details

December 1, 2023

1 Details

```
[54]: import pandas as pd
      import numpy as np
```

```
[55]: # read all cleaned tables
      box_scores = pd.read_csv("../phase2/cleaned-data/box_scores_2010_2017.csv").
       ↪drop(columns=["Unnamed: 0"])
      game_data = pd.read_csv("../phase2/cleaned-data/nfl_game_data_2010_2023.csv").
       ↪drop(columns=["Unnamed: 0"])
      team_stats = pd.read_csv("../phase2/cleaned-data/nfl_team_stats_2010_2021.csv").
       ↪drop(columns=["Unnamed: 0"])
      nfl_teams = pd.read_csv("../phase2/cleaned-data/nfl_teams_info.csv").
       ↪drop(columns=["Unnamed: 0"])
```

1.1 Setting Up Data

```
[56]: # adding values for the St. Louis Rams and Las Vegas Raiders as they show up in␣
       ↪the other datasets
      rams = pd.DataFrame({"team_name": "St. Louis Rams", "team_name_short": "Rams",␣
       ↪"team_id": "LAR", "team_conference": "NFC", "team_division": "NFC West"},␣
       ↪index=[0])
      raiders = pd.DataFrame({"team_name": "Las Vegas Raiders", "team_name_short":␣
       ↪"Raiders", "team_id": "LVR", "team_conference": "AFC", "team_division": "AFC␣
       ↪West"}, index=[0])
      nfl_teams = pd.concat([nfl_teams, rams, raiders], ignore_index=True)
      nfl_teams = nfl_teams.sort_values(by=["team_division"]).reset_index(drop=True)
      nfl_teams.head()
```

```
[56]:              team_name team_name_short team_id team_conference team_division
      0  New England Patriots        Patriots      NE             AFC      AFC East
      1         Buffalo Bills           Bills     BUF             AFC      AFC East
      2        Miami Dolphins        Dolphins     MIA             AFC      AFC East
      3         New York Jets            Jets     NYJ             AFC      AFC East
      4      Baltimore Ravens          Ravens     BAL             AFC     AFC North
```

[57]:
```
# function to get team id from city/team name
def get_team_id(city):
    # find the team name
    for team in nfl_teams["team_name"]:
        if city in team:
            return nfl_teams[nfl_teams["team_name"] == team]["team_id"].
 ↪values[0]
        elif city == "NY Giants":
            return "NYG"
        elif city == "NY Jets":
            return "NYJ"
        elif city == "LA Rams":
            return "LAR"
        elif city == "LA Chargers":
            return "LAC"
```

[58]:
```
# adding team ids to the box scores dataset
box_scores["home_id"] = box_scores["home"].apply(get_team_id)
box_scores["away_id"] = box_scores["visitor"].apply(get_team_id)

box_scores.head()
```

[58]:
```
         date      visitor          home  visitor_score  home_score  \
0  2014-09-07     Cleveland    Pittsburgh             27          30
1  2014-09-07  Jacksonville  Philadelphia             17          34
2  2014-09-04     Green Bay       Seattle             16          36
3  2014-09-07     Minnesota     St. Louis             34           6
4  2014-09-07    Cincinnati     Baltimore             23          16

   visitor_first_downs  visitor_net_yards  visitor_total_plays  \
0                   23                389                   64
1                   18                306                   70
2                   19                255                   57
3                   18                355                   57
4                   16                380                   64

   visitor_avg_gain visitor_time_of_possession  home_first_downs  \
0               6.1                      27:33                24
1               4.4                      29:14                24
2               4.5                      26:40                25
3               6.2                      28:17                15
4               5.9                      30:30                26

   home_net_yards  home_total_plays  home_avg_gain home_time_of_possession  \
0             503                67            7.5                   32:27
1             420                82            5.1                   30:46
2             398                66            6.0                   33:20
```

```
3            318                63          5.0                    31:43
4            423                85          5.0                    29:30

  home_id away_id
0     PIT     CLE
1     PHI     JAX
2     SEA      GB
3     LAR     MIN
4     BAL     CIN
```

[59]:
```
# adding team ids to the game data dataset
game_data["home_id"] = game_data["team_home"].apply(get_team_id)
game_data["away_id"] = game_data["team_away"].apply(get_team_id)

game_data.head()
```

[59]:
```
  schedule_date  schedule_season schedule_week            team_home  \
0    2010-09-09             2010             1    New Orleans Saints
1    2010-09-12             2010             1         Buffalo Bills
2    2010-09-12             2010             1         Chicago Bears
3    2010-09-12             2010             1        Houston Texans
4    2010-09-12             2010             1  Jacksonville Jaguars

   score_home  score_away           team_away team_favorite_id  \
0        14.0         9.0   Minnesota Vikings               NO
1        10.0        15.0      Miami Dolphins              MIA
2        19.0        14.0        Detroit Lions             CHI
3        34.0        24.0  Indianapolis Colts              IND
4        24.0        17.0      Denver Broncos              JAX

   spread_favorite               stadium  weather_temperature  \
0             -5.0   Louisiana Superdome                 72.0
1             -3.0  Ralph Wilson Stadium                 64.0
2             -6.5         Soldier Field                 75.0
3             -1.0       Reliant Stadium                 89.0
4             -3.0        EverBank Field                 91.0

   weather_wind_mph home_id away_id
0               0.0      NO     MIN
1               7.0     BUF     MIA
2               1.0     CHI     DET
3               5.0     HOU     IND
4               1.0     JAX     DEN
```

1.2 Joining Data

[60]:
```
# merge the box scores and game data using inner join on date and home/away id
box_game_data_merged = pd.merge(box_scores, game_data, how="inner",␣
 ↪left_on=["date", "home_id", "away_id"], right_on=["schedule_date",␣
 ↪"home_id", "away_id"])
box_game_data_merged.head()
```

[60]:
```
         date      visitor          home  visitor_score  home_score  \
0  2014-09-07    Cleveland    Pittsburgh             27          30
1  2014-09-07  Jacksonville  Philadelphia             17          34
2  2014-09-04    Green Bay       Seattle             16          36
3  2014-09-07    Minnesota     St. Louis             34           6
4  2014-09-07   Cincinnati     Baltimore             23          16

   visitor_first_downs  visitor_net_yards  visitor_total_plays  \
0                   23                389                   64
1                   18                306                   70
2                   19                255                   57
3                   18                355                   57
4                   16                380                   64

   visitor_avg_gain visitor_time_of_possession  …  schedule_week  \
0               6.1                      27:33  …              1
1               4.4                      29:14  …              1
2               4.5                      26:40  …              1
3               6.2                      28:17  …              1
4               5.9                      30:30  …              1

             team_home  score_home  score_away           team_away  \
0  Pittsburgh Steelers        30.0        27.0    Cleveland Browns
1  Philadelphia Eagles        34.0        17.0  Jacksonville Jaguars
2     Seattle Seahawks        36.0        16.0    Green Bay Packers
3       St. Louis Rams         6.0        34.0    Minnesota Vikings
4     Baltimore Ravens        16.0        23.0   Cincinnati Bengals

  team_favorite_id spread_favorite                   stadium  \
0              PIT            -5.5               Heinz Field
1              PHI           -10.0  Lincoln Financial Field
2              SEA            -4.5         CenturyLink Field
3              LAR            -3.0         Edward Jones Dome
4              BAL            -1.0          M&T Bank Stadium

   weather_temperature weather_wind_mph
0                 72.0              6.0
1                 80.0              6.0
2                 70.0              5.0
```

```
3                  72.0              0.0
4                  78.0              0.0

[5 rows x 29 columns]
```

[61]:
```
# sort the merged dataset by date
box_game_data_merged = box_game_data_merged.sort_values(by=["date"]).
 ↪reset_index(drop=True)
box_game_data_merged.head()
```

[61]:
```
         date       visitor          home  visitor_score  home_score  \
0  2010-09-09     Minnesota   New Orleans              9          14
1  2010-09-12  Indianapolis       Houston             24          34
2  2010-09-12       Detroit       Chicago             14          19
3  2010-09-12       Arizona     St. Louis             17          13
4  2010-09-12      Carolina     NY Giants             18          31

   visitor_first_downs  visitor_net_yards  visitor_total_plays  \
0                   12                253                   51
1                   25                463                   69
2                   13                168                   57
3                   21                378                   64
4                   14                237                   63

   visitor_avg_gain visitor_time_of_possession  …  schedule_week  \
0               5.0                      26:17  …              1
1               6.7                      29:07  …              1
2               2.9                      25:18  …              1
3               5.9                      27:09  …              1
4               3.8                      25:21  …              1

            team_home  score_home  score_away            team_away  \
0  New Orleans Saints        14.0         9.0    Minnesota Vikings
1      Houston Texans        34.0        24.0   Indianapolis Colts
2        Chicago Bears       19.0        14.0        Detroit Lions
3      St. Louis Rams        13.0        17.0   Arizona Cardinals
4     New York Giants        31.0        18.0   Carolina Panthers

  team_favorite_id spread_favorite               stadium  weather_temperature  \
0               NO            -5.0  Louisiana Superdome                  72.0
1              IND            -1.0       Reliant Stadium                  89.0
2              CHI            -6.5         Soldier Field                  75.0
3              ARI            -3.0     Edward Jones Dome                  72.0
4              NYG            -6.0       MetLife Stadium                  65.0

  weather_wind_mph
0              0.0
```

```
1              5.0
2              1.0
3              0.0
4              1.0

[5 rows x 29 columns]
```

[62]:
```
# drop unnecessary columns
box_game_data_merged = box_game_data_merged.drop(columns=["schedule_date",␣
↪"visitor", "home", "visitor_score", "home_score"])
box_game_data_merged.head()
```

[62]:
```
         date  visitor_first_downs  visitor_net_yards  visitor_total_plays  \
0  2010-09-09                   12                253                   51
1  2010-09-12                   25                463                   69
2  2010-09-12                   13                168                   57
3  2010-09-12                   21                378                   64
4  2010-09-12                   14                237                   63

   visitor_avg_gain visitor_time_of_possession  home_first_downs  \
0               5.0                      26:17                18
1               6.7                      29:07                23
2               2.9                      25:18                23
3               5.9                      27:09                20
4               3.8                      25:21                21

   home_net_yards  home_total_plays  home_avg_gain  … schedule_week  \
0             308                62            5.0  …             1
1             355                61            5.8  …             1
2             463                70            6.6  …             1
3             325                81            4.0  …             1
4             376                67            5.6  …             1

            team_home score_home  score_away           team_away  \
0  New Orleans Saints       14.0         9.0   Minnesota Vikings
1      Houston Texans       34.0        24.0  Indianapolis Colts
2       Chicago Bears       19.0        14.0        Detroit Lions
3      St. Louis Rams       13.0        17.0   Arizona Cardinals
4     New York Giants       31.0        18.0   Carolina Panthers

  team_favorite_id  spread_favorite             stadium weather_temperature  \
0               NO             -5.0  Louisiana Superdome                72.0
1              IND             -1.0      Reliant Stadium                89.0
2              CHI             -6.5        Soldier Field                75.0
3              ARI             -3.0     Edward Jones Dome                72.0
4              NYG             -6.0      MetLife Stadium                65.0
```

```
  weather_wind_mph
0              0.0
1              5.0
2              1.0
3              0.0
4              1.0

[5 rows x 24 columns]
```

[63]:
```
# convert time of possession to a float for minutes
def convert_time_to_float(time):
  if time == "None":
    return 0
  else:
    time_split = time.split(":")
    return float(time_split[0]) + float(time_split[1])/60
```

[64]:
```
box_game_data_merged["visitor_time_of_possession"] =␣
 ↪box_game_data_merged["visitor_time_of_possession"].
 ↪apply(convert_time_to_float)
box_game_data_merged["home_time_of_possession"] =␣
 ↪box_game_data_merged["home_time_of_possession"].apply(convert_time_to_float)
```

[65]:
```
def get_winner_id(row):
    if row["score_home"] > row["score_away"]:
      return 0
    elif row["score_home"] < row["score_away"]:
      return 1
    else:
      return 2
```

[66]:
```
# apply the get_winner_id function to the merged dataset
box_game_data_merged["winner_id"] = box_game_data_merged.apply(get_winner_id,␣
 ↪axis=1)
box_game_data_merged.head()
```

[66]:
```
         date  visitor_first_downs  visitor_net_yards  visitor_total_plays  \
0  2010-09-09                   12                253                   51
1  2010-09-12                   25                463                   69
2  2010-09-12                   13                168                   57
3  2010-09-12                   21                378                   64
4  2010-09-12                   14                237                   63

   visitor_avg_gain  visitor_time_of_possession  home_first_downs  \
0               5.0                   26.283333                18
1               6.7                   29.116667                23
2               2.9                   25.300000                23
```

```
3              5.9                   27.150000                20
4              3.8                   25.350000                21

   home_net_yards  home_total_plays  home_avg_gain  …            team_home  \
0             308                62            5.0  …   New Orleans Saints
1             355                61            5.8  …       Houston Texans
2             463                70            6.6  …        Chicago Bears
3             325                81            4.0  …       St. Louis Rams
4             376                67            5.6  …      New York Giants

  score_home score_away            team_away team_favorite_id spread_favorite  \
0       14.0        9.0    Minnesota Vikings               NO            -5.0
1       34.0       24.0   Indianapolis Colts              IND            -1.0
2       19.0       14.0        Detroit Lions              CHI            -6.5
3       13.0       17.0    Arizona Cardinals              ARI            -3.0
4       31.0       18.0    Carolina Panthers              NYG            -6.0

               stadium  weather_temperature weather_wind_mph winner_id
0  Louisiana Superdome                 72.0              0.0         0
1       Reliant Stadium                 89.0              5.0         0
2         Soldier Field                 75.0              1.0         0
3     Edward Jones Dome                 72.0              0.0         1
4       MetLife Stadium                 65.0              1.0         0

[5 rows x 25 columns]
```

[67]:
```python
box_game_data_merged.columns
```

[67]:
```
Index(['date', 'visitor_first_downs', 'visitor_net_yards',
       'visitor_total_plays', 'visitor_avg_gain', 'visitor_time_of_possession',
       'home_first_downs', 'home_net_yards', 'home_total_plays',
       'home_avg_gain', 'home_time_of_possession', 'home_id', 'away_id',
       'schedule_season', 'schedule_week', 'team_home', 'score_home',
       'score_away', 'team_away', 'team_favorite_id', 'spread_favorite',
       'stadium', 'weather_temperature', 'weather_wind_mph', 'winner_id'],
      dtype='object')
```

Most Important Columns: ‘home_id’, ‘away_id’, ‘visitor_net_yards’, ‘visitor_time_of_possession’, ‘home_net_yards’, ‘home_time_of_possession’, ‘score_home’, ‘score_away’, ‘stadium’

1.3 Train Model

[68]:
```python
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsRegressor
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import make_column_transformer
```

[69]:
```python
df_box_game = box_game_data_merged[:1500].copy()

X_train = df_box_game[['home_id', 'away_id', 'visitor_net_yards',␣
 ↪'visitor_time_of_possession', 'home_net_yards', 'home_time_of_possession',␣
 ↪'score_home', 'score_away']]
y_train = df_box_game["winner_id"]

ct = make_column_transformer(
    (OneHotEncoder(), ['home_id', 'away_id']),
    remainder='passthrough'
)

pipeline = make_pipeline(
    ct,
    StandardScaler(with_mean=False),
    KNeighborsRegressor(n_neighbors=5)
)

grid_search = GridSearchCV(
    pipeline,
    param_grid={
        "kneighborsregressor__n_neighbors": range(1, 20),
        "kneighborsregressor__metric": ["euclidean", "manhattan"]
    },
    scoring="neg_root_mean_squared_error",
    cv=10
)

grid_search.fit(X_train, y_train)
grid_search.best_estimator_
```

[69]:
```
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(remainder='passthrough',
                                   transformers=[('onehotencoder',
                                                  OneHotEncoder(),
                                                  ['home_id', 'away_id'])])),
                ('standardscaler', StandardScaler(with_mean=False)),
                ('kneighborsregressor',
                 KNeighborsRegressor(metric='manhattan', n_neighbors=10))])
```

[70]:
```python
df_cv_results = pd.DataFrame(grid_search.cv_results_)
df_cv_results.head()
```

```
[70]:    mean_fit_time  std_fit_time  mean_score_time  std_score_time  \
      0       0.003609      0.000218         0.006066        0.001064   
      1       0.003842      0.000507         0.007651        0.003816   
      2       0.003308      0.000155         0.005224        0.000894   
      3       0.003216      0.000214         0.004503        0.000533   
      4       0.003102      0.000132         0.004607        0.000399   

        param_kneighborsregressor__metric param_kneighborsregressor__n_neighbors  \
      0                         euclidean                                      1   
      1                         euclidean                                      2   
      2                         euclidean                                      3   
      3                         euclidean                                      4   
      4                         euclidean                                      5   

                                                    params  split0_test_score  \
      0  {'kneighborsregressor__metric': 'euclidean', '…          -0.577350   
      1  {'kneighborsregressor__metric': 'euclidean', '…          -0.484768   
      2  {'kneighborsregressor__metric': 'euclidean', '…          -0.457044   
      3  {'kneighborsregressor__metric': 'euclidean', '…          -0.444410   
      4  {'kneighborsregressor__metric': 'euclidean', '…          -0.435125   

         split1_test_score  split2_test_score  split3_test_score  split4_test_score  \
      0          -0.559762          -0.605530          -0.583095          -0.571548   
      1          -0.479583          -0.503322          -0.483046          -0.509902   
      2          -0.470618          -0.453791          -0.479969          -0.480740   
      3          -0.451848          -0.456435          -0.459619          -0.456435   
      4          -0.435125          -0.430813          -0.430813          -0.449296   

         split5_test_score  split6_test_score  split7_test_score  split8_test_score  \
      0          -0.541603          -0.565685          -0.529150          -0.535413   
      1          -0.426224          -0.509902          -0.454606          -0.433974   
      2          -0.414550          -0.495162          -0.426875          -0.407340   
      3          -0.411805          -0.483477          -0.408758          -0.420813   
      4          -0.399333          -0.453725          -0.390555          -0.402989   

         split9_test_score  mean_test_score  std_test_score  rank_test_score  
      0          -0.547723        -0.561686        0.022561               38  
      1          -0.474342        -0.475967        0.028087               36  
      2          -0.446385        -0.453247        0.028090               34  
      3          -0.441116        -0.443472        0.022350               33  
      4          -0.451073        -0.427885        0.021481               31  
```

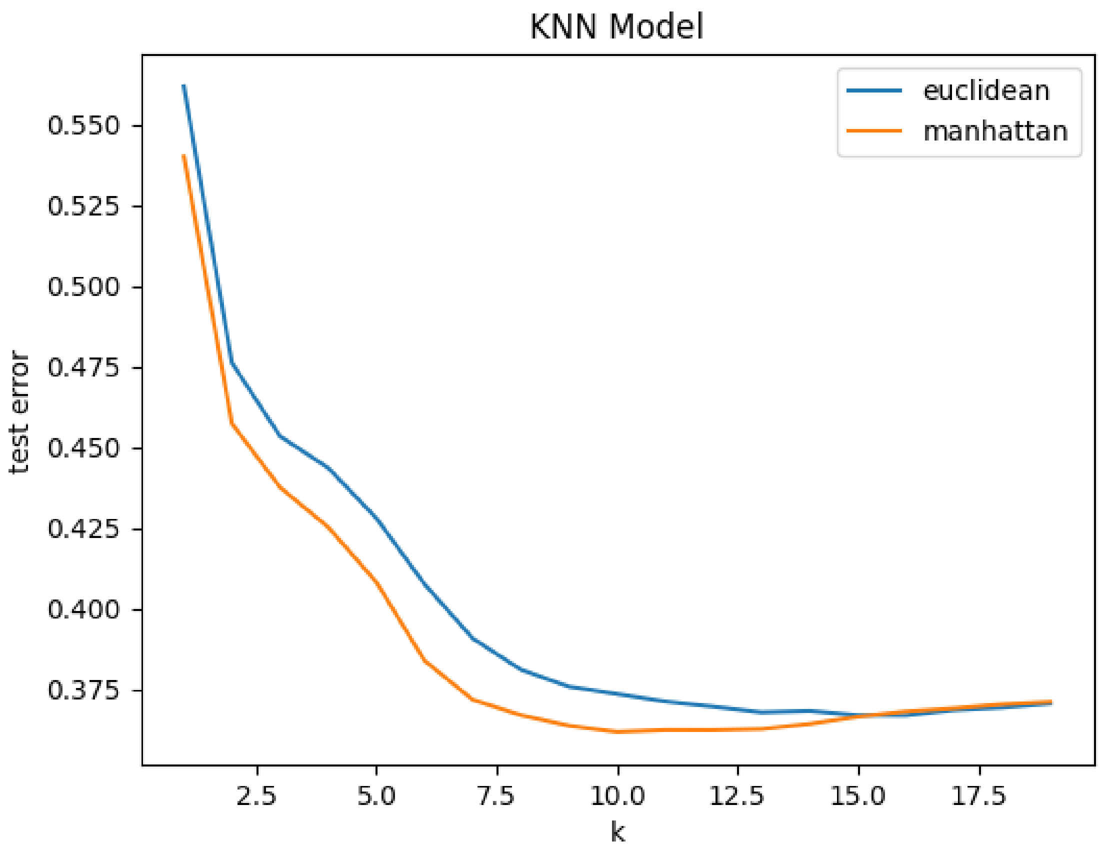
```
[71]: df_cv_results["param_kneighborsregressor__n_neighbors"] =␣
        ↪df_cv_results["param_kneighborsregressor__n_neighbors"].astype("int")

      df_cv_results.set_index("param_kneighborsregressor__n_neighbors", inplace =␣
        ↪True)
```

```
[80]: df_cv_results["pos_mean_test_score"] = -df_cv_results["mean_test_score"]

      (df_cv_results.
       ↪groupby("param_kneighborsregressor__metric")["pos_mean_test_score"]).plot.
       ↪line(xlabel = "k", ylabel = "test error",
                                                                                  ␣
       ↪                title = "KNN Model",
                                                                                  ␣
       ↪                legend = True)
```

```
[80]: param_kneighborsregressor__metric
      euclidean    Axes(0.125,0.11;0.775x0.77)
      manhattan    Axes(0.125,0.11;0.775x0.77)
      Name: pos_mean_test_score, dtype: object
```



```
[73]: cv_errs = -cross_val_score(grid_search.best_estimator_, X=X_train.
       ↪fillna(X_train.mean()),
                                 y=y_train,
                                 scoring="neg_root_mean_squared_error", cv=10)
```

```
cv_errs.mean()
```

```
/var/folders/q8/mqm68gfx7pjfpqftf7y_v6140000gn/T/ipykernel_10674/2256501085.py:1
: FutureWarning: The default value of numeric_only in DataFrame.mean is
deprecated. In a future version, it will default to False. In addition,
specifying 'numeric_only=None' is deprecated. Select only valid columns or
specify the value of numeric_only to silence this warning.
  cv_errs = -cross_val_score(grid_search.best_estimator_,
X=X_train.fillna(X_train.mean()),
```

```
[73]: 0.3616783691950907
```

```
[74]: df_box_game_test = box_game_data_merged[1500:].copy()
```

```
[75]: y_new = pd.Series(
          grid_search.best_estimator_.predict(X=df_box_game_test[['home_id',␣
       ↪'away_id', 'visitor_net_yards', 'visitor_time_of_possession',␣
       ↪'home_net_yards', 'home_time_of_possession', 'score_home', 'score_away']]),
          index=df_box_game_test.index
      )

      y_new
```

```
[75]: 1500    0.2
      1501    0.0
      1502    0.4
      1503    0.6
      1504    0.0
             …
      1912    0.6
      1913    0.3
      1914    0.0
      1915    0.0
      1916    0.7
      Length: 417, dtype: float64
```

```
[79]: pred_vs_actual = pd.DataFrame({
          "Winner_pred": y_new,
          "Winner_actual": df_box_game_test["winner_id"],
          "Home_Team": df_box_game_test["home_id"],
          "Away_Team": df_box_game_test["away_id"]
      })

      pred_vs_actual["Winner_pred"] = pred_vs_actual["Winner_pred"].apply(lambda x: 0␣
       ↪if x < 0.5 else 1)
      pred_vs_actual.count()
```

[79]:
```
Winner_pred      417
Winner_actual    417
Home_Team        417
Away_Team        417
dtype: int64
```

[78]:
```
pred_vs_actual[pred_vs_actual["Winner_pred"] ==␣
↪pred_vs_actual["Winner_actual"]].count()
```

[78]:
```
Winner_pred      340
Winner_actual    340
Home_Team        340
Away_Team        340
dtype: int64
```