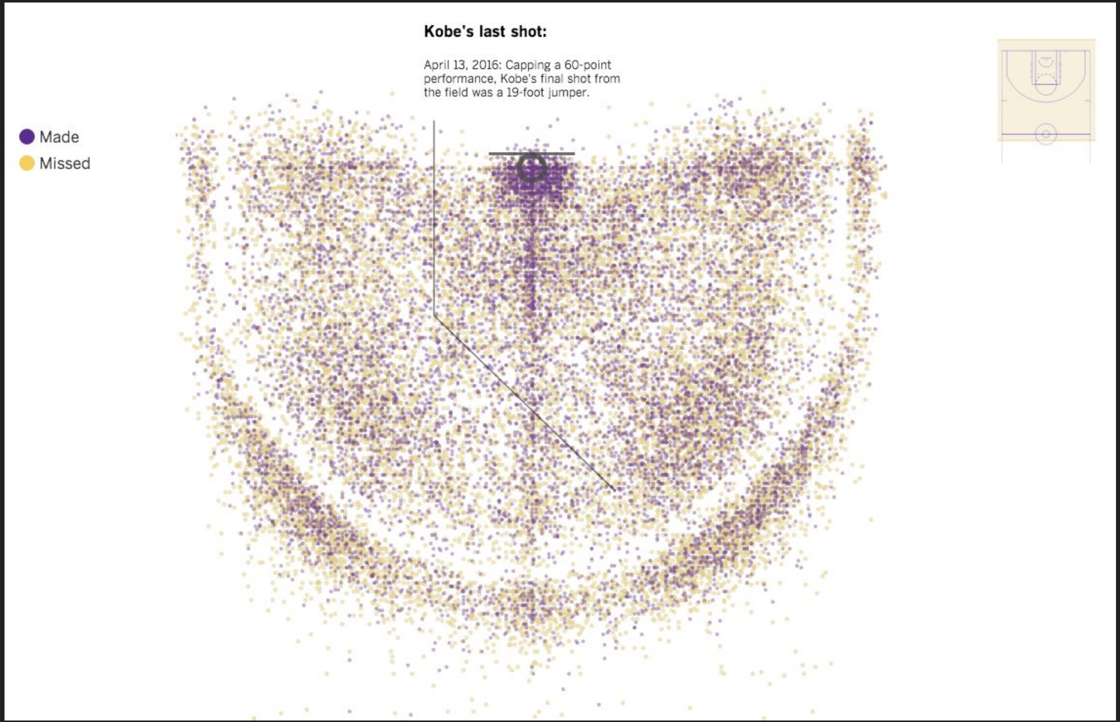
a. Assesses the performance of a classification problem by indicating how close the prediction probability is to the corresponding actual/true value.
b. The closer the values, the lower the log-loss value

Data leakage

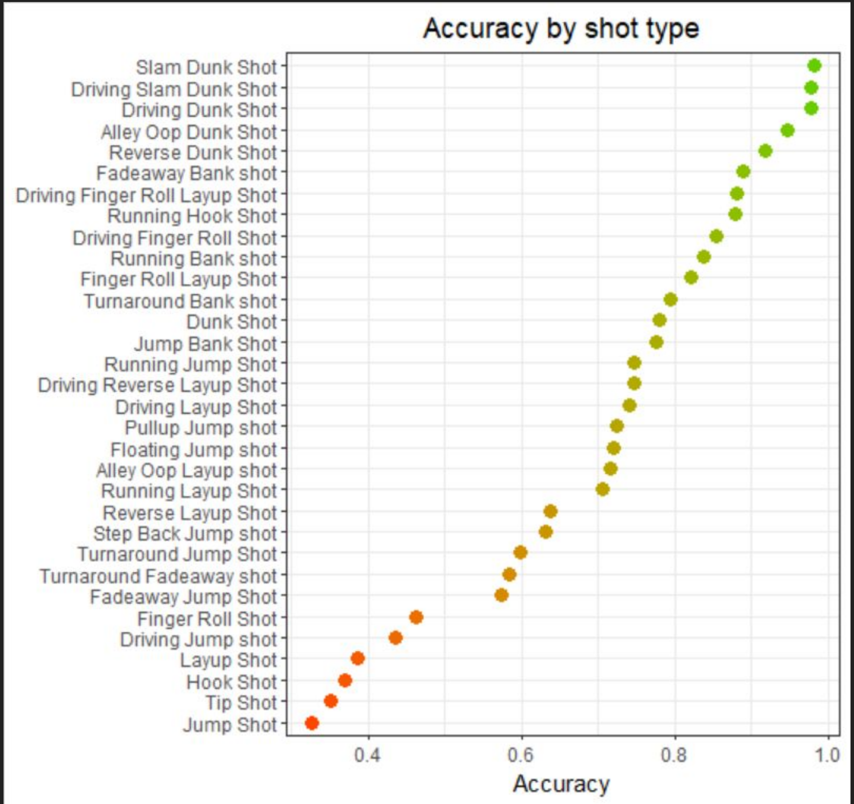
c. For the context of this problem, is using data from future events to predict past events
d. Scaling introduces leakage
e. Training a model using data sampled from all 20 years would introduce leakage since we are predicting shots made randomly throughout all 20 years

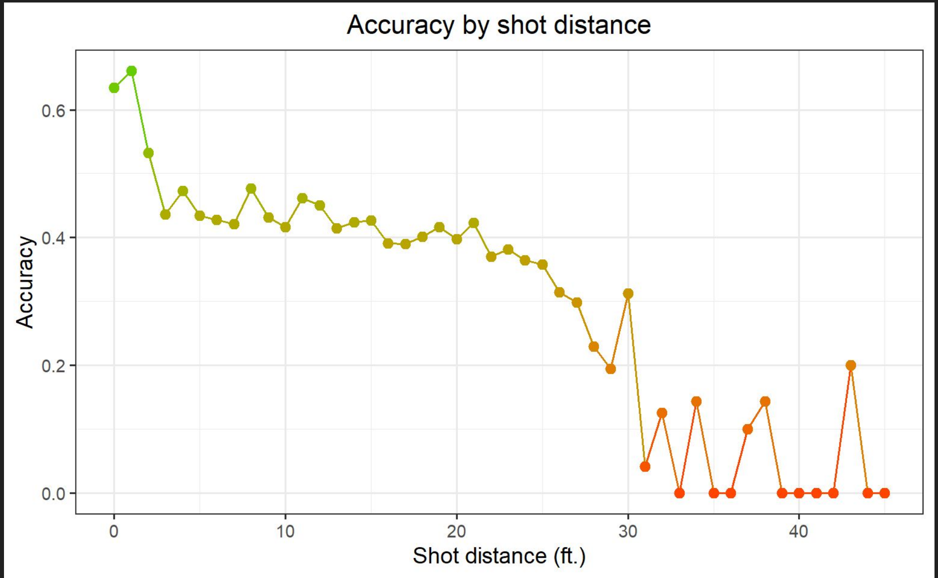
About the dataset

- Come from stats.nba.com
- 30697 rows x 25 columns
  - Mix of categorical & numerical
- 5000 predictions to be made
- Target variable: shot_made_flag
  - 0: Shot missed
  - 1: Shot made
  - NA: Predictions to be made



Intuitive Features





Kaggle Notebook Critique

Final model:

Ensemble model that included Logistic regression, Gradient boosting classifier, Random forest classifier, and AdaBoost classifier

Pros

- Assumed independence of shots
- Removed some features
- Data cleaning and feature engineering
- Handled categorical variables well
- Very thorough examination of possible models
- Tuned hyperparameters on models
- Evaluated algorithms using k-fold

Cons

- Removed outliers too soon
- Did not address leakage
- No graphs aside from initial feature distribution visualizations
- Log loss is very high (0.95)
- Scales the values which leaks information

First Attempts

Data preprocessing - cleaned data

1. Logistic Regression
2. Lasso and Ridge Regression
3. Random Forest
4. SVM

Final Model

XGBoost

- Tuned hyperparameters
- Compared against uncleaned data

Model Rationale

- Limited as to what models we could use because the metric is log loss
- Due to time constraint, unable to implement sliding window to reduce leakage- therefore our model is under the assumption that each shot is independent of the others
    - Studies done on “hot streak” psychological effect are inconclusive
- XGBoost with uncleaned data produced the lowest log loss values
    - Powerful enough to run without cleaned data

Conclusion

- Best model: **XGBoost**
  - Log Loss:
    - Kaggle: **0.53** (with sliding window & tuned hyperparameters)
    - Team 14: **0.587114** (with uncleaned data & tuned hyperparameters)
    - Team 14: **0.635801** (with cleaned data & tuned hyperparameters)
- Data leakage: Inevitable due to computational limitations
- Most important features:
  - **Action type, Combined Shot Type, Latitude, Shot Distance, Game Date**
- Hyperparameter optimization and experimentation led to a decrease in logloss
  - Can be implemented regardless of model


THE END