SC1015 MINI PROJECT

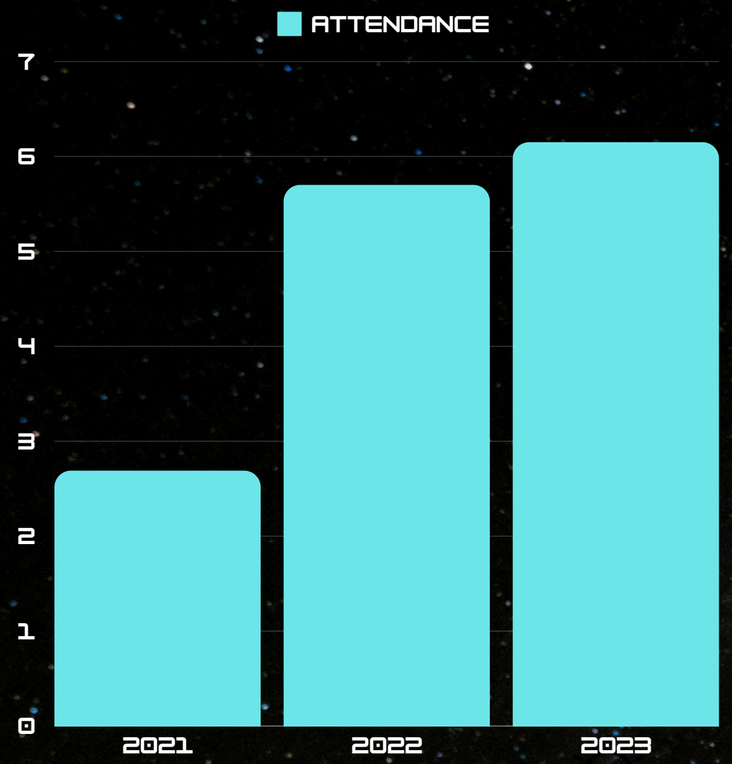
FORMULA 1

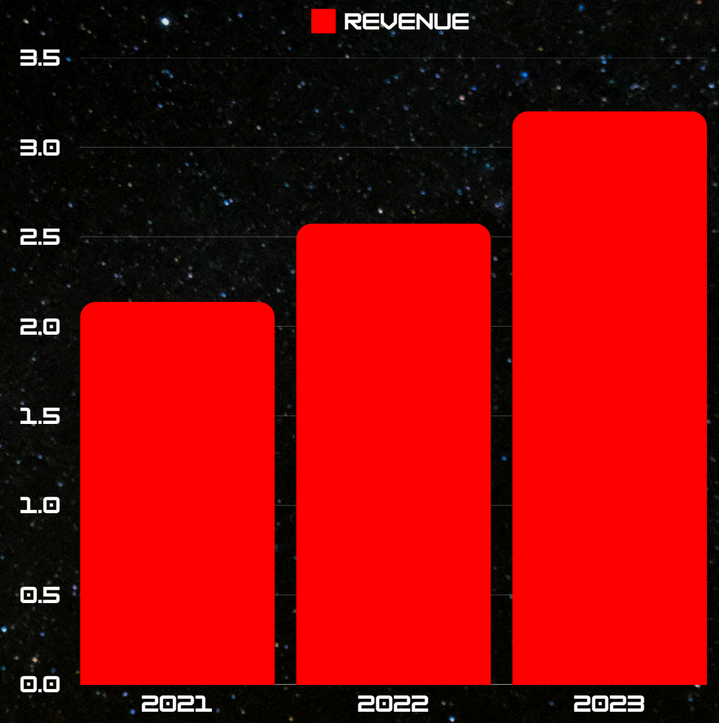
Tan Jun Wei Adison (U2321020A)
Isaac Leow (U2322697H)
Goh Jin Long Abdillah (U2321634L)

FCEA, Lab Group 8

CONTENT PAGE

FORMULA 1 STATS

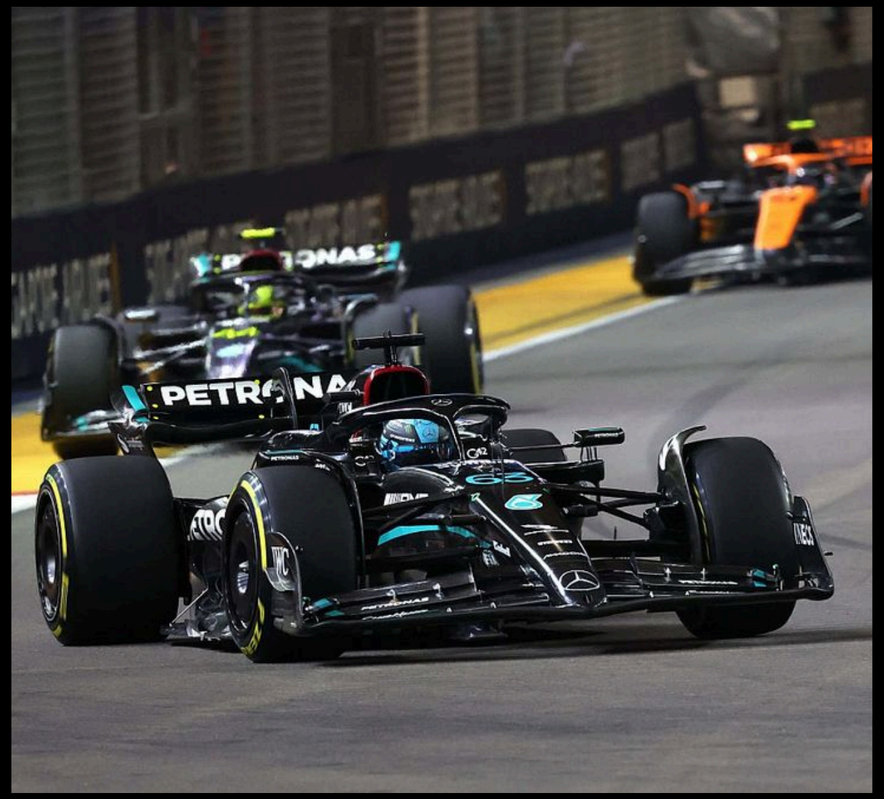




2022 attendance = 5.7 million
2023 attendance = 6.15 million

2021 revenue = $2.136 billion
2022 revenue = $2.573 billion
2023 revenue = $3.2 billion

FORMULA 1 IN SINGAPORE



The only night race in F1 has generated > $1.5 Billion in incremental tourism receipts.

> 550,000 unique international visitors.

All organisations from the GP is Singapore-based.

WHY IS WINNING SO IMPORTANT IN F1?

For eg, in 2024, the most dominant constructor RedBull entered the season with $7.4 M in spending whilst Haas, the last in constructors in 2023, only had $700k to spend.

OUR OBJECTIVES

Leverage on Machine Learning (ML), given race factors to:

- formulate time-crucial **race strategies** to secure **better finishing positions**.
- **improve planning** by teams for future courses of action given current data, such as areas of improvement, to **increase race excitability**.
- gauge **performance and position** of a driver.

ABSTRACTION

PROBLEM STATEMENT

"How can we utilise machine learning to accurately predict the top 5 drivers given factors of a race?"

OUR DATA SET

Mitchell Gleason

Official Formula 1

INITIAL DATA INSIGHTS/ LAPTIME

```
In [82]: print(sgplaptimes.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 14661 entries, 16314 to 532875
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   raceId        14661 non-null  int64 
 1   driverId      14661 non-null  int64 
 2   lap           14661 non-null  int64 
 3   position      14661 non-null  int64 
 4   time          14661 non-null  object
 5   milliseconds  14661 non-null  int64 
dtypes: int64(5), object(1)
memory usage: 801.8+ KB
None
```

```
In [81]: print(sgplaptimes)
```

	raceId	driverId	lap	position	time	milliseconds
16314	854	20	1	1	1:56.910	116910
16315	854	20	2	1	1:55.464	115464
16316	854	20	3	1	1:54.935	114935
16317	854	20	4	1	1:54.261	114261
16318	854	20	5	1	1:53.910	113910
...	...	...	...	...	...	...
532871	1091	848	21	16	2:20.921	140921
532872	1091	848	22	16	2:29.359	149359
532873	1091	848	23	16	2:03.038	123038
532874	1091	848	24	16	2:02.121	122121
532875	1091	848	25	17	2:50.661	170661

[14661 rows x 6 columns]

- Singapore Grand Prix **driver lap times** dataframe.

INITIAL DATA INSIGHTS/PITSTOP

- **Singapore Grand Prix pitstop timing dataframe.**

```
In [4]: sgppits.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 600 entries, 0 to 599
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  600 non-null    int64
 1   duration    600 non-null    float64
 2   stop        600 non-null    int64
 3   driverId    600 non-null    int64
 4   raceId      600 non-null    int64
dtypes: float64(1), int64(4)
memory usage: 23.6 KB
```

```
In [80]: print(sgppits)
```

	duration	stop	driverId	raceId
0	44.170	1	817	2011
1	29.549	1	3	2011
2	30.621	1	4	2011
3	29.610	1	30	2011
4	35.398	1	811	2011
5	30.290	1	13	2011
6	30.142	1	1	2011
7	30.718	1	16	2011
8	30.888	1	813	2011
9	32.827	1	67	2011
10	32.202	1	5	2011
11	36.856	2	811	2011
12	30.462	1	17	2011
13	30.456	1	22	2011
14	32.351	2	13	2011
15	33.456	2	1	2011
16	34.329	1	24	2011
17	30.023	1	20	2011
18	30.160	1	18	2011

INITIAL DATA INSIGHTS/PITSTOP

- **Singapore Grand Prix fastest lap timing dataframe.**

```
In [5]: results.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25840 entries, 0 to 25839
Data columns (total 18 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   resultId         25840 non-null  int64
 1   raceId           25840 non-null  int64
 2   driverId         25840 non-null  int64
 3   constructorId    25840 non-null  int64
 4   number           25840 non-null  object
 5   grid             25840 non-null  int64
 6   position         25840 non-null  object
 7   positionText     25840 non-null  object
 8   positionOrder    25840 non-null  int64
 9   points           25840 non-null  float64
 10  laps             25840 non-null  int64
 11  time             25840 non-null  object
 12  milliseconds     25840 non-null  object
 13  fastestLap       25840 non-null  object
 14  rank             25840 non-null  object
 15  fastestLapTime   25840 non-null  object
 16  fastestLapSpeed  25840 non-null  object
 17  statusId         25840 non-null  int64
dtypes: float64(1), int64(8), object(9)
memory usage: 3.5+ MB
```

```
In [7]: print(sgpresults)
```

	raceId	driverId	constructorId	points	fastestLapTime	position
7830	14	16	10	0.0	1:52.623	\N
7815	14	4	4	6.0	1:48.240	3
7816	14	20	9	5.0	1:48.398	4
7817	14	18	23	4.0	1:48.369	5
7818	14	22	23	3.0	1:48.598	6
7819	14	5	1	2.0	1:49.283	7
7820	14	9	2	1.0	1:48.847	8
7821	14	6	3	0.0	1:49.371	9
7822	14	8	6	0.0	1:48.391	10
7823	14	3	3	0.0	1:48.352	11
7824	14	15	7	0.0	1:48.816	12
7825	14	21	6	0.0	1:49.417	13
7826	14	24	10	0.0	1:49.852	14
7827	14	153	5	0.0	1:52.483	\N
7828	14	67	5	0.0	1:50.636	\N
7829	14	17	9	0.0	1:49.319	\N
7831	14	2	2	0.0	1:51.346	\N
7832	14	154	4	0.0	1:57.192	\N

ABSTRACTION

Our Variables:

- Average **lap time**.
- Average **pitstop** time.
- Driver **fastest lap** time.
- Driver **starting grid** position.
- **Weather** conditions (wet / not wet).


DATA
PREPARATION
& CLEANING

DATA PREPARATION AND CLEANING

merge_df3

	driverId	raceId	averagepit(s)	averagelaptime(ms)	fastestlaptime(ms)	grid	position
0	FIS	2008	31.375	116063.0	109101.0	20	0
1	VET	2008	28.133	115517.0	107271.0	7	1
2	BUT	2008	26.919	115675.0	108128.0	12	0
3	WEB	2008	31.518	120278.0	109183.0	13	0
4	SUT	2008	29.138	116013.0	109270.0	19	0
5	TRU	2008	28.346	116431.0	106972.0	11	0
6	COU	2008	34.811	115617.0	107562.0	14	0
7	MAS	2008	55.004	115925.0	105757.0	1	0
8	BAR	2008	28.532	112792.0	110320.0	18	0
9	GLO	2008	27.584	115482.0	107044.0	8	1
10	RAI	2008	37.271	115999.0	105599.0	3	0

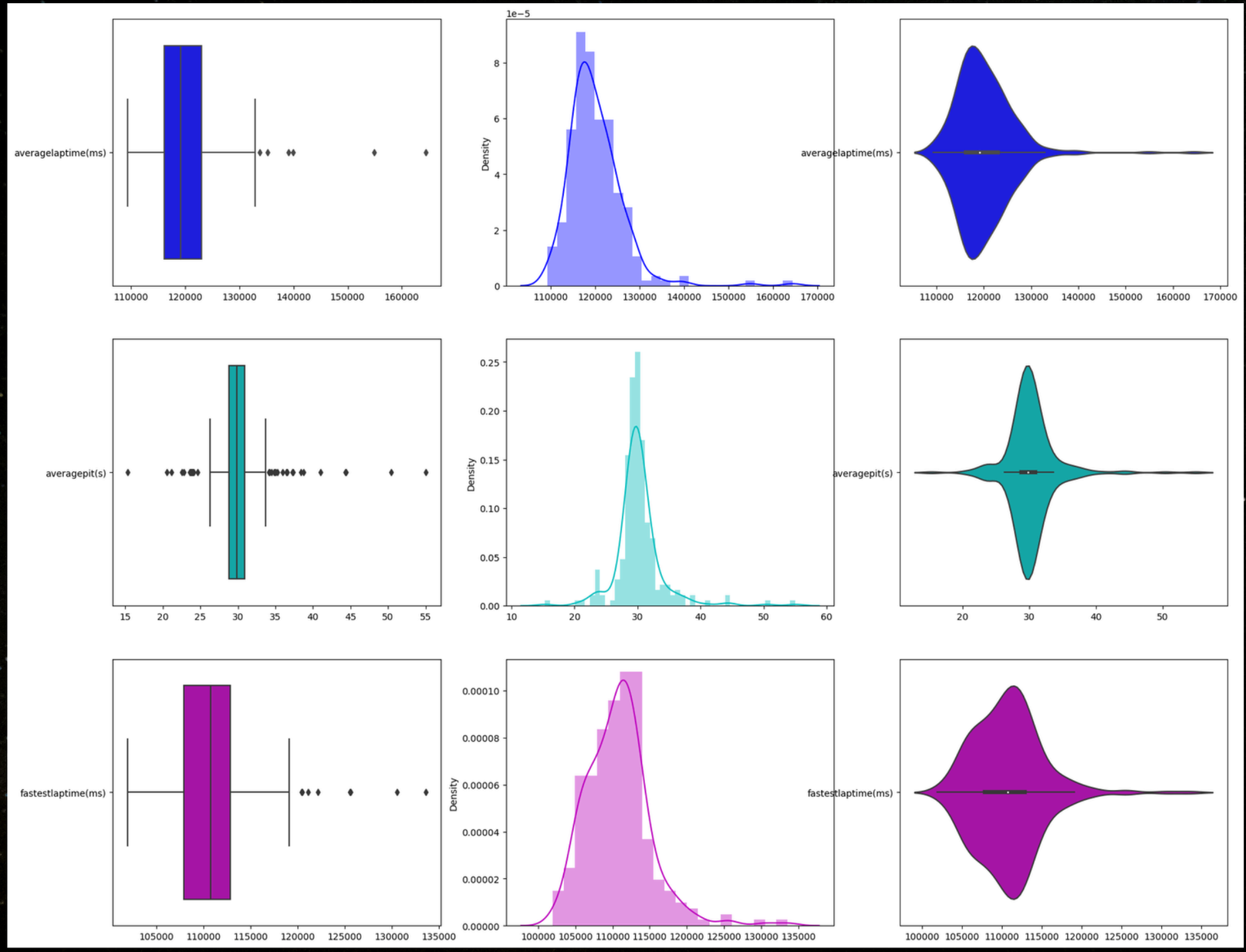
1	Calculate average lap times and pitstop times
2	Converted fastest lap times from a string to a float
3	Dropped unimportant/repeated data
4	Merged all the data frames together
5	Changed positions to 0/1, indicating top 5
6	Changed driverId and raceId to code names and race years

EXPLORATORY DATA ANALYSIS

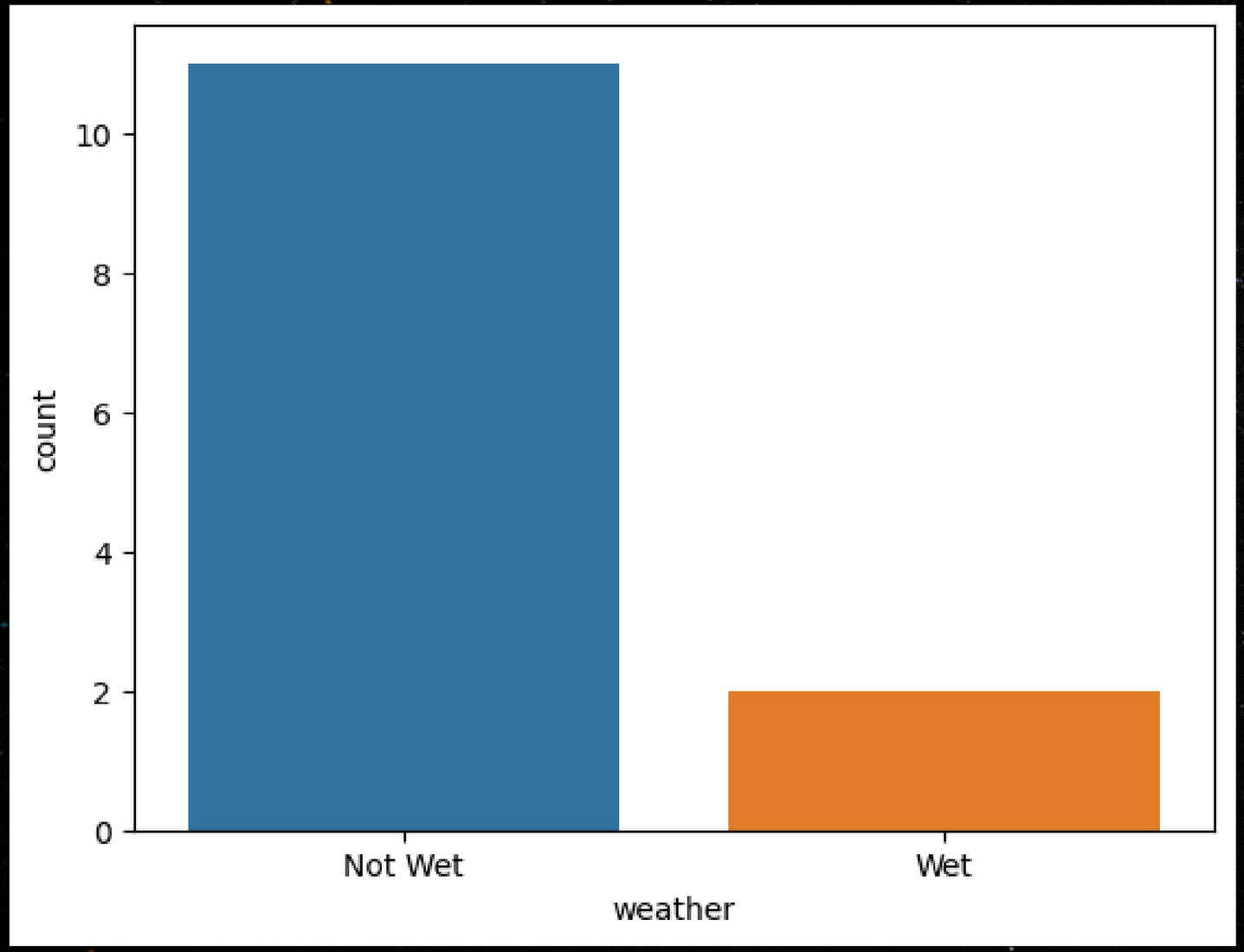
UNI-VARIATE VISUALIZATION

UNI-VARIATE VISUALIZATION

- Average **lap time** (ms)
- Average **pitstop** time (s)
- **Fastest lap** time (ms)



UNI-VARIATE VISUALIZATION

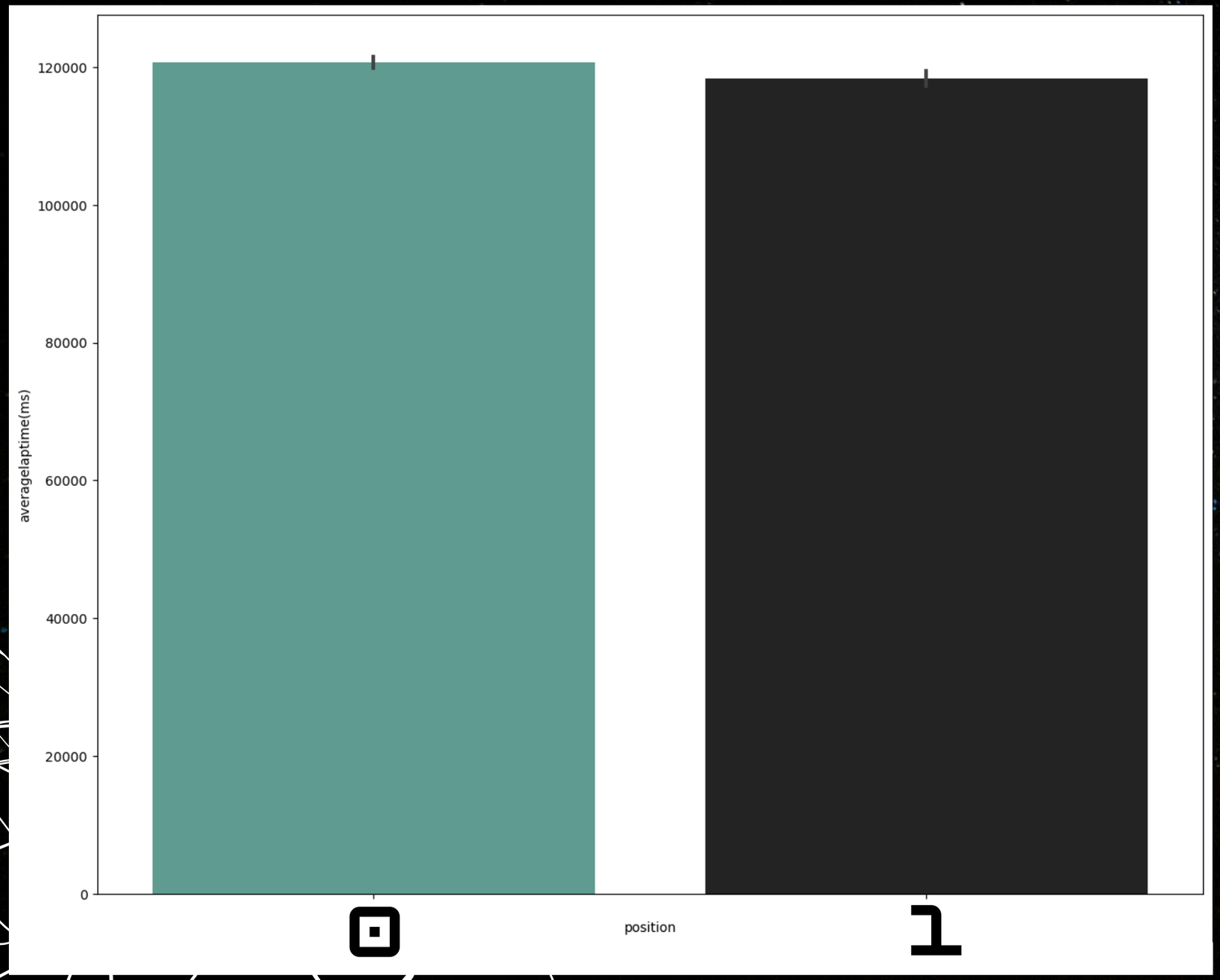


- Singapore GP **weather** conditions.

EXPLORATORY DATA ANALYSIS

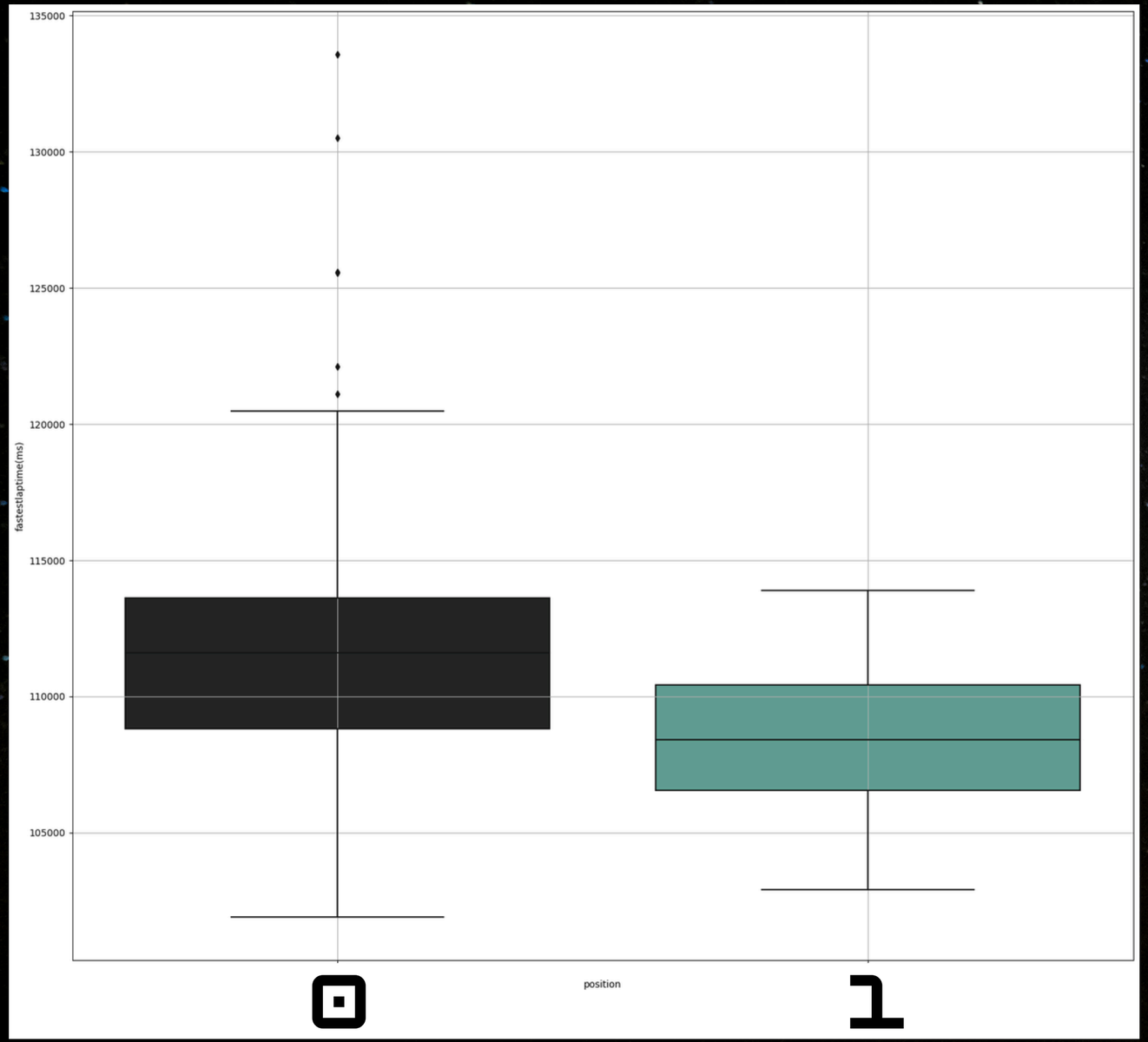
BI-VARIATE VISUALIZATION

BI-VARIATE VISUALIZATION



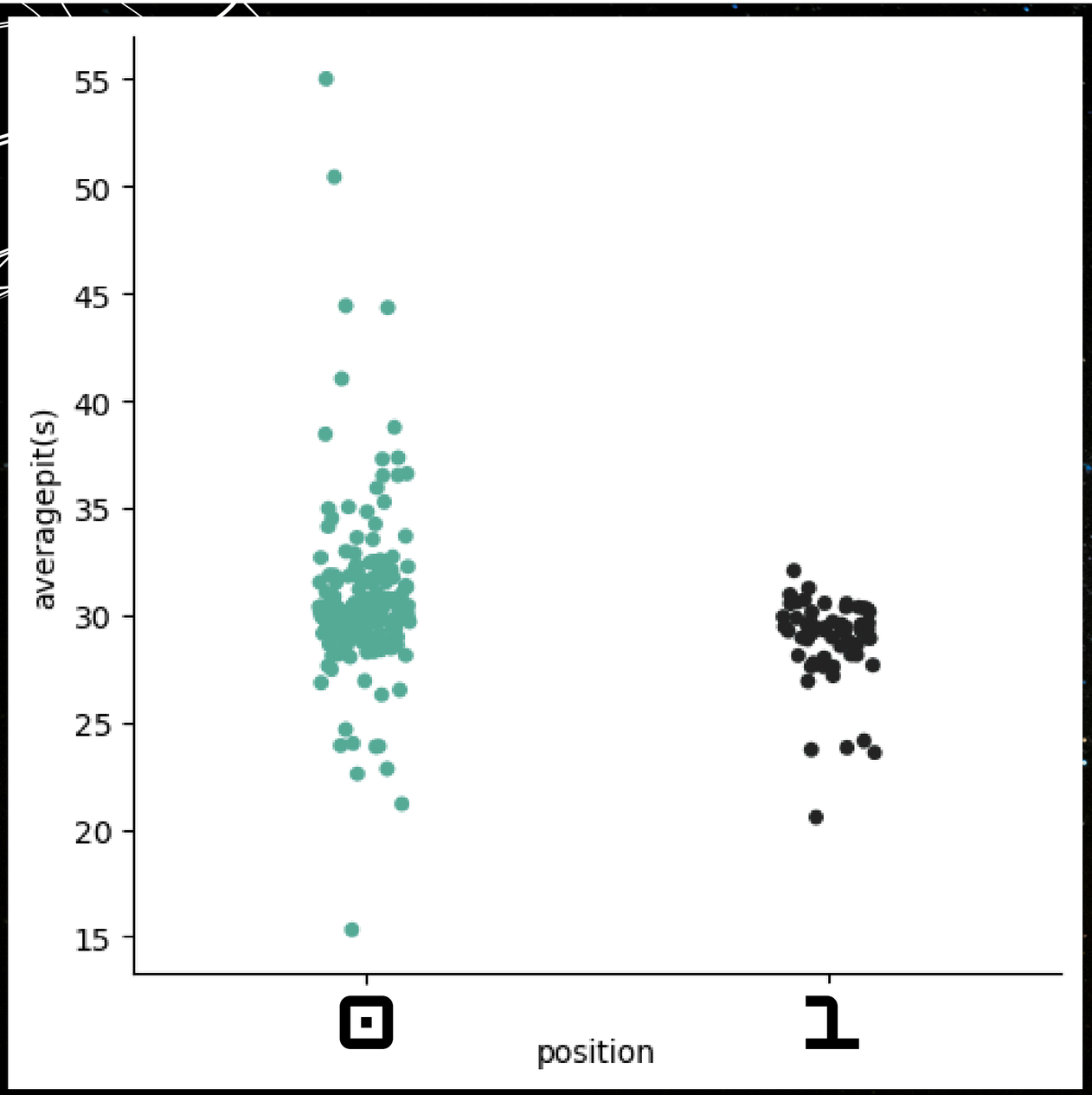
- Average **lap** times (ms) vs **Position**

BI-VARIATE VISUALIZATION



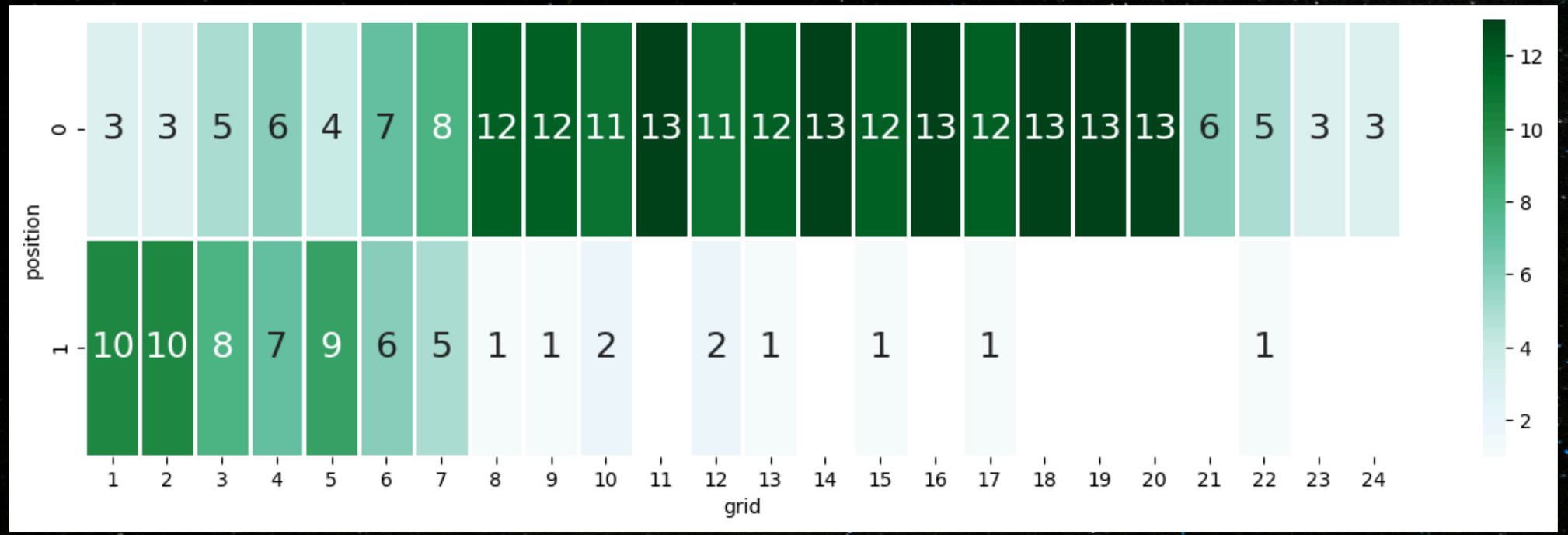
- **Fastest lap times (ms) vs Position**

BI-VARIATE VISUALIZATION



- Average **pitstop** (s) vs **Position**

BI-VARIATE VISUALIZATION

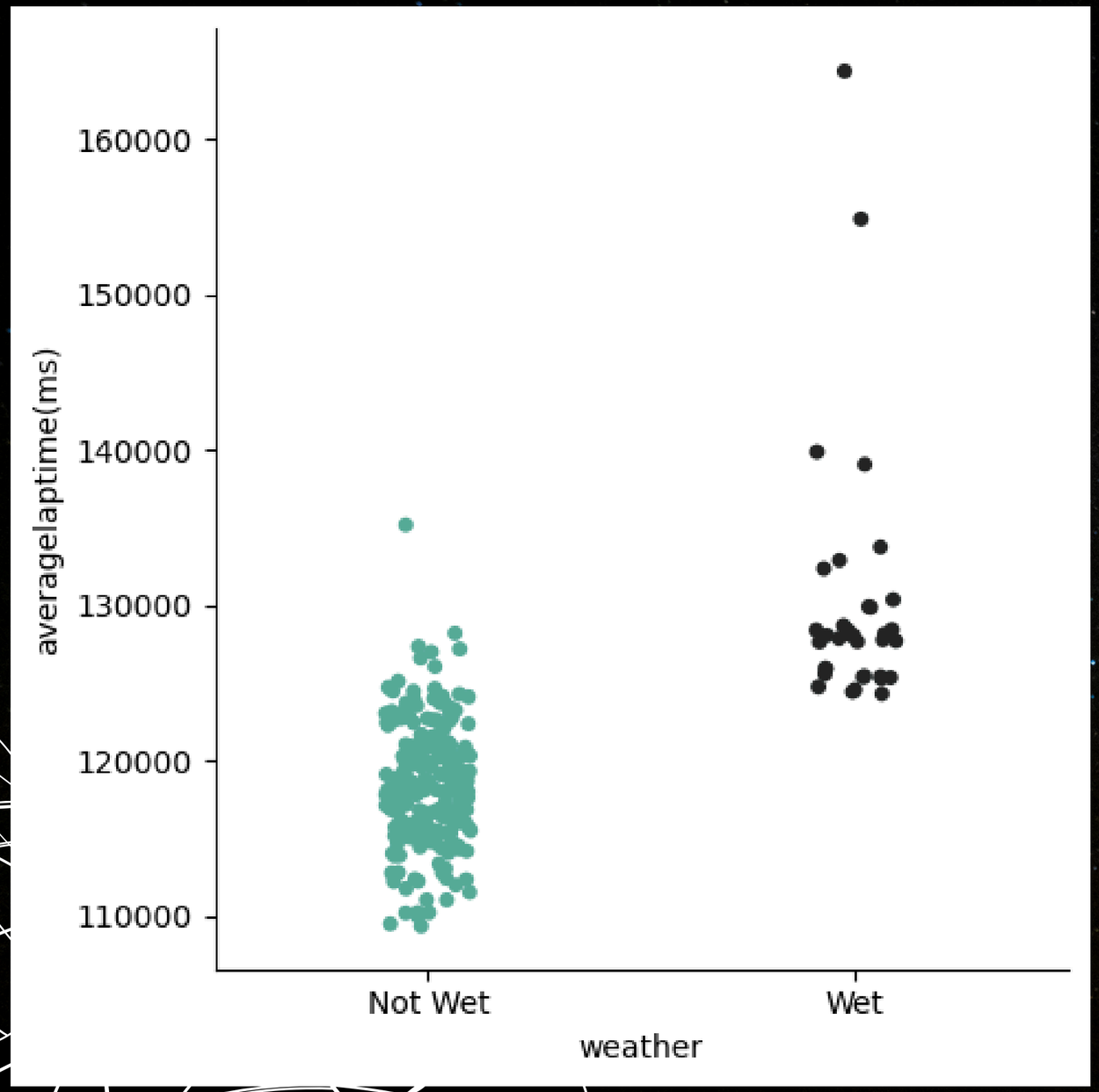


- **Starting grid** positions vs **Position**

EXPLORATORY DATA ANALYSIS

MULTI-VARIATE VISUALIZATION

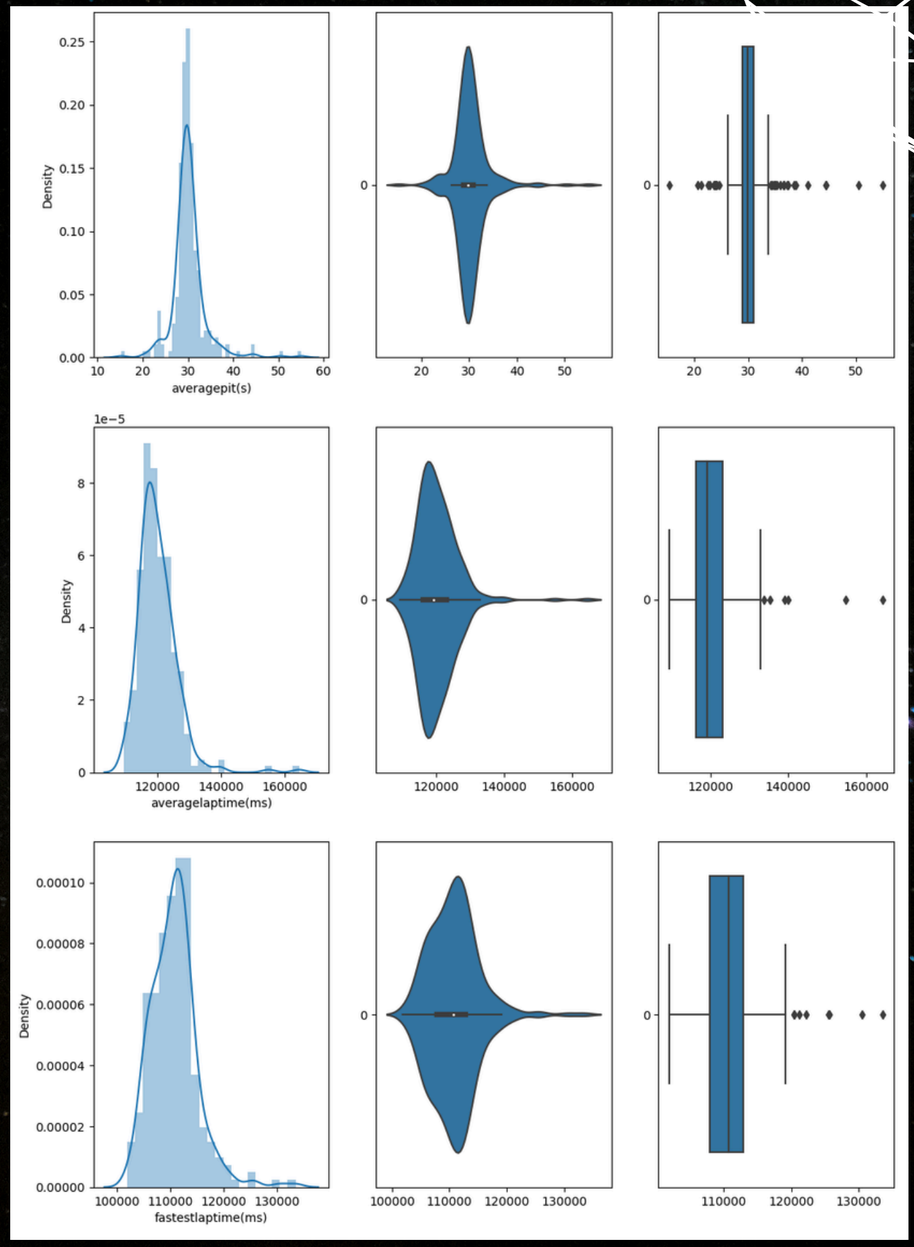
MULTI-VARIATE VISUALIZATION



- Average **lap time** (ms) vs **weather**

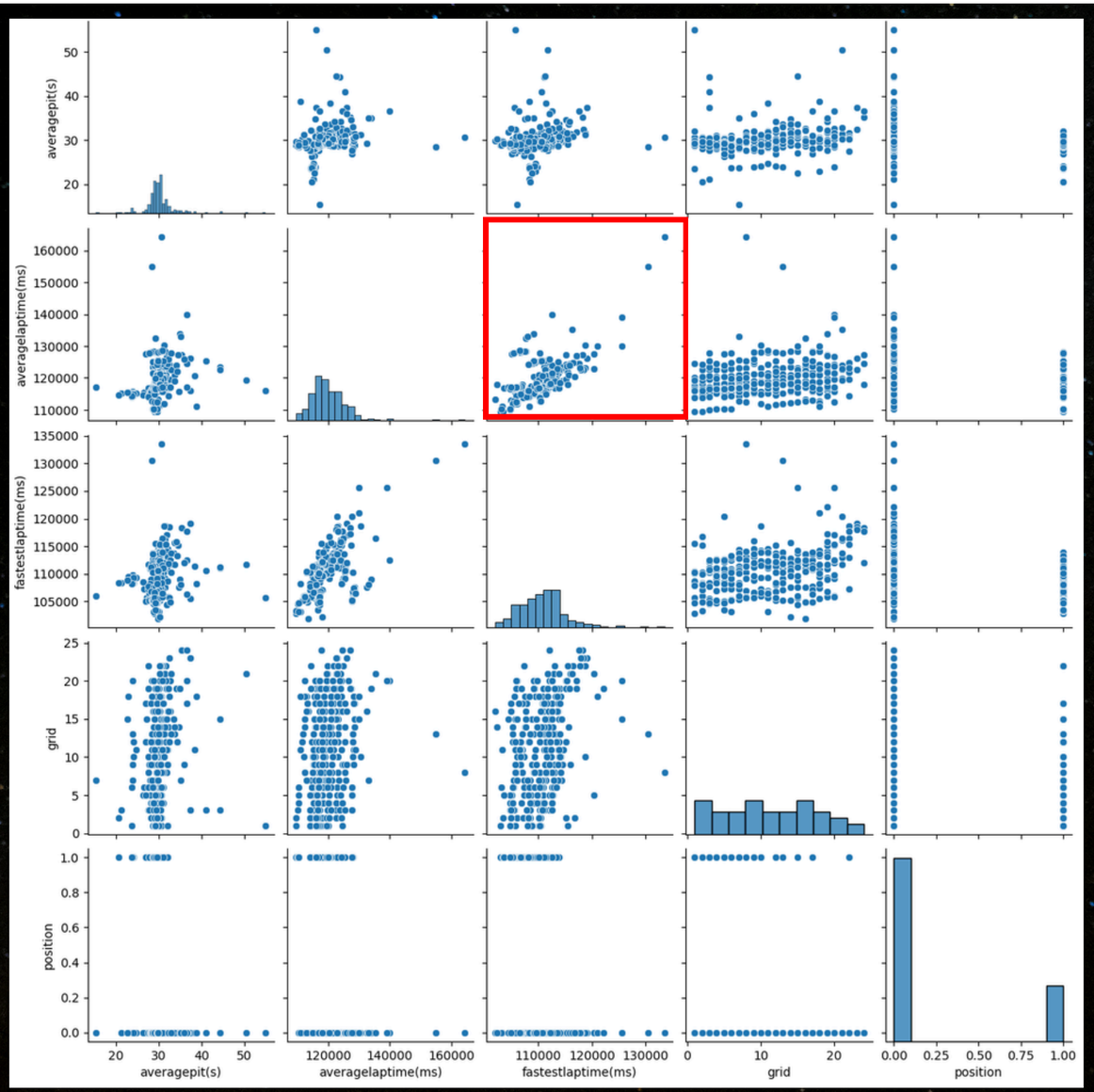
MULTI-VARIATE VISUALIZATION

- Average **pitstop** time (s)
- Average **lap time** (ms)
- **Fastest lap** time (ms)



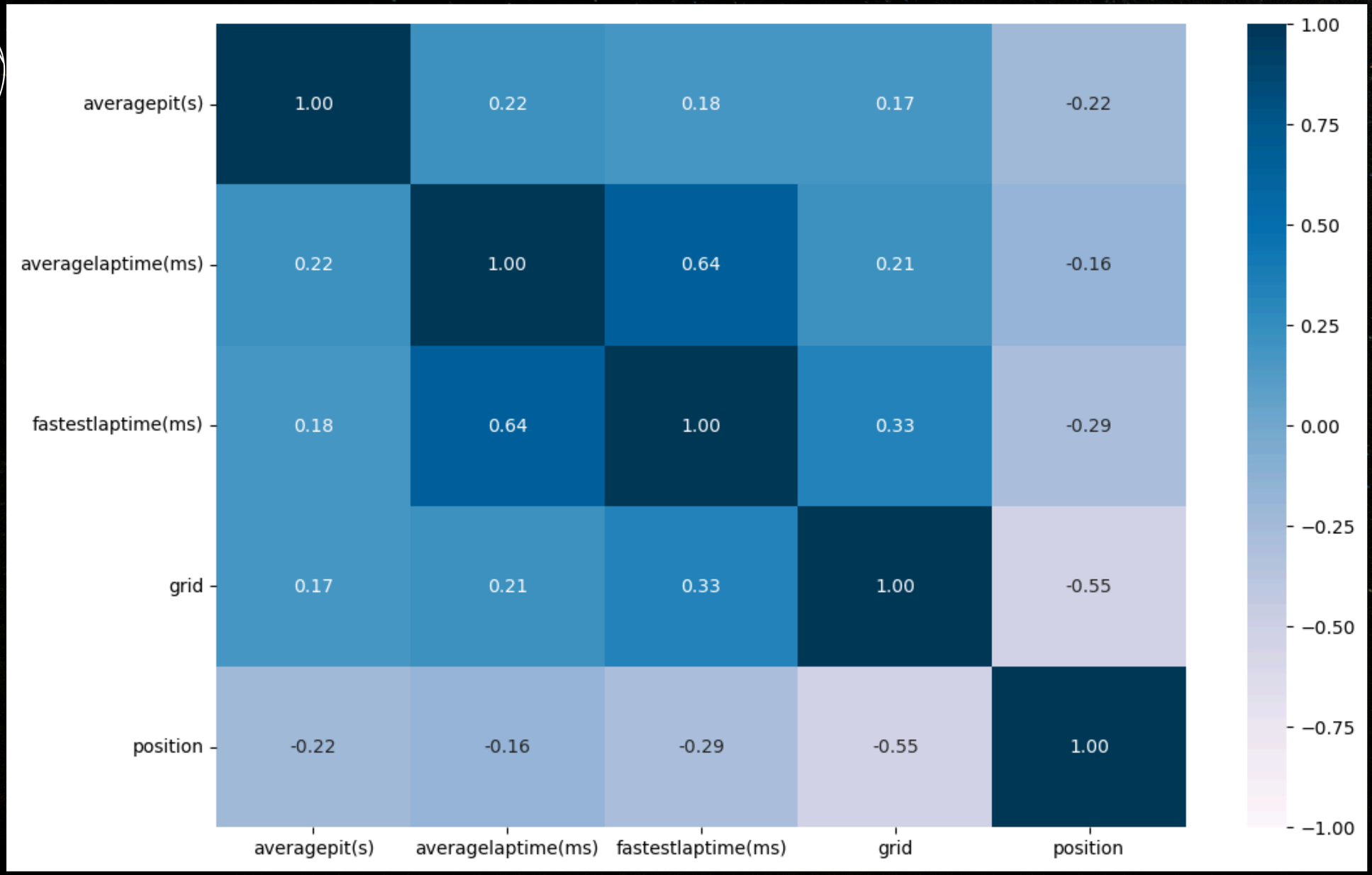
MULTI-VARIATE VISUALIZATION

- Average **pitstop** time (s)
- Average **lap time** (ms)
- **Fastest lap** time (ms)
- **Starting grid** positions
- **Position**

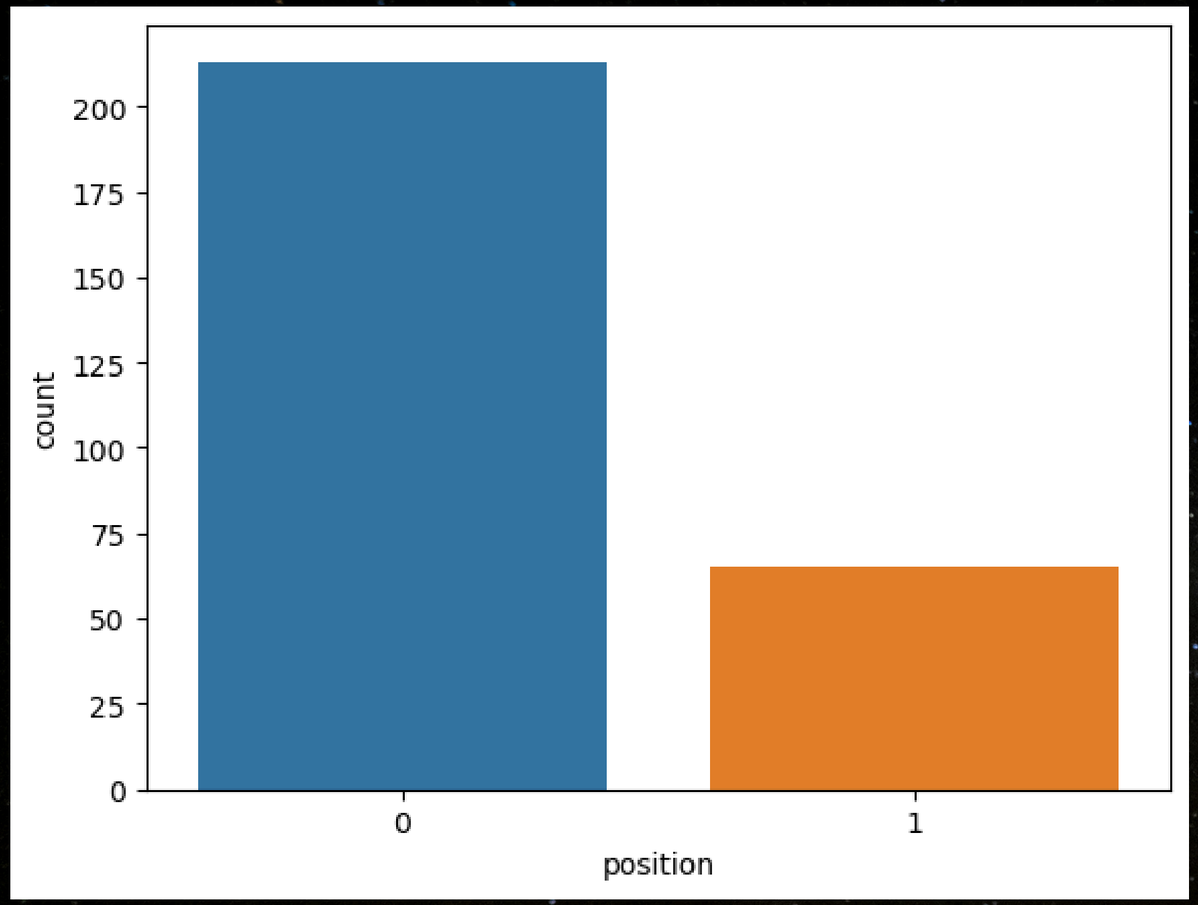


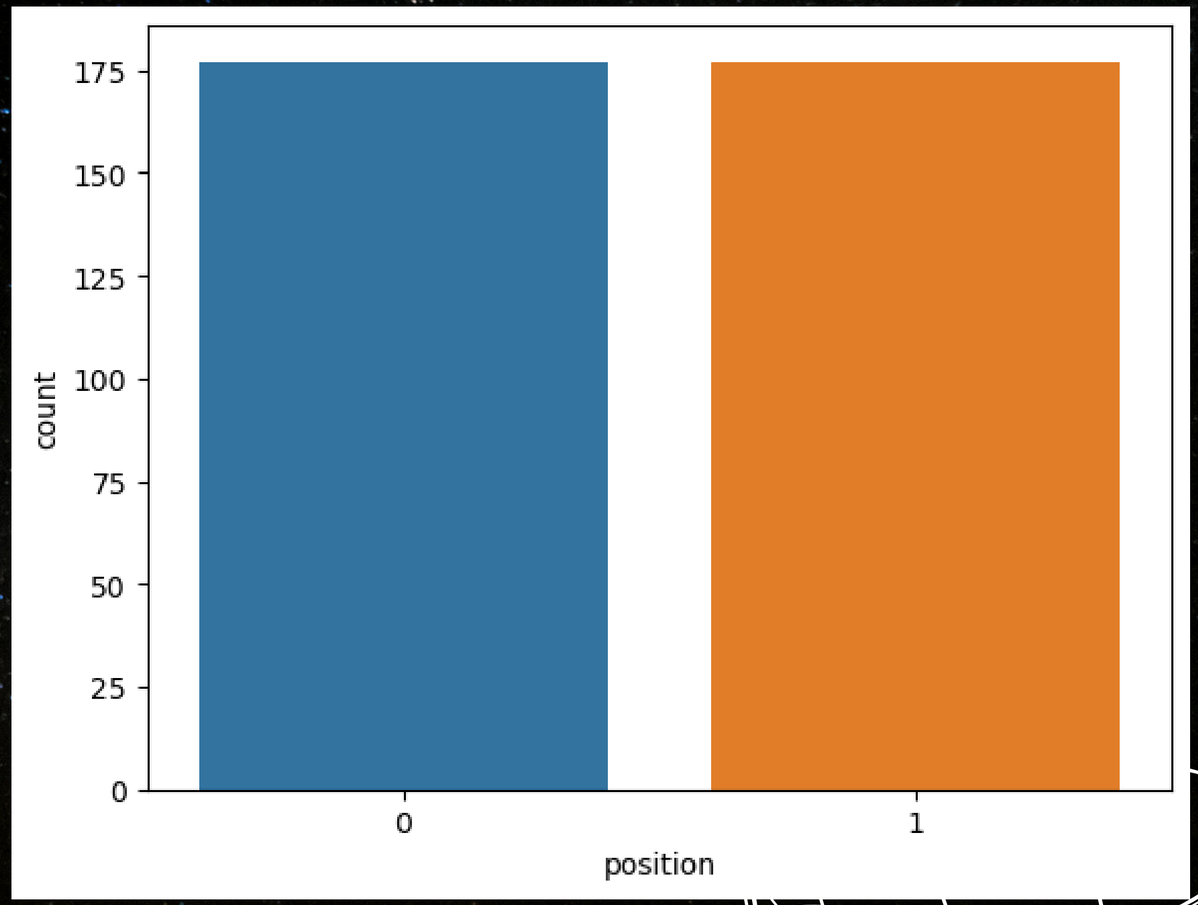
- Average **pitstop** time (s)
- Average **lap time** (ms)
- **Fastest lap** time (ms)
- **Starting grid** positions
- **Position**

MULTI-VARIATE VISUALIZATION



UPSAMPLING

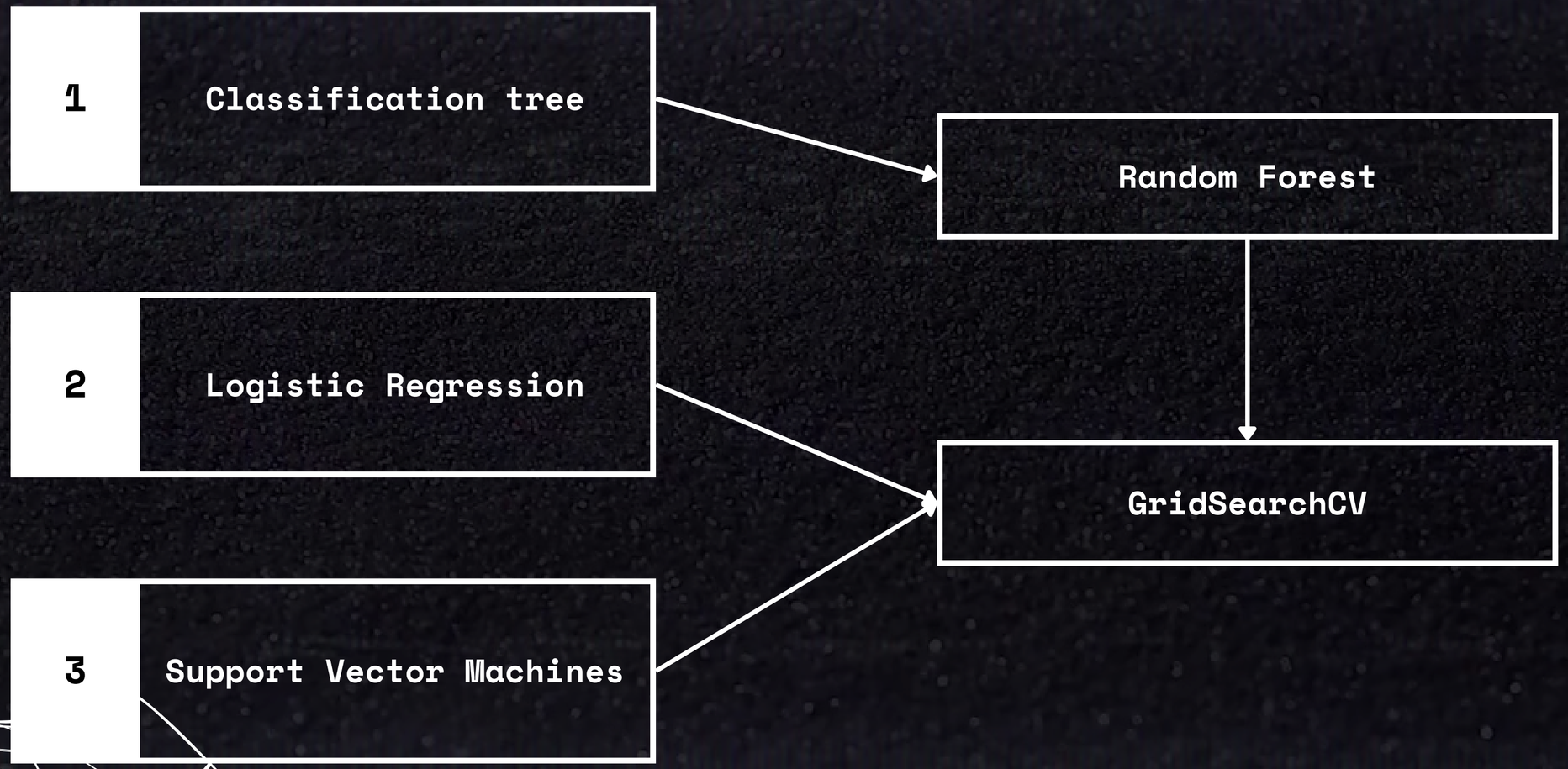




- **Before** upsampling
- **After** upsampling

MACHINE LEARNING (ML) MODULES

MACHINE LEARNING (ML) MODULES

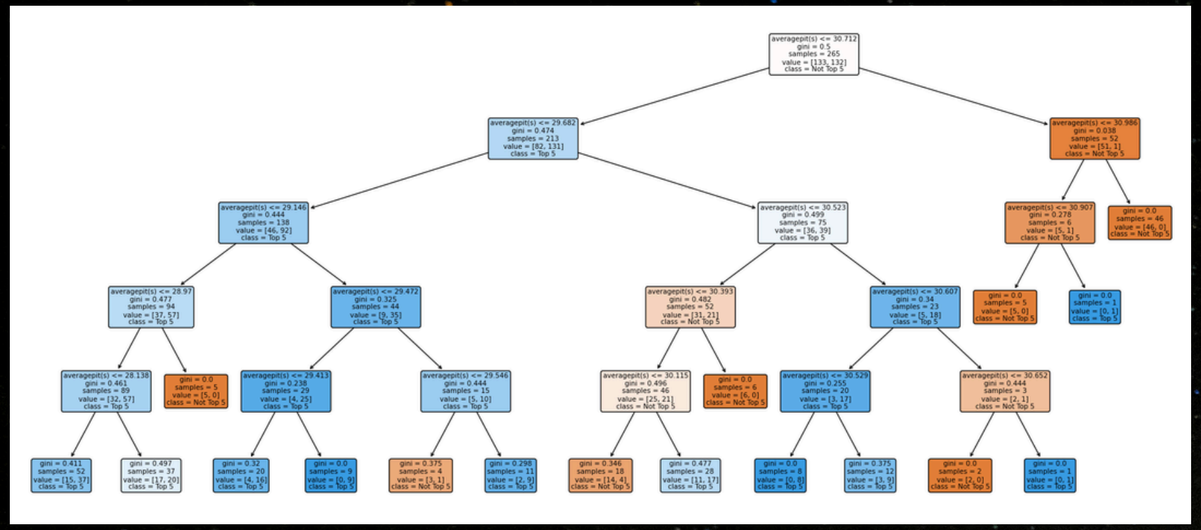


MACHINE LEARNING

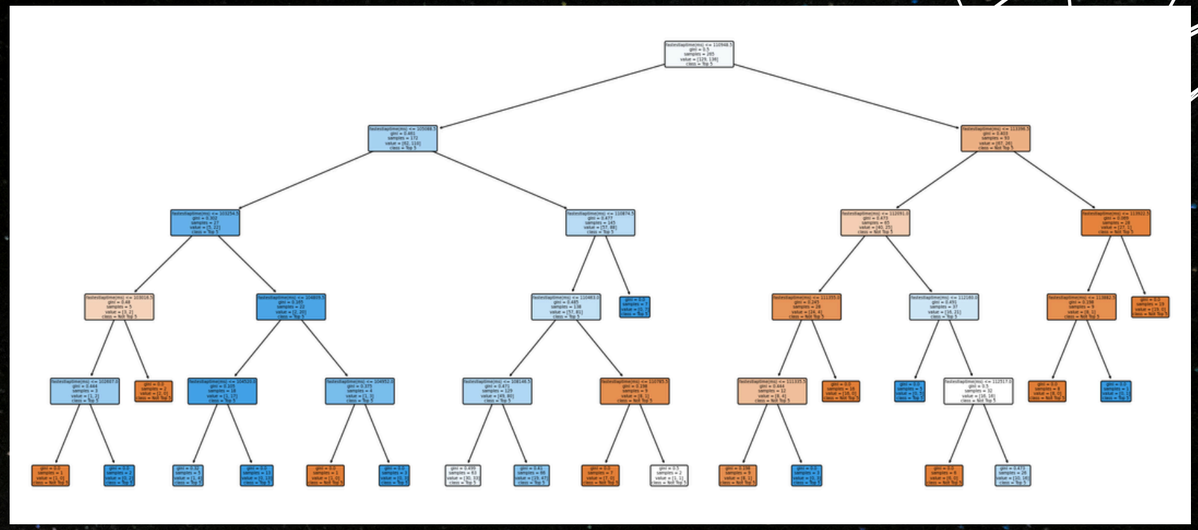
CLASSIFICATION TREE

UNI-VARIATE CLASSIFICATION

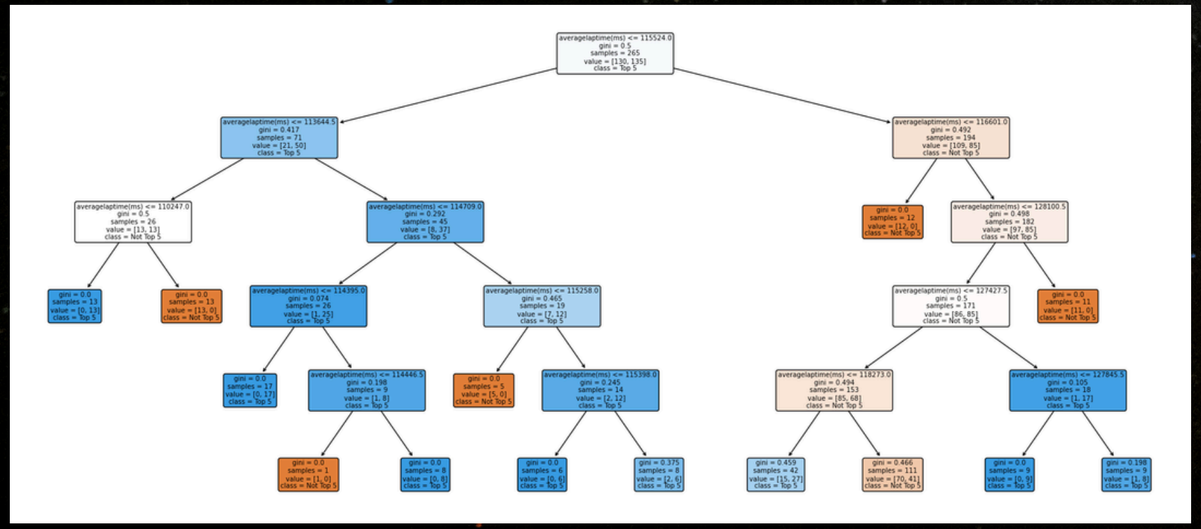
- Average **pitstop** time (s)



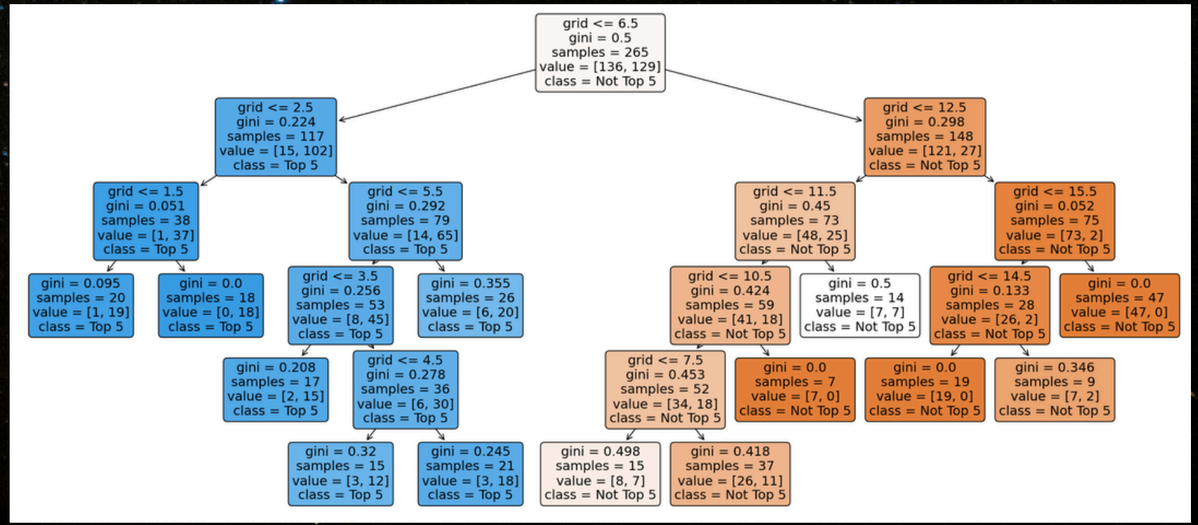
- **Fastest lap** time (ms)



- Average **lap time** (ms)



- **Starting grid** positions

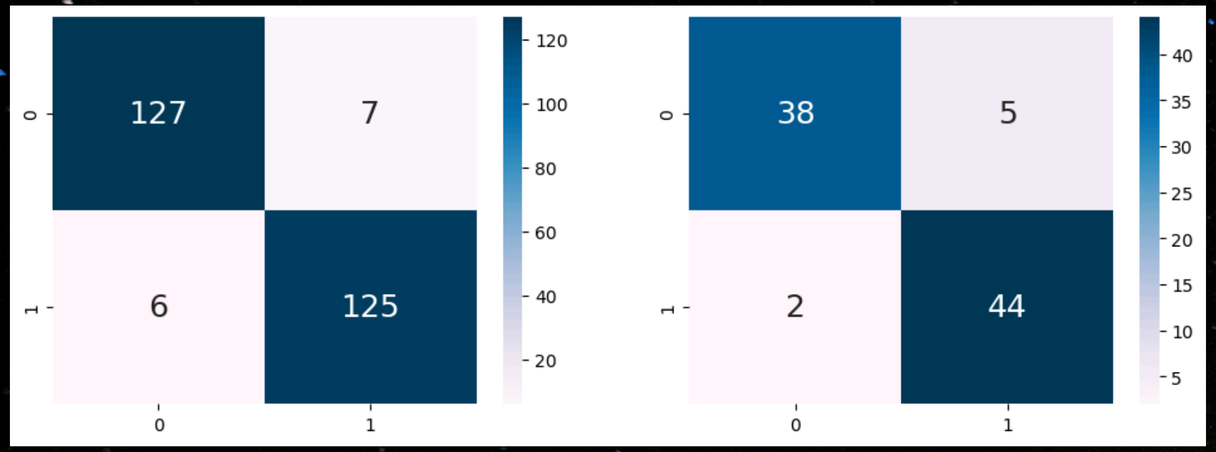


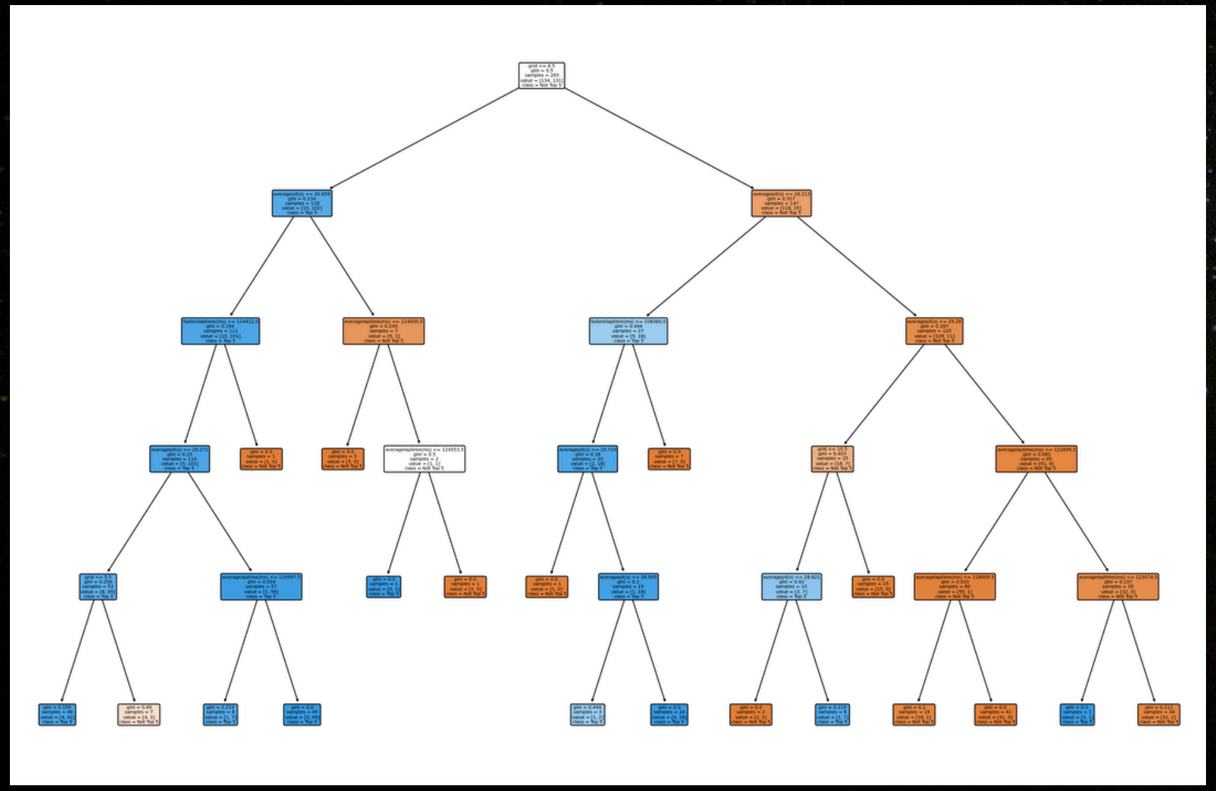
UNI-VARIATE CLASSIFICATION

	CLASSIFICATION ACCURACY/ TRAIN	CLASSIFICATION ACCURACY/ TEST
AVERAGE LAP TIME (MS)	0.7773584905660378	0.7191011235955056
AVERAGE PITSTOP TIME (S)	0.7849056603773585	0.7078651685393258
FASTEST LAP TIME (MS)	0.7660377358490567	0.7078651685393258
STARTING GRID POSITION	0.8415094339622642	0.8202247191011236

Tree depth 5

MULTI-VARIATE CLASSIFICATION





Train Dataset
Classification Accuracy : 0.9509433962264151

Test Dataset
Classification Accuracy : 0.9213483146067416

Train Dataset
True Positive Rate : 0.9541984732824428

Test Dataset
True Positive Rate : 0.9565217391304348

Train Dataset
True Negative Rate : 0.9477611940298507

Test Dataset
True Negative Rate : 0.8837209302325582

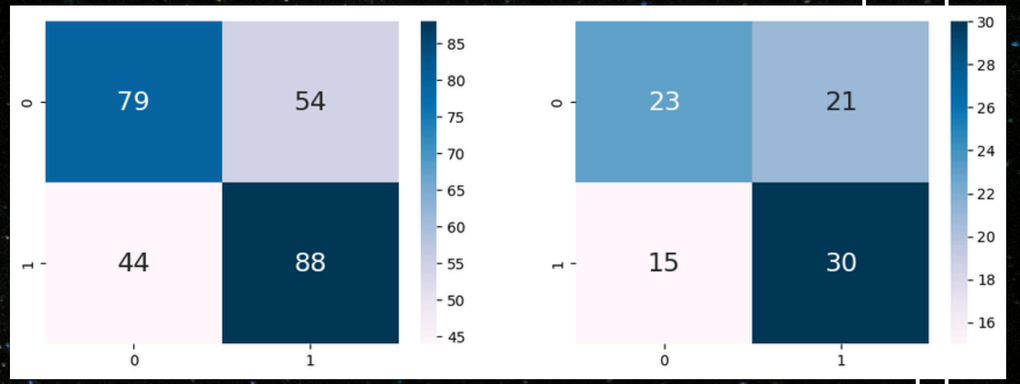
MACHINE LEARNING

SUPPORT VECTOR MACHINE & LOGISTIC REGRESSION

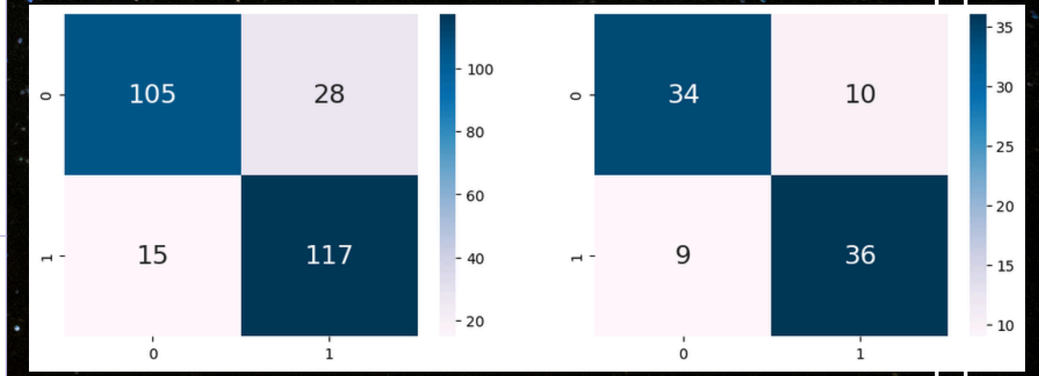
SVM & LOGISTIC REGRESSION

	CLASSIFICATION ACCURACY/ TRAIN	CLASSIFICATION ACCURACY/ TEST
SUPPORT VECTOR MACHINE (SVM)	0.630188679245283	0.5955056179775281
LOGISTIC REGRESSION	0.8377358490566038	0.7865168539325843
MULTI-VARIATE CLASSIFICATION TREE	0.9509433962264151	0.9213483146067416

- **Support Vector Machine**



- **Logistic Regression**

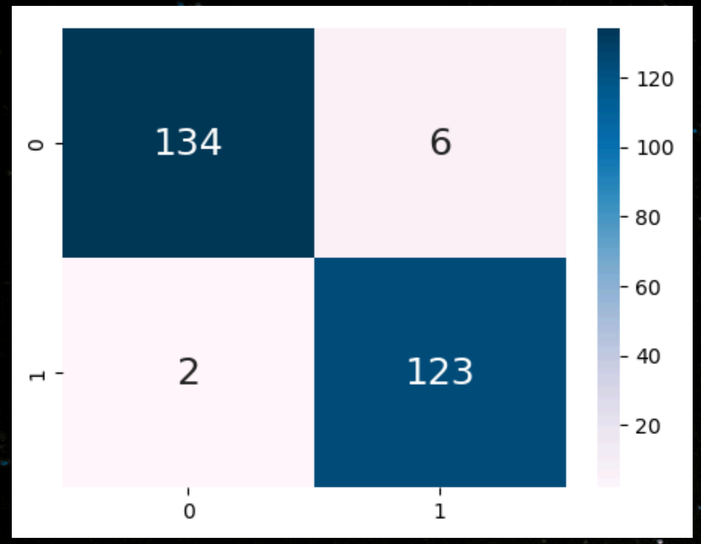


MACHINE LEARNING

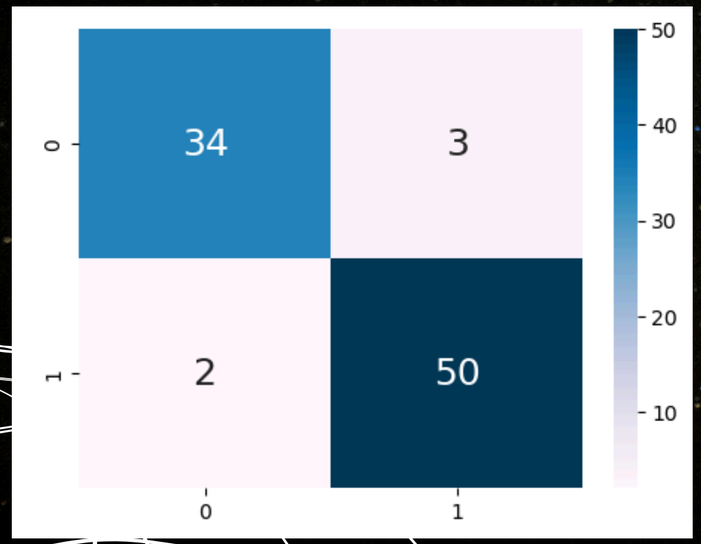
RANDOM FOREST

RANDOM FOREST

- **Train** Data: Depth **5**



- **Test** Data: Depth **5**



Train Dataset
Classification Accuracy : 0.969811320754717

Test Dataset
Classification Accuracy : 0.9438202247191011

Train Dataset
True Positive Rate : 0.984

Test Dataset
True Positive Rate : 0.9615384615384616

Train Dataset
True Negative Rate : 0.9571428571428572

Test Dataset
True Negative Rate : 0.9189189189189189

GRID SEARCH

1 Support Vector Machines

```
search_space = {
    'kernel' : ['rbf', 'poly', 'sigmoid', 'linear'],
    'C' : [0.1, 1, 100, 1000],
    'degree' : [1,3,5,7,9]
}
```

2 Logistic Regression

```
search_space2 = {
    'penalty' : ['l1', 'l2', 'elasticnet'],
    'C' : np.logspace(-4, 4, 20),
    'max_iter' : [100, 1000, 2500, 5000],
    'solver' : ['lbfgs', 'newton-cg', 'liblinear', 'sag', 'saga']
}
```

3 Random Forest

```
search_space3 = {
    'n_estimators' : [100,500,1000,1500],
    'max_depth' : [4,5,6],
    'max_samples' : [0.25,0.5,0.75,1.0],
    'max_features' : ['sqrt', 'log2', 0.2, 0.4, 0.6, 0.8, 1.0]
}
```

GRID SEARCH RESULTS

CLASSIFICATION ACCURACY

	BEFORE HYPERPARAMETER TUNING		AFTER HYPERPARAMETER TUNING	
	TRAIN	TEST	TRAIN	TEST
SUPPORT VECTOR MACHINE (SVM)	0.630188679245283	0.5955056179775281	0.7584905660377359	0.797752808988764
LOGISTIC REGRESSION	0.8377358490566038	0.7865168539325843	0.8566037735849057	0.8426966292134831
RANDOM FOREST	0.969811320754717	0.9438202247191011	0.9773584905660377	0.9438202247191011

PROJECT OUTCOME

CONCLUSION

1	Able to predict top 5 with relatively high accuracy in Singapore
2	F1 Teams can see which department they are lacking in and work to improve on them
3	Model has not been tested overseas, effectiveness is unconfirmed if model is used in other countries
4	3 of the prediction variables would be difficult to obtain when watching f1 race in real time

WHAT WE LEARNT

1	Logistic Regression, SVM and Random Forest machine learning models
2	Hypertuning our parameters using GridSearchCV

DATA DRIVEN INSIGHTS

Better Starting Position

Teams and drivers to commit fully during qualifiers

Lower average lap times

Hire better drivers and have more practice to improve their consistency

Lower fastest lap times

More investment into R&D

Lower average pit stop times

Increase amount of practice and employ better strategies


THANK
YOU