```
import pandas as pd
import pickle
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
from google.colab import drive
drive.mount('/content/drive')
```

loading train/test data

```
with open('/content/drive/MyDrive/Intro ML 2022 Summer/dataset/basketball_train.pkl',
  train = pickle.load(train_data)

with open('/content/drive/MyDrive/Intro ML 2022 Summer/dataset/basketball_test.pkl', '
  test = pickle.load(test_data)
```

calculating "k" by cross validation

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import cross_val_score

# find best k, range from 3 to half of the number of data
max_k_range = train.shape[0] //2
k_list = []
for i in range(3, max_k_range, 1):
    k_list.append(i)

cross_validation_scores = []
x_train = train[['3P', 'BLK' , 'TRB']]
y_train = train[['Pos']]
```

```
# 10-fold cv
for k in k_list:
    knn = KNeighborsClassifier(n_neighbors=k)
    scores = cross_val_score(knn, x_train, y_train.values.ravel(),
                             cv=10, scoring='accuracy')
    cross_validation_scores.append(scores.mean())

cross_validation_scores
```

```
```

```
# visualize accuracy according to k
plt.plot(k_list, cross_validation_scores)
plt.xlabel('k')
plt.ylabel('Accuracy')
plt.show()
```

```
# find best k
cvs = cross_validation_scores
k = k_list[cvs.index(max(cross_validation_scores))]
print("The best number of k : " + str(k) )
```

using two features only (3P, BLK)

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

knn = KNeighborsClassifier(n_neighbors=k)

x_train = train[['3P', 'BLK']]
y_train = train[['Pos']]

# setup knn using train data
knn.fit(x_train, y_train.values.ravel())

# select data feature to be used for prediction
x_test = test[['3P', 'BLK']]

# select target value
y_test = test[['Pos']]

# test
pred = knn.predict(x_test)
```

```
# check ground_truth with knn prediction
comparison = pd.DataFrame(
    {'prediction':pred, 'ground_truth':y_test.values.ravel()})
comparison
```

```
# check accuracy
print("accuracy is "+str(accuracy_score(y_test.values.ravel(), pred)) )
```

using three features (3P, BLK, TRB)

```

```

```
knn = KNeighborsClassifier(n_neighbors=k)

# select data features to be used in train
x_train = train[['3P', 'BLK', 'TRB']]
# select target
y_train = train[['Pos']]

# build knn model
knn.fit(x_train, y_train.values.ravel())

# select features to be used for prediction
x_test = test[['3P', 'BLK', 'TRB']]

# select target
y_test = test[['Pos']]

# test
pred = knn.predict(x_test)
```

```
# check ground_truth with knn prediction
comparison = pd.DataFrame(
    {'prediction':pred, 'ground_truth':y_test.values.ravel()})
comparison
```

```
# check accuracy
print("accuracy is " + str( accuracy_score(y_test.values.ravel(), pred)) )
```