RIOT GAMES

VALORANT

Streamers Influence

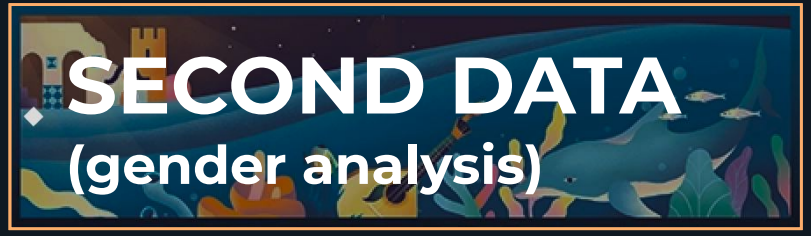
Data Research

DEFY
THE
LIMITS


VALORANT™
A 5v5 Character-Based Tactical Shooter

AGENDA

- **FIRST DATA**
  (viewers analysis)



- **SECOND DATA**
  (gender analysis)

- **SECOND DATA**
  (age analysis)

AGENDA

- **FIRST DATA**
  (viewers analysis)

SECOND DATA
(gender analysis)

SECOND DATA
(age analysis)

20

FIRST DATA (Viewers & Players Count)

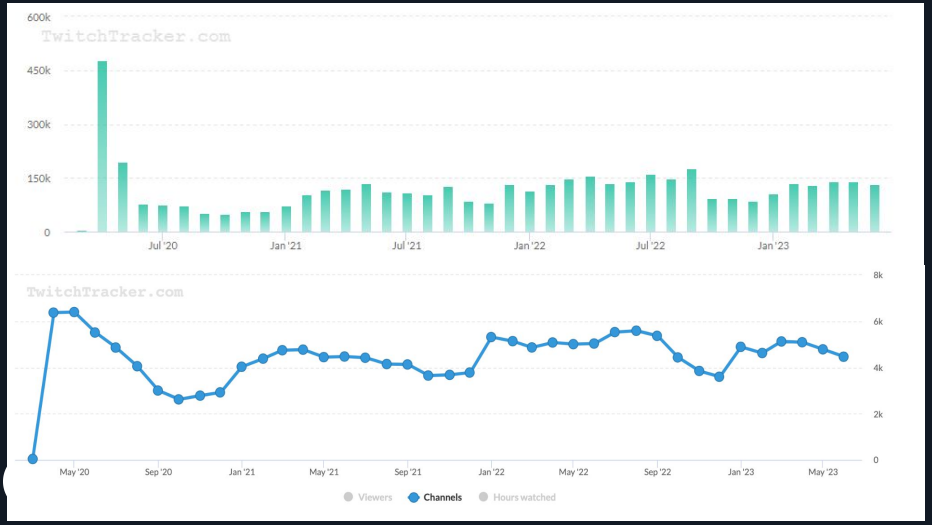
21

First (Our Findings)

22

First (Model Predictions)

Average Valorant Viewership and Channels on Twitch Per Month:



Valorant Player Count per Month:

Date	Players Count
June 2023	18,665,639
May 2023	19,834,991
April 2023	20,100,075
March 2023	20,124,255
February 2023	18,296,396
January 2023	18,886,523

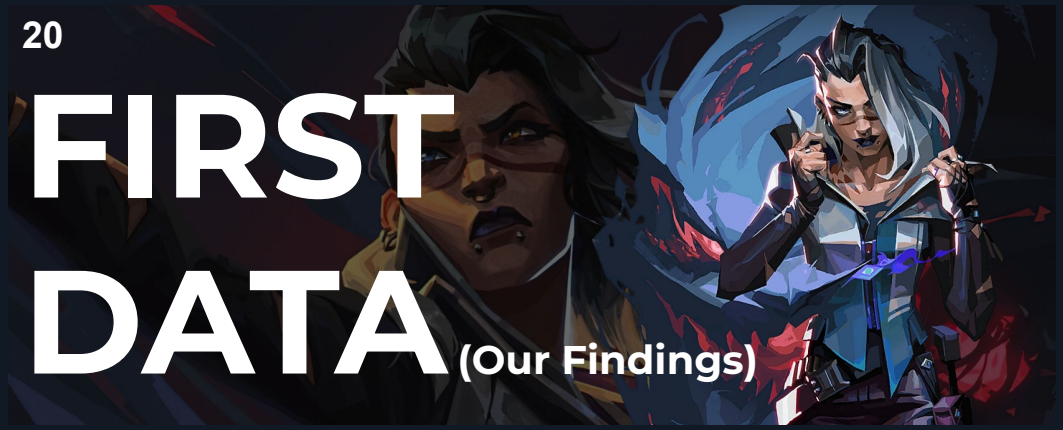
21

First

(Viewers and Player Count Graph)

20

FIRST DATA (Our Findings)



22

First

(Model Predictions)

```
In [15]: import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt

         data = pd.read_csv('players_viewers.csv')
         x = data['AverageView'].values.reshape(-1,1)
         y = data['Players'].values.reshape(-1,1)

         from sklearn.model_selection import train_test_split
         x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.25, random_state = 0)

         from sklearn.linear_model import LinearRegression
         ml = LinearRegression()
         ml.fit(x_train,y_train)

         # Predicting test set result
         y_predict = ml.predict(x_test)
         print(y_predict)
         ml.predict([[129000]])

         from sklearn.metrics import r2_score
         r2_score(y_test, y_predict)

         # Visualization of the training set results
         plt.scatter(y_test, y_predict)
         plt.xlabel('Actual Players')
         plt.ylabel('Predicted Players')
         plt.title('Actual vs. Predicted')
```

First Data Finding:

- Linear Regression
- Scikit-learn train test split data
- Test Size: 0.25, Random State: 0

21

First

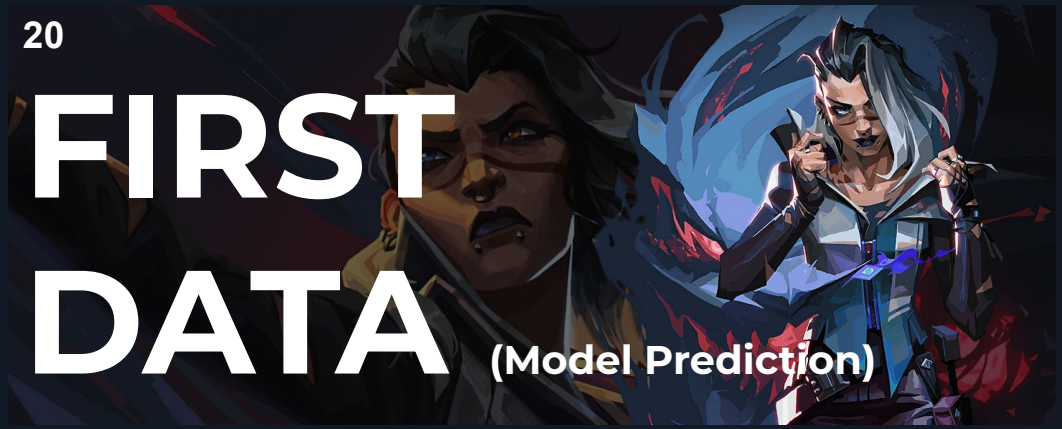
(Viewers and Player Count Graph)

21

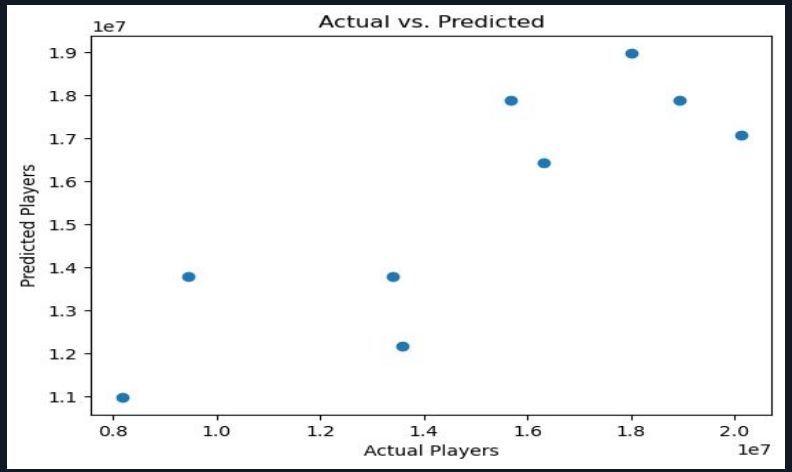
First

(Our Findings)

20



FIRST DATA (Model Prediction)

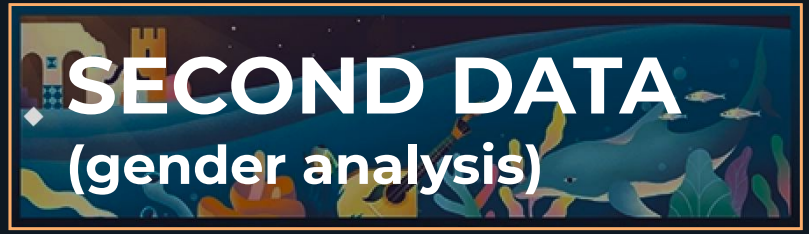


Predictions:

- If there are 129,000 average viewers, the average players would be around 17m
- If there are 141,000 average viewers, the average players would be 18m

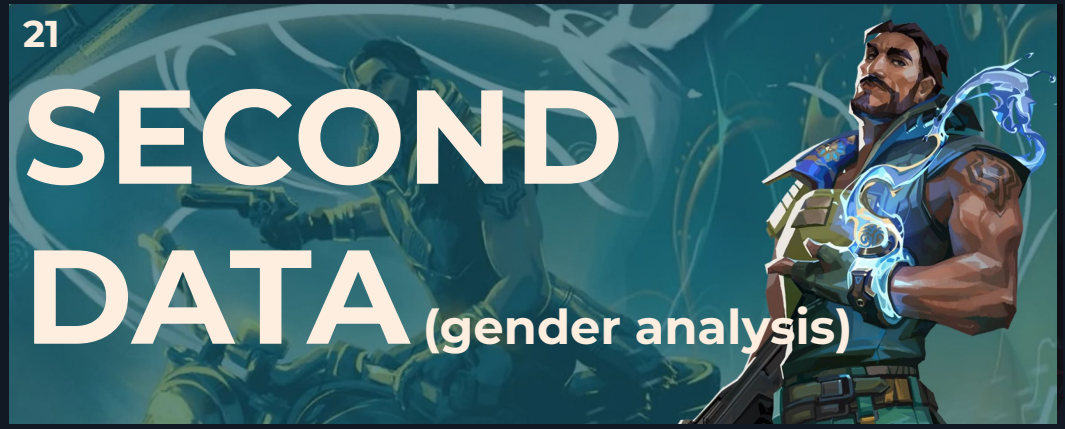
AGENDA

FIRST DATA
(viewers analysis)



SECOND DATA
(gender analysis)

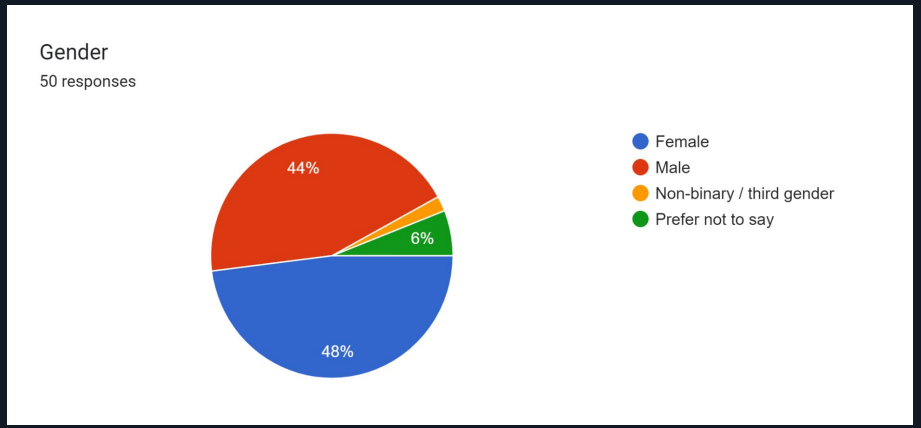
SECOND DATA
(age analysis)



21
Second
(Our Findings)

22
Second
(Model Predictions)

Data from Survey



	Gender	Age	watch_stream	was_influenced	game_influenced	play_regardless
2	Female	20-25	Yes	Yes	Valorant	No
3	Female	14-19	No	No	None	Yes
4	Female	20-25	Yes	Yes	Other	Yes
5	Female	20-25	Yes	Yes	Valorant	Yes
6	Female	20-25	Yes	Yes	Valorant	No
7	Male	20-25	Yes	Yes	Other	No
8	Male	14-19	Yes	Yes	Valorant	No
9	Male	14-19	Yes	Yes	Other	Yes
10	Female	20-25	Yes	No	Valorant	Yes
11	Male	20-25	Yes	Yes	Valorant	No
12	Male	20-25	Yes	Yes	Other	Yes
13	Female	20-25	Yes	Yes	Valorant	Yes
14	Female	14-19	Yes	Yes	Other	Yes
15	Male	26+	Yes	Yes	Valorant	No
16	Male	20-25	Yes	No	None	Yes
17	Female	20-25	Yes	No	Other	Yes
18	Male	26+	Yes	No	None	Yes
19	Female	20-25	Yes	No	None	Yes
20	Female	20-25	Yes	Yes	Valorant	Yes

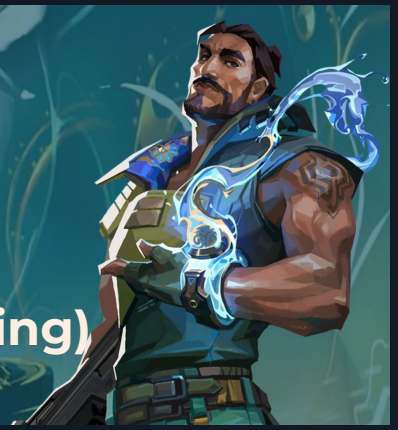
20

Second

(Gender Analysis)

21

SECOND DATA (Our Finding)



22

Second

(Model Prediction)

```
In [10]: import pandas as pd
df = pd.read_csv("streamers_influence_response.csv")
inputs = df.drop(['play_regardless', 'Age', 'game_influenced'], axis = 'columns')
target = df['game_influenced']

from sklearn.preprocessing import LabelEncoder
le_gender = LabelEncoder()
le_watch_stream = LabelEncoder()
le_was_influenced = LabelEncoder()
le_game_influenced = LabelEncoder()

inputs['Gender_n'] = le_gender.fit_transform(inputs['Gender'])
inputs['watch_stream_n'] = le_gender.fit_transform(inputs['watch_stream'])
inputs['was_influenced_n'] = le_gender.fit_transform(inputs['was_influenced'])
inputs.head()

inputs_n = inputs.drop(['Gender', 'watch_stream', 'was_influenced'], axis='columns')
inputs_n


from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(inputs_n, target, test_size = 0.25, random_state =

from sklearn import tree
model = tree.DecisionTreeClassifier()
model.fit(inputs_n, target)
model.score(inputs_n, target)
y_pred = model.predict(x_test)

from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred)

Out[10]: 0.8461538461538461
```

Second Data Finding:

- Use Decision Tree
- Scikit-learn train test split data
- Test Size: 0.25, Random State: 0

Predictions:

For gender, there is an 84% chance that a player may be influenced by Content Creators

AGENDA

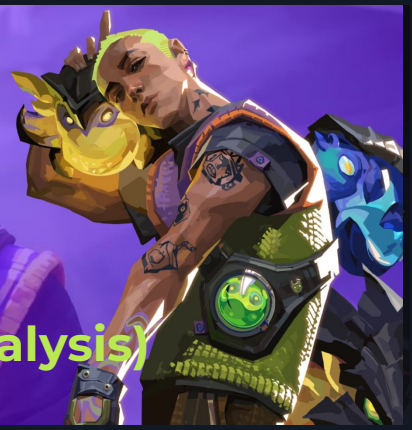
FIRST DATA
(viewers analysis)

SECOND DATA
(gender analysis)

SECOND DATA
(age analysis)

22

SECOND DATA (Age Analysis)



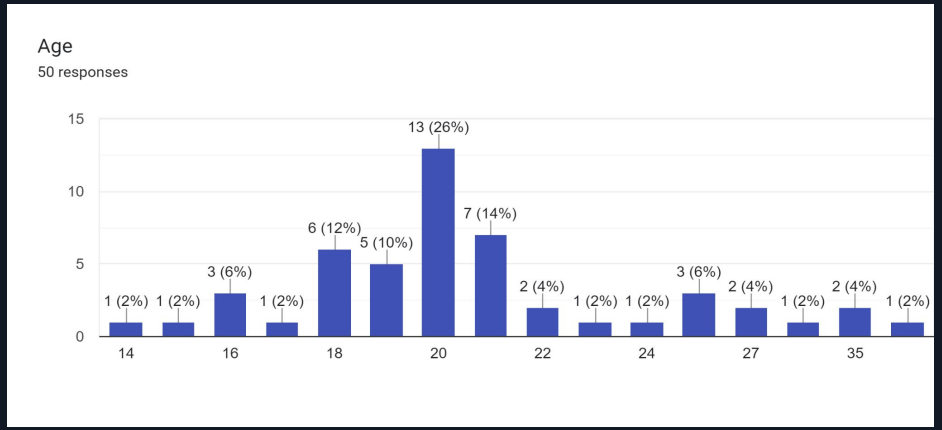
21

Second (Our Findings)

22

Second (Model Predictions)

Data from Survey



	Gender	Age	watch_stream	was_influenced	game_influenced	play_regardless
2	Female	20-25	Yes	Yes	Valorant	No
3	Female	14-19	No	No	None	Yes
4	Female	20-25	Yes	Yes	Other	Yes
5	Female	20-25	Yes	Yes	Valorant	Yes
6	Female	20-25	Yes	Yes	Valorant	No
7	Male	20-25	Yes	Yes	Other	No
8	Male	14-19	Yes	Yes	Valorant	No
9	Male	14-19	Yes	Yes	Other	Yes
10	Female	20-25	Yes	No	Valorant	Yes
11	Male	20-25	Yes	Yes	Valorant	No
12	Male	20-25	Yes	Yes	Other	Yes
13	Female	20-25	Yes	Yes	Valorant	Yes
14	Female	14-19	Yes	Yes	Other	Yes
15	Male	26+	Yes	Yes	Valorant	No
16	Male	20-25	Yes	No	None	Yes
17	Female	20-25	Yes	No	Other	Yes
18	Male	26+	Yes	No	None	Yes
19	Female	20-25	Yes	No	None	Yes
20	Female	20-25	Yes	Yes	Valorant	Yes

20

Second

(Gender Analysis)

22

22

Second

(Model Prediction)

```
In [17]: import pandas as pd
         df = pd.read_csv("streamers_influence_age.csv")
         inputs = df.drop(['play_regardless', 'Gender', 'game_influenced'], axis = 'columns')
         target = df['game_influenced']

         from sklearn.preprocessing import LabelEncoder
         le_age = LabelEncoder()
         le_watch_stream = LabelEncoder()
         le_was_influenced = LabelEncoder()
         le_game_influenced = LabelEncoder()

         inputs['Age_n'] = le_age.fit_transform(inputs['Age'])
         inputs['watch_stream_n'] = le_age.fit_transform(inputs['watch_stream'])
         inputs['was_influenced_n'] = le_age.fit_transform(inputs['was_influenced'])
         inputs.head()

         inputs_n = inputs.drop(['Age', 'watch_stream', 'was_influenced'], axis='columns')
         inputs_n

         from sklearn.model_selection import train_test_split
         x_train, x_test, y_train, y_test = train_test_split(inputs_n, target, test_size = 0.25, random_state =

         from sklearn import tree
         model = tree.DecisionTreeClassifier()
         model.fit(inputs_n, target)
         model.score(inputs_n, target)
         y_pred = model.predict(x_test)

         from sklearn.metrics import accuracy_score
         accuracy_score(y_test, y_pred)

Out[17]: 0.6923076923076923
```

Second Data Finding:

- Use Decision Tree

Predictions:

For age, there is an 69% chance that a player may be influenced by Content Creators

CONCLUSIONS

- There is a correlation between viewers and players based
- Age: older age were less likely to be influenced.
- Gender: the different groups had about the same influence.

REFERENCES

- Twitch Valorant Viewership Counts:
  - https://twitchtracker.com/games/516575
- Valorant Players Counts:
  - https://tracker.gg/valorant/population
- Research Survey
  - https://docs.google.com/forms/d/1g71FmDLDZDxJ0Je7Wve4eSkN0OY2hD9Ou4yXzTP5Oys/edit#responses
- Slide Template
  - https://www.dropbox.com/s/1akk9p8bfu9ifva/VALORANT%20PowerPoint_adjusted.pptx?dl=0