data_cleaning

April 26, 2024

[ ]:
```
import pandas as pd
```

Join circuits and race_circuits to get turns and track length in one file

[ ]:
```
circuits = "./raw_data/circuits.csv"
race_cicuits = "./in_progress_data/race_circuits.csv"
```

[ ]:
```
circuits_df = pd.read_csv(circuits)
circuits_df.columns = circuits_df.columns.str.lower()
circuits_df.columns
```

[ ]:
```
Index(['circuitid', 'circuitref', 'name', 'location', 'country', 'lat', 'lng',
       'alt', 'url'],
      dtype='object')
```

[ ]:
```
race_circuits_df = pd.read_csv(race_cicuits)
race_circuits_df.columns = race_circuits_df.columns.str.lower()
race_circuits_df.columns
```

[ ]:
```
Index(['circuit', 'map', 'type', 'direction', 'location', 'country',
       'last length used', 'turns', 'grands prix', 'season(s)',
       'grands prix held'],
      dtype='object')
```

[ ]:
```
replace = {"United States": "USA", "United Arab Emirates": "UAE", "United␣
 ↪Kingdom":"UK"}
race_circuits_df['country'] = race_circuits_df['country'].replace(replace)
```

[ ]:
```
final_circuits_df = pd.merge(circuits_df, race_circuits_df[['location',␣
 ↪'country','last length used', 'turns']], on=['location', 'country'],␣
 ↪how='left')
```

[ ]:
```
final_circuits_df['last length used'] = final_circuits_df['last length used'].
 ↪str.replace('km', '')
final_circuits_df['last length used'] = pd.to_numeric(final_circuits_df['last␣
 ↪length used'])
final_circuits_df.head
```

[ ]:
```
<bound method NDFrame.head of     circuitid   circuitref
name  \
0           1  albert_park        Albert Park Grand Prix Circuit
1           2       sepang          Sepang International Circuit
2           3      bahrain         Bahrain International Circuit
3           4    catalunya        Circuit de Barcelona-Catalunya
4           5     istanbul                          Istanbul Park
..        …           …                                     …
74         75     portimao    Autódromo Internacional do Algarve
75         76      mugello  Autodromo Internazionale del Mugello
76         77       jeddah               Jeddah Corniche Circuit
77         78       losail          Losail International Circuit
78         79        miami         Miami International Autodrome

        location       country       lat        lng  alt  \
0      Melbourne     Australia -37.84970  144.96800   10
1   Kuala Lumpur      Malaysia   2.76083  101.73800   18
2         Sakhir       Bahrain  26.03250   50.51060    7
3       Montmeló         Spain  41.57000    2.26111  109
4       Istanbul        Turkey  40.95170   29.40500  130
..           …           …       …         …  …
74      Portimão      Portugal  37.22700   -8.62670  108
75       Mugello         Italy  43.99750   11.37190  255
76        Jeddah  Saudi Arabia  21.63190   39.10440   15
77     Al Daayen         Qatar  25.49000   51.45420   \N
78         Miami           USA  25.95810  -80.23890   \N

                                              url  last length used turns
0   http://en.wikipedia.org/wiki/Melbourne_Grand_P…             5.278    16
1   http://en.wikipedia.org/wiki/Sepang_Internatio…               NaN   NaN
2   http://en.wikipedia.org/wiki/Bahrain_Internati…             5.412    15
3   http://en.wikipedia.org/wiki/Circuit_de_Barcel…             4.657    14
4          http://en.wikipedia.org/wiki/Istanbul_Park             5.338    14
..                                                …               …   …
74  http://en.wikipedia.org/wiki/Algarve_Internati…             4.653    15
75       http://en.wikipedia.org/wiki/Mugello_Circuit                NaN   NaN
76  http://en.wikipedia.org/wiki/Jeddah_Street_Cir…             6.174    27
77  http://en.wikipedia.org/wiki/Losail_Internatio…               NaN   NaN
78  http://en.wikipedia.org/wiki/Miami_Internation…               NaN   NaN

[79 rows x 11 columns]>
```

[ ]:
```
#final_circuits_df.to_csv("./in_progress_data/final_circuits.csv", index=False)
```

The above is commented out so that that file is no longer touched here after.

Filter `races.csv` to only have races from 2018

[ ]:
```
races = "./raw_data/races.csv"
races_df = pd.read_csv(races)
races_df.columns = races_df.columns.str.lower()

races_df = races_df.drop(columns=['fp1_date', 'fp1_time','fp2_date',␣
 ↪'fp2_time', 'fp3_date', 'fp3_time', 'quali_date', 'quali_time',␣
 ↪'sprint_date', 'sprint_time',])
races_df = races_df[races_df['year'] >= 2018]
raceid = races_df['raceid'].unique()
print(raceid)

races_df.to_csv("./in_progress_data/final_races.csv", index=False)
```

```
[ 989  990  991  992  993  994  995  996  997  998  999 1000 1001 1002
 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016
 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030
 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044
 1045 1046 1047 1053 1074 1052 1051 1054 1055 1056 1057 1058 1059 1060
 1061 1062 1063 1064 1065 1066 1067 1069 1070 1071 1072 1073 1075 1076
 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1091
 1092 1093 1094 1095 1096 1098 1099 1100 1101 1102 1104 1105 1106 1107
 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120]
```

Filter `lap_times.csv` to only have the required lap_times (i.e lap times from races fromm 2018 to 2023.)

[ ]:
```
lap_times = "./raw_data/lap_times.csv"
lap_times_df = pd.read_csv(lap_times)
lap_times_df.columns = lap_times_df.columns.str.lower()

lap_times_df = lap_times_df[lap_times_df['raceid'].isin(raceid)]
laptime_raceid = lap_times_df['raceid'].unique()
print(laptime_raceid)

lap_times_df = lap_times_df.rename(columns={'time': 'lap_time'})
lap_times_df.columns

lap_times_df.to_csv("./in_progress_data/final_laptimes.csv", index=False)
```

```
[ 989  990  991  992  993  994  995  996  997  998  999 1000 1001 1002
 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016
 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030
 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044
 1045 1046 1047 1052 1053 1054 1055 1056 1057 1059 1058 1060 1061 1062
 1063 1064 1065 1066 1067 1069 1070 1071 1051 1072 1073 1074 1075 1076
 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1091
 1092 1093 1094 1095 1096 1098 1099 1100 1101 1102 1104 1105 1106 1107
 1108 1109 1110]
```

Change the name of the column in `drivers.csv` to avoid duplicates when merging to form final table

[ ]:
```
drivers = "./raw_data/drivers.csv"
drivers_df = pd.read_csv(drivers)
drivers_df.columns = drivers_df.columns.str.lower()

drivers_df = drivers_df.rename(columns={'url': 'driver_url'})
drivers_df.columns

drivers_df.to_csv("./in_progress_data/final_drivers.csv", index=False)
```

Filter the `pit_stops.csv` to only have the required pit_stops (i.e pit stops from races fromm 2018 to 2023.) and clean up the dataset

[ ]:
```
pit_stops = "./raw_data/pit_stops.csv"
pit_stops_df = pd.read_csv(pit_stops)
pit_stops_df.columns = pit_stops_df.columns.str.lower()

pit_stops_df = pit_stops_df[pit_stops_df['raceid'].isin(raceid)]

pit_stops_df = pit_stops_df.drop(columns=["time"])
pit_stops_df = pit_stops_df.rename(columns={'milliseconds':␣
 ↪'pitstop_milliseconds','lap': 'pit_lap'})
pit_stops_df.columns

pit_stops_df.to_csv("./in_progress_data/final_pitstops.csv", index=False)
```

Changing the columns in `weather.csv` to match with other datasets as keys.

[ ]:
```
weather_df = pd.read_csv('./in_progress_data/weather.csv')
weather_df.columns = weather_df.columns.str.lower()

weather_df = weather_df.drop(columns=['time'])
weather_df = weather_df.rename(columns={'round number': 'round'})
weather_df.columns

weather_df.to_csv('./in_progress_data/weather.csv', index=False)
```

Converting all columns to lower case

[ ]:
```
tire_df = pd.read_csv('./in_progress_data/tire.csv')
tire_df.columns = tire_df.columns.str.lower()
tire_df.to_csv('./in_progress_data/tire.csv', index=False)
```

Creating the final dataset to be used in the model selection and training

[ ]:
```
races_df = pd.read_csv('./in_progress_data/final_races.csv')
tire_df = pd.read_csv('./in_progress_data/tire.csv')
```

```
weather_df = pd.read_csv('./in_progress_data/weather.csv')
circuits_df = pd.read_csv('./in_progress_data/final_circuits.csv')
drivers_df = pd.read_csv('./in_progress_data/final_drivers.csv')
lap_times_df = pd.read_csv('./in_progress_data/final_laptimes.csv')
pit_stop_df = pd.read_csv('./in_progress_data/final_pitstops.csv')
```

Dropping the unwanted columns

[ ]:
```
races_df = races_df.drop(columns=['name', 'date', 'time', 'url'])
circuits_df = circuits_df.drop(columns=['circuitref', 'url'])
drivers_df = drivers_df.drop(columns=['driverref','number', 'forename',␣
 ↪'surname', 'dob', 'nationality', 'driver_url'])
lap_times_df = lap_times_df.drop(columns=['lap_time', 'position'])
pit_stop_df = pit_stop_df.drop(columns=['stop','duration',␣
 ↪'pitstop_milliseconds'])
pit_stop_df = pit_stop_df.rename(columns={'pit_lap': 'lap'})
```

[ ]:
```
merged_df = pd.merge(lap_times_df, pit_stop_df, on=['raceid', 'driverid',␣
 ↪'lap'], how='left', indicator=True)
# Filter out rows where there is a match between pitstop and laptime data
filtered_df = merged_df[merged_df['_merge'] == 'left_only'].drop('_merge',␣
 ↪axis=1)
```

[ ]:
```
# Convert 'rainfall' column to numeric (True=1, False=0)
weather_df['rainfall'] = weather_df['rainfall'].astype(int)

# Group by 'year' and 'round' and calculate the mean for each group
average_weather_df = weather_df.groupby(['year', 'round']).mean().reset_index()
```

[ ]:
```
data = pd.merge(races_df, tire_df, on=['year', 'round'])
data = pd.merge(data, average_weather_df, on=['year', 'round'])
data = pd.merge(data, drivers_df, on=['code'])
data = pd.merge(data, lap_times_df, on=['raceid', 'driverid'])
data.reset_index(drop=True, inplace=True)
```

[ ]:
```
data['avg_lap_time'] = data.groupby(['raceid', 'driverid',␣
 ↪'stint'])['milliseconds'].transform('mean')
data = data.drop(columns=['milliseconds','lap', 'code'])
data = data.drop_duplicates()
```

[ ]:
```
data.columns
```

[ ]:
```
Index(['raceid', 'year', 'round', 'circuitid', 'stint', 'compound',
       'stint start lap', 'stint end lap', 'stint length', 'airtemp',
       'humidity', 'pressure', 'rainfall', 'tracktemp', 'winddirection',
       'windspeed', 'driverid', 'avg_lap_time'],
      dtype='object')
```

[ ]:
```
data.to_csv('./final_data/final_data.csv', index=False)
```