Project #1 – Machine Learning

by Rami Abu Ahmad & Stepan Kostyukov

INFO3142 – Emerging Technologies

Prof. Osam Ali

November 5, 2023

Table of contents

Overview of the forecasting problem

"Races are won at the track. Championships are won at the factory." - Mercedes (2019)

In this forecasting problem, the objective is to predict the average number of Formula 1 races a driver must win to become a champion in a season with 'n' races. Three machine learning algorithms will be employed: Linear Regression, Support Vector Regression, and Neural Networks. The dataset will comprise historical records of past racing seasons, including the number of races and the corresponding number of wins required to clinch the championship. Each algorithm will be trained on this dataset to learn the underlying patterns and relationships between the variables. Performance metrics such as Mean Absolute Error (MAE) and R-squared are utilized to evaluate the models' accuracy and generalization capabilities. This assignment aims to compare the predictive power of these three algorithms in the context of determining the average number of victories needed for championship success in motorsports.

Dataset

To solve the chosen forecasting problem data from the following open-source dataset was utilised:
https://www.kaggle.com/datasets/rohanrao/formula-1-world-championship-1950-2020/data?select=races.csv
Even though the initial dataset consisted of *"all information on the Formula 1 races, drivers, constructors, qualifying, circuits, lap times, pit stops, championships from 1950 till the latest 2023 season."*, only part of it was used to create a working dataset. First of all, only seasons of 1950-2022 were utilized for this project, as the current year's season is still in progress with multiple Grand Prix to complete, thus it is not possible to use it for solving our problem. Secondly, only 3 out of 14 available CSV files were needed for our task, which includes races.csv, drivers.csv, and driver_standings.csv. Here's a short overview of each used CSV file.

races.csv

Contains information about every race in Formula 1 history. The following columns were used in the project:

Column name	Notes
raceId	PK
date	Used to find the last race of each season. And used to retrieve the year when the race took place to use it in model training

drivers.csv

Contains information about every driver in Formula 1 history. The following columns were used in the project:

Column name	Notes
driverId	PK
driverRef	Used to identify drivers by name, just so that the outputs are human-readable

driver_standings.csv

Contains information about every driver's standings in the championship after every race in Formula 1 history. The following columns were used in the project:

Column name	Notes
raceId	FK
driverId	FK
position	Used to find the champion after the last race of each season
wins	Number of wins of each driver after each completed race. Used in model training

The 3 previous files were used to create a working dataset - my_dataset.csv with the following structure:

my_dataset.csv

Column name	Notes
year	Year of the season
driverRef	Name of the champions
wins	Number of wins the champion had after the last race of the season
racesCount	Total number of races in the corresponding year

The following script generates the resulting my_dataset.csv file:

```
###################################
# Prepare and save a work dataset #
###################################
import pandas as pd

races = pd.read_csv("data/races.csv")
drivers = pd.read_csv("data/drivers.csv")
driver_standings = pd.read_csv("data/driver_standings.csv")

# Join driver standings and drivers into single DataFrame
df = pd.merge(driver_standings[["raceId", "driverId", "position", "wins"]],
drivers[["driverId", "driverRef"]], on = "driverId", how = "left")

# Get year of each season, and exclude current year's season as it's not
finished yet
df = pd.merge(df, races[["raceId", "date"]], on = "raceId", how = "left")
df["date"] = pd.to_datetime(df["date"])
df = df[df["date"].dt.year < 2023]

# Get only the champion's results
df = df[(df["position"] == 1) & (df["wins"] > 0)]

# Get total number of races per each year
df["year"] = df["date"].dt.year
races_count = df.groupby("year")["raceId"].count()
races_count = races_count.reset_index().rename(columns = {"raceId":
"racesCount"})

# Get only the results after the final race of the season
final_races = df.groupby("year")["date"].max()
final_races = final_races.reset_index()

df = df.merge(final_races["date"], left_on=df["date"], right_on =
final_races["date"])
df = pd.merge(df, races_count, on = "year", how = "left")
df = df.sort_values(by="year")

# Save the resulting DataFrame into a separate my_dataset.csv file for
easier use later on
df[["year", "driverRef", "wins",
"racesCount"]].to_csv("data/my_dataset.csv")
print("DataFrame was saved into data/my_dataset.csv!")
```

Selected algorithms

Linear Regression

Linear Regression is a fundamental algorithm in machine learning that establishes a linear relationship between input features and the target variable. In the context of Formula 1, this algorithm could model the relationship between factors like driver performance metrics, team investments, and race conditions against the number of wins required for championship victory. By fitting a linear model, we can gain insights into how these variables contribute to a driver's path to becoming a champion.

Support Vector Regression

Support Vector Regression is a variant of the Support Vector Machine (SVM) algorithm, adapted for regression tasks. In the world of Formula 1, SVR could be employed to capture complex non-linear relationships between input features and the target variable. For instance, it could help discern how factors like track characteristics, weather conditions, and driver experience interact to determine the number of wins needed for championship glory.

Neural Networks

Neural Networks are highly versatile and powerful models capable of capturing intricate patterns in data. In the context of Formula 1, a neural network could be designed to process a wide array of features, including driver statistics, team performance, and race-specific details. By leveraging hidden layers and non-linear activation functions, the neural network can uncover complex relationships that may not be readily apparent through traditional methods. This can provide valuable insights into the nuanced factors influencing a driver's championship journey.

These algorithms, when applied to Formula 1 data, offer distinct approaches to understanding the predictive dynamics of championship success. By comparing their performance, we can gain valuable insights into which modeling technique is best suited for forecasting the average number of wins necessary for a driver to become a champion in a given season.

Model training and evaluation

To train and evaluate the models it was decided to use hold-out and split the work dataset into training and testing parts. The training set contains the first 80% of the dataset and the testing set contains the remaining 20%. The “racesCount” column was chosen as X axis and the “wins” column as Y. Here’s how it was implemented in code:

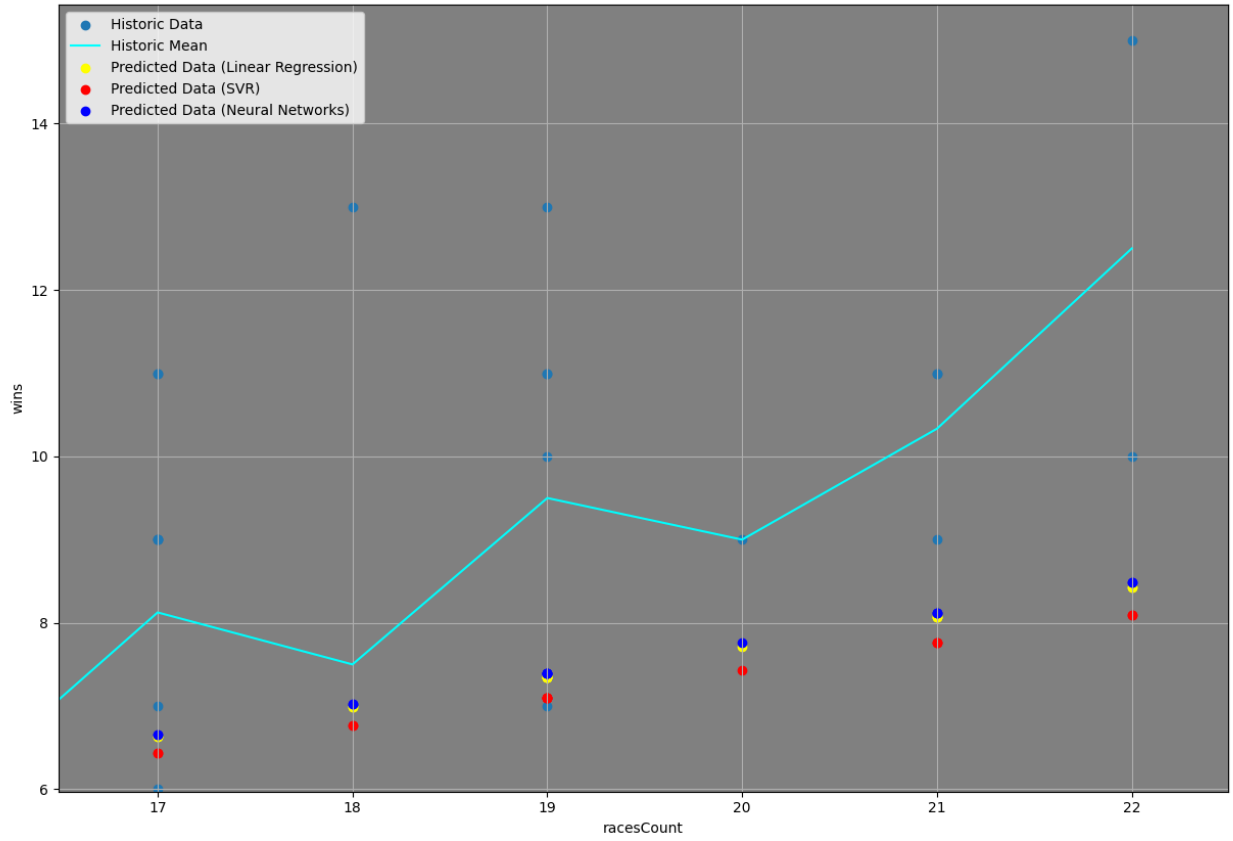
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import mean_absolute_error, mean_squared_error,
r2_score

my_dataset = pd.read_csv("data/my_dataset.csv")

# Split the dataset into train and test parts
train_test_split_ratio = 0.8
train_size = int(len(my_dataset) * train_test_split_ratio)
train_set = my_dataset.iloc[:train_size, :]
test_set = my_dataset.iloc[train_size:, :]


x_train = train_set["racesCount"]
x_train = np.array(x_train)
y_train = train_set["wins"]

x_test = test_set["racesCount"]
x_test = np.array(x_test)
y_test = test_set["wins"]
```

To evaluate the models, mean absolute error, mean squared error, root mean squared error, and R^2 score were calculated for each algorithm.

Accuracy comparison

	Linear Regression	Support Vector Regression	Neural Networks
Mean Absolute Error	2.88	3.02	2.85
Mean Squared Error	10.93	12.1	10.73
Root Mean Squared Error	3.31	3.47	3.28
R^2 score	-0.28	-0.42	-0.26

Here's how they compare on a diagram:



From the values and the figure above we can see that in our case Neural Networks provide the closest to the reality prediction, with Linear Regression following closely behind and SVR being the least accurate model for this task. However, it is also obvious that neither of the models provides accurate enough results to confidently predict the average number of races a driver has to win to become a champion in a season with ***n*** races.

This brings us to the following conclusion: Formula 1 in particular and motorsports, in general, are such complex and unpredictable types of sport, which are affected by myriads of variables, that such basic machine learning algorithms as those used in this project are not nearly sufficient to predict results based on so few parameters accurately.