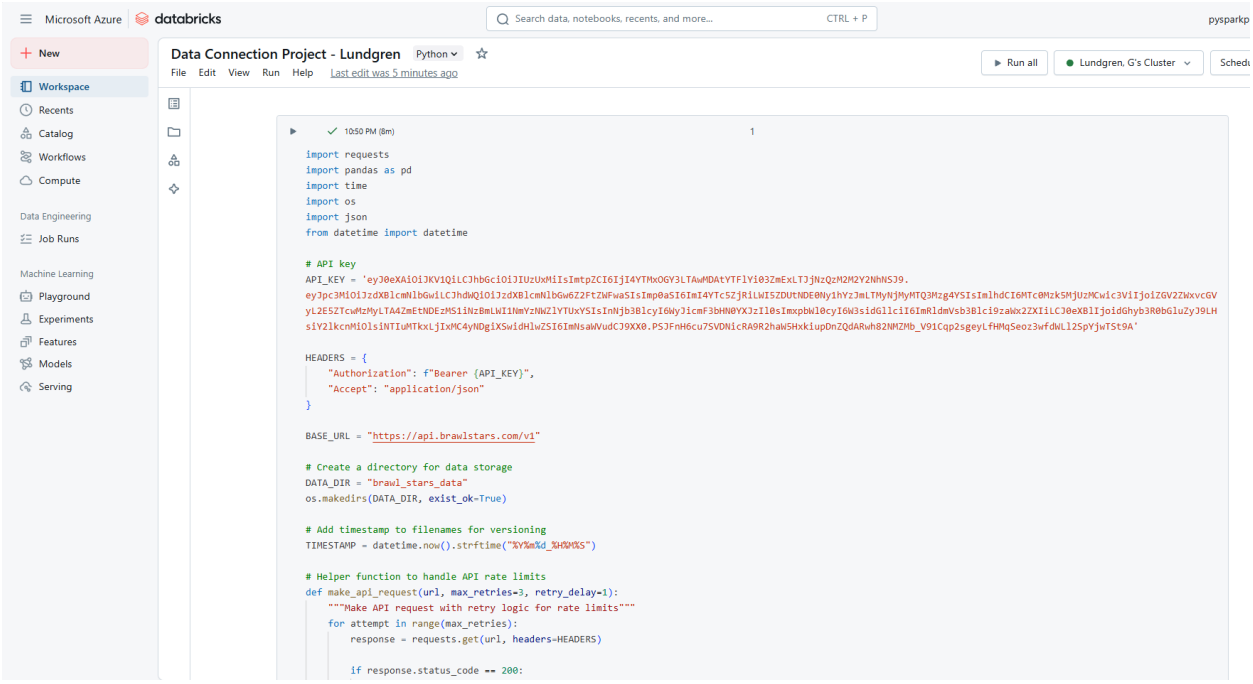
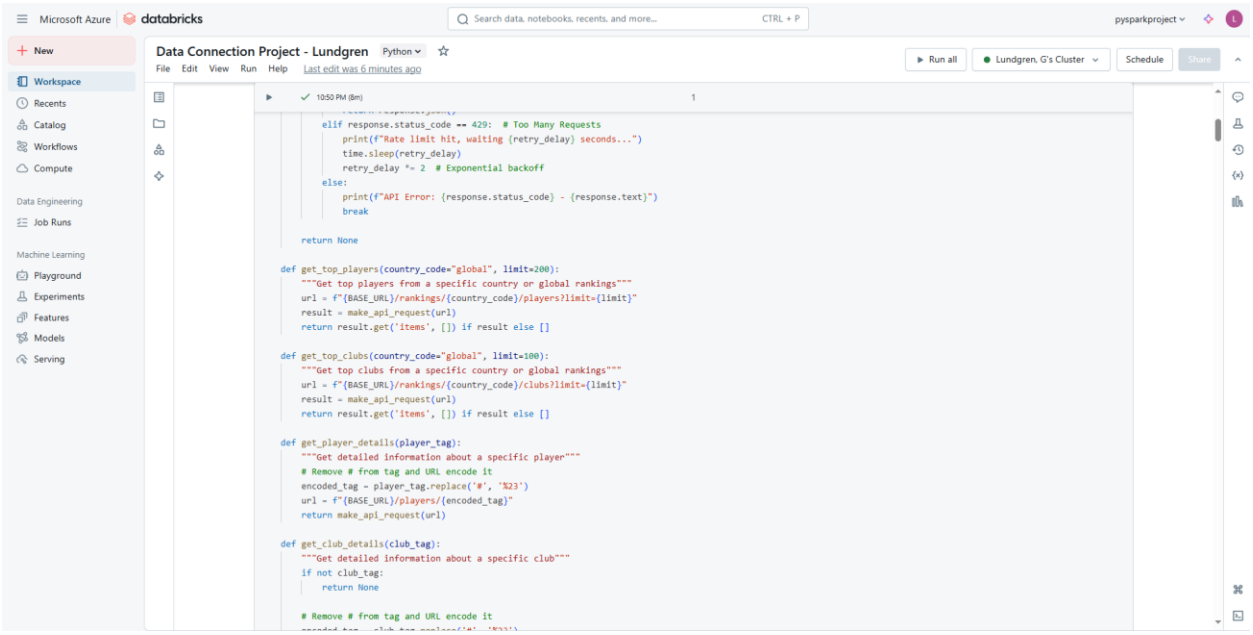
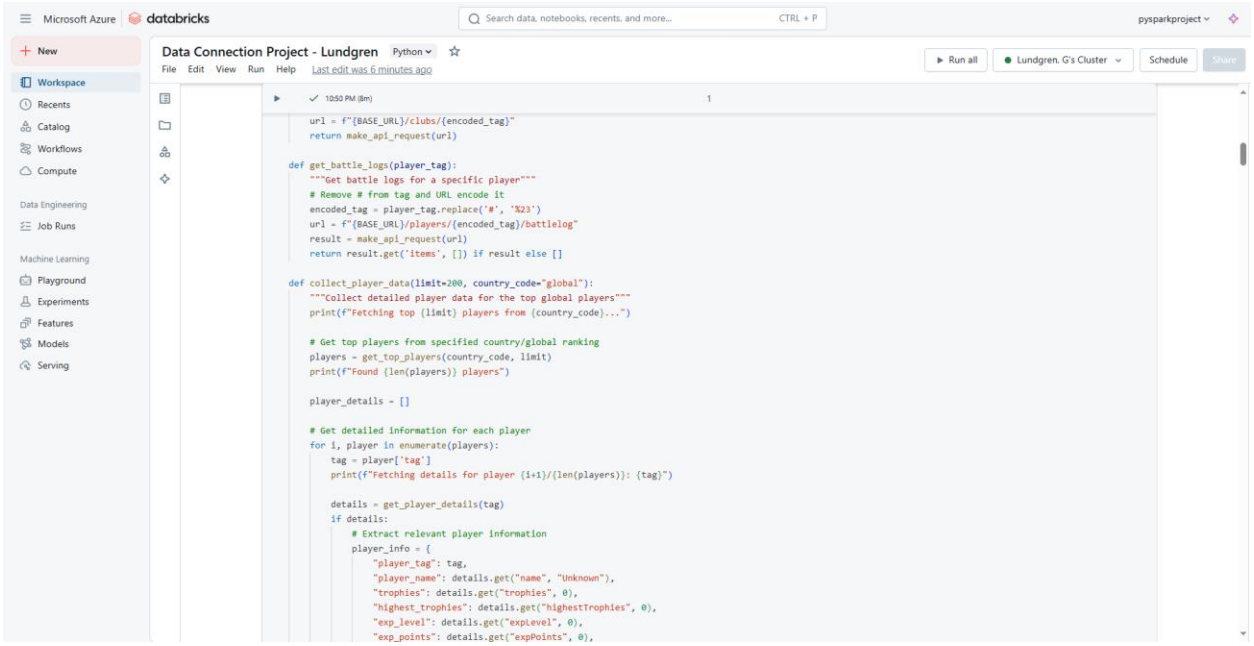
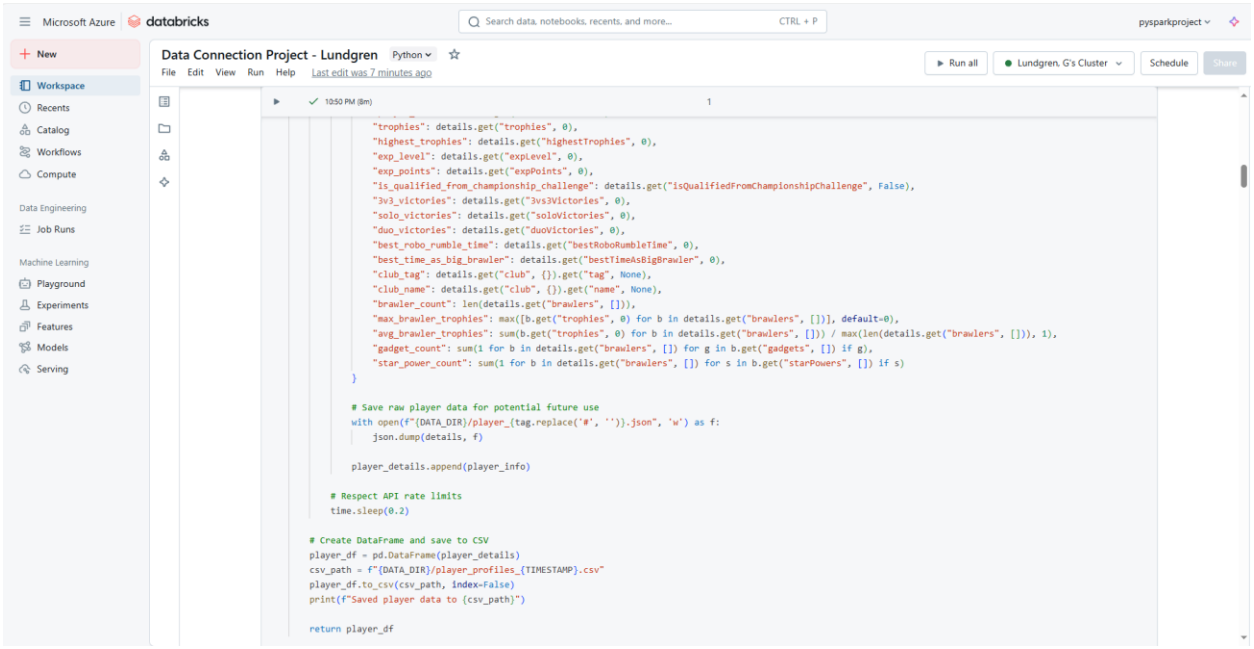
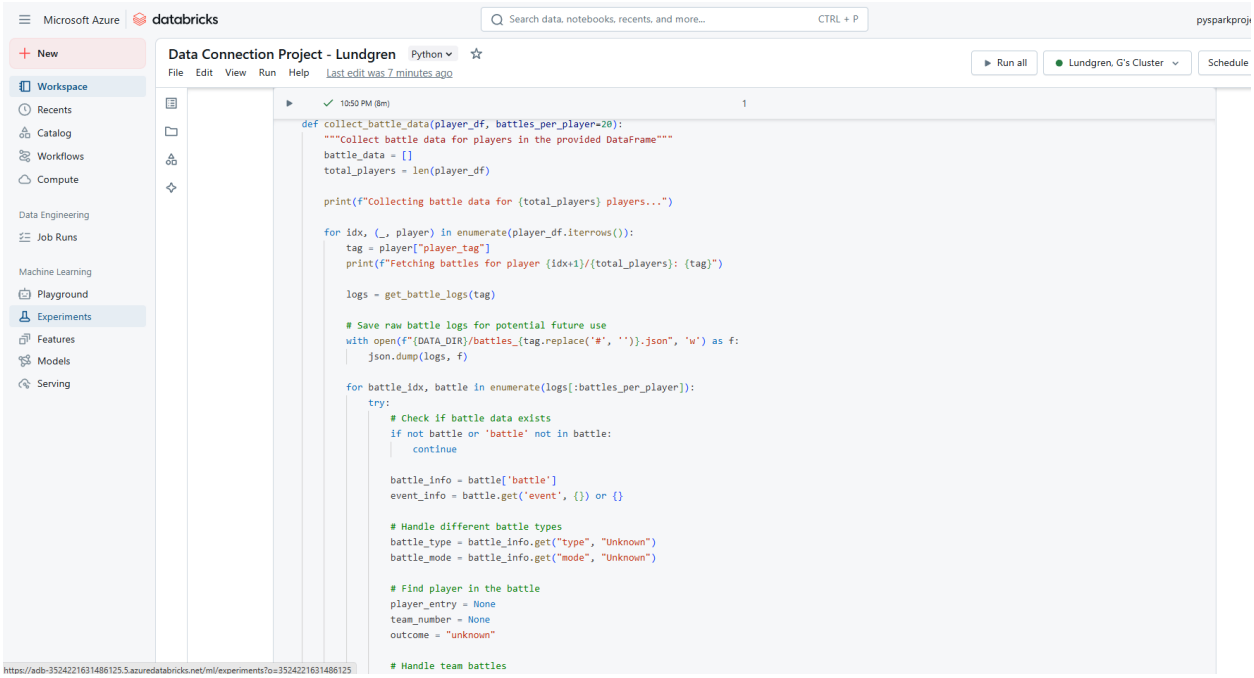
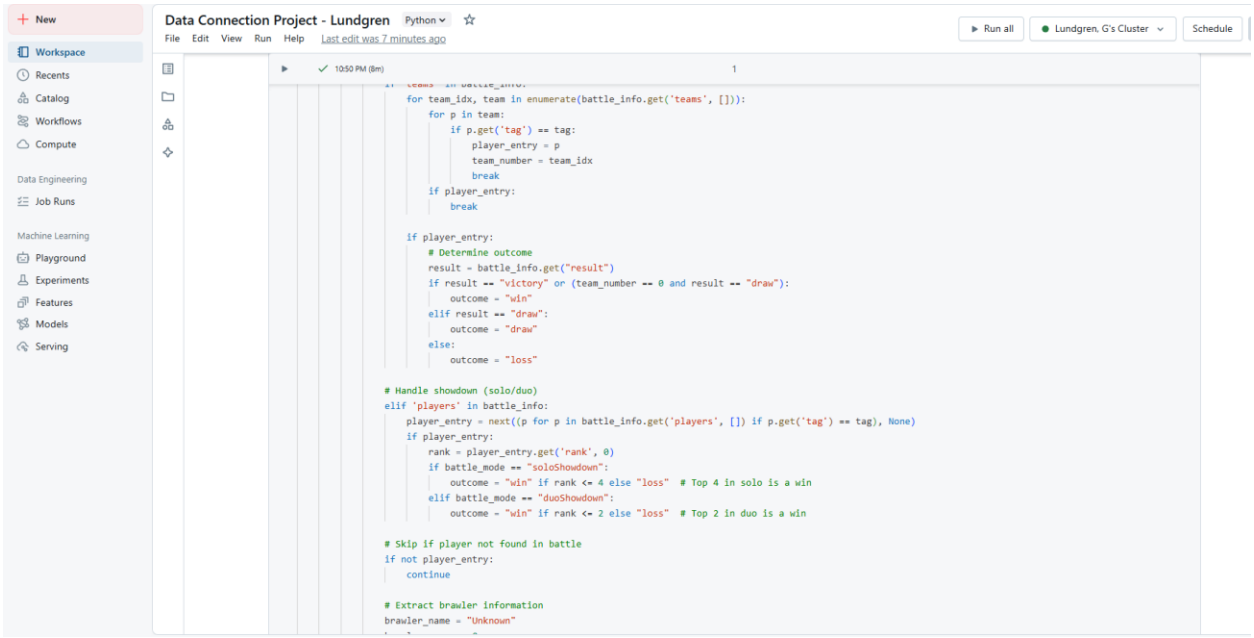

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
pysparkp
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 5 minutes ago
Run all
Lundgren, G's Cluster
10:50 PM (8m)
import requests
import pandas as pd
import time
import os
import json
from datetime import datetime
# API key
API_KEY = 'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiIsImtpZCI6IjI4YTMxOGY3LTAwMDAtYTFlYi03ZmExLTJjNzQzM2M2Y2NhNSJ9.
eyJpc3MiOiJzdXBlcmNlbGwiLCJhdWQiOiJzdXBlcmNlbGw6Z2FtZWFwaSIsImp0aSI6ImI4YTc5ZjRiLWI5ZDUtNDE0Ny1hYzJmLTMyNjMyMTQ3Mzg4YSIsImlhdCI6MTc0Mzk5MjUzMCwic3ViIjoiZGV2ZWxvcGVyL2E5ZTcwMzMyLTA4ZmEtNDEzMS1iNzBmLWI1NmYzNWZlYTUxYSIsInNjb3BlcyI6WyJicmF3bHN0YXJzIl0sImxpbWl0cyI6W3sidGllciI6ImRldmVsb3BlciI6MTAsIm1heCI6MTAwfV19.
siY2lkcnMiOlsiNTIuMTkxLjIxMC4yNDgiXSwidHlwZSI6ImNsaWVudCJ9XX0.PSJFnH6cu7SVDNicRA9R2haW5HxkiupDnZQdARwh82NMZMb_V91Cqp2sgeyLfHMqSeoz3wfdWll2SpYjwTSt9A'
HEADERS = {
"Authorization": f"Bearer {API_KEY}",
"Accept": "application/json"
}
BASE_URL = "https://api.brawlstars.com/v1"
# Create a directory for data storage
DATA_DIR = "brawl_stars_data"
os.makedirs(DATA_DIR, exist_ok=True)
# Add timestamp to filenames for versioning
TIMESTAMP = datetime.now().strftime("%Y%m%d_%H%M%S")
# Helper function to handle API rate limits
def make_api_request(url, max_retries=3, retry_delay=1):
"""Make API request with retry logic for rate limits"""
for attempt in range(max_retries):
response = requests.get(url, headers=HEADERS)
if response.status_code == 200:

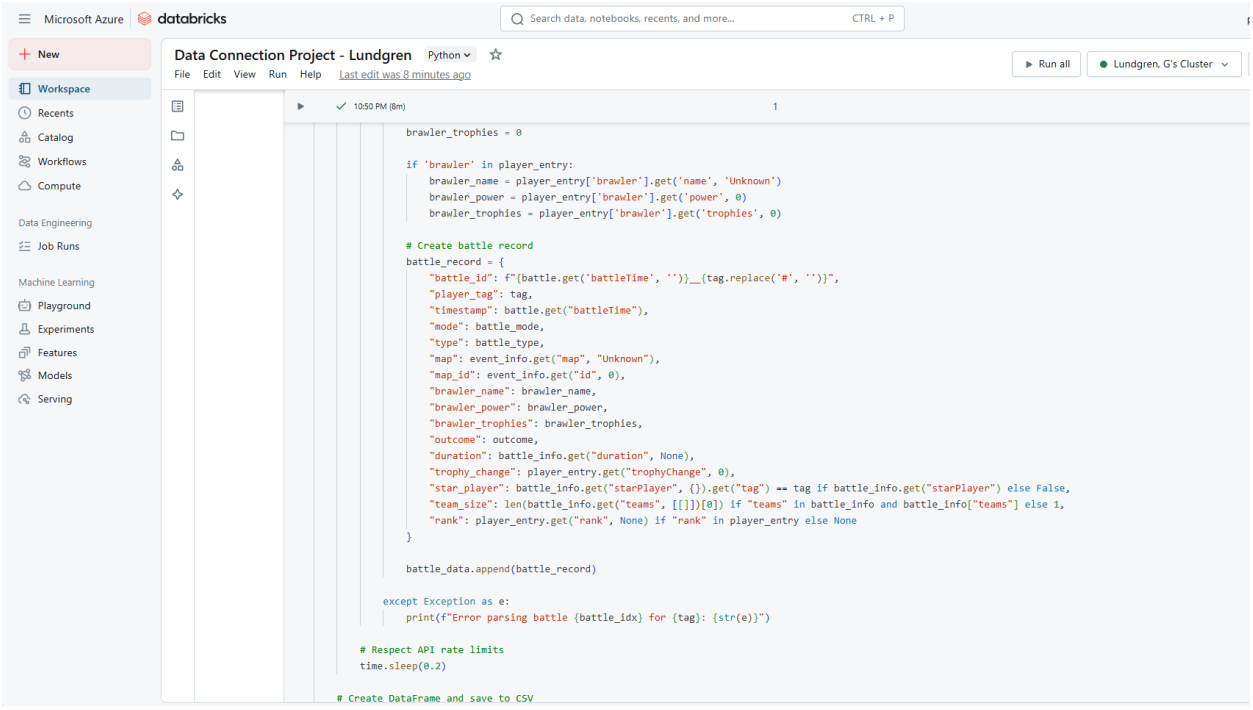

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
pysparkproject
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 6 minutes ago
Run all
Lundgren, G's Cluster
Schedule
Share
10:50 PM (8m)
elif response.status_code == 429: # Too Many Requests
print(f"Rate limit hit, waiting {retry_delay} seconds...")
time.sleep(retry_delay)
retry_delay *= 2 # Exponential backoff
else:
print(f"API Error: {response.status_code} - {response.text}")
break
return None
def get_top_players(country_code="global", limit=200):
"""Get top players from a specific country or global rankings"""
url = f"{BASE_URL}/rankings/{country_code}/players?limit={limit}"
result = make_api_request(url)
return result.get('items', []) if result else []
def get_top_clubs(country_code="global", limit=100):
"""Get top clubs from a specific country or global rankings"""
url = f"{BASE_URL}/rankings/{country_code}/clubs?limit={limit}"
result = make_api_request(url)
return result.get('items', []) if result else []
def get_player_details(player_tag):
"""Get detailed information about a specific player"""
# Remove # from tag and URL encode it
encoded_tag = player_tag.replace('#', '%23')
url = f"{BASE_URL}/players/{encoded_tag}"
return make_api_request(url)
def get_club_details(club_tag):
"""Get detailed information about a specific club"""
if not club_tag:
return None
Remove # from tag and URL encode it

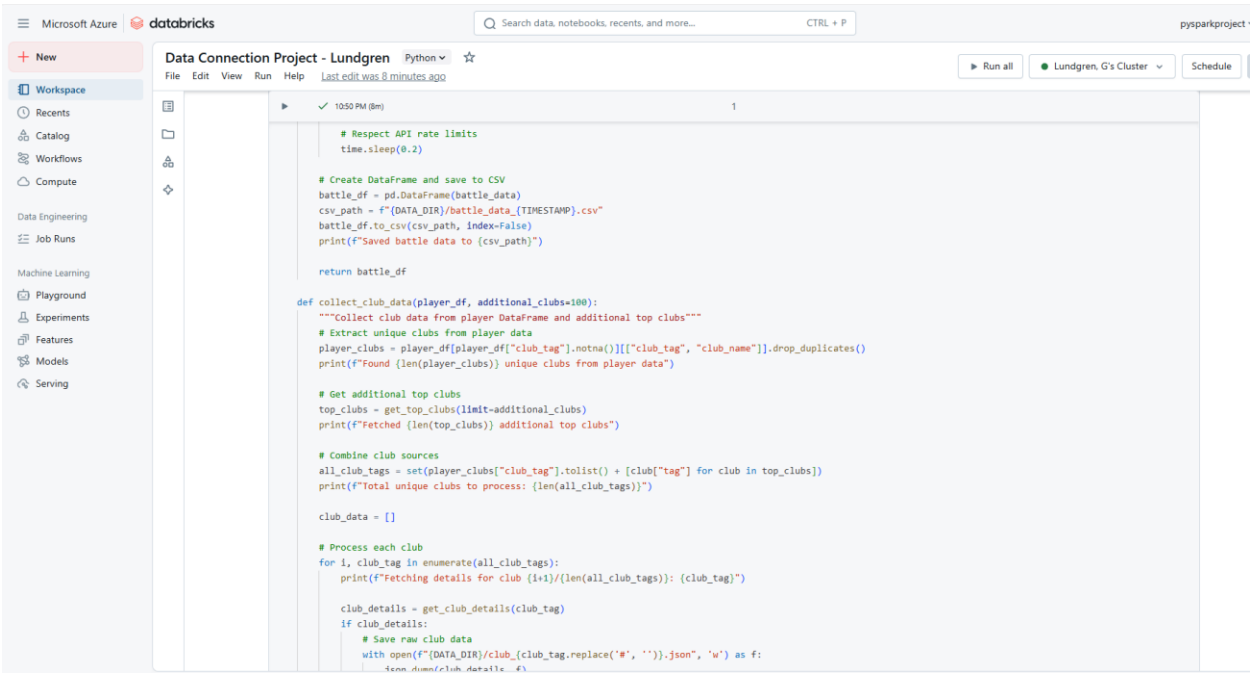

Microsoft Azure
databricks
Search data, notebooks, recents, and more... CTRL + P
pysparkproject
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 6 minutes ago
Run all
Lundgren, G's Cluster
Schedule
Share
10:50 PM (8m)
1
url = f"{BASE_URL}/clubs/{encoded_tag}"
return make_api_request(url)
def get_battle_logs(player_tag):
"""Get battle logs for a specific player"""
# Remove # from tag and URL encode it
encoded_tag = player_tag.replace('#', '%23')
url = f"{BASE_URL}/players/{encoded_tag}/battlelog"
result = make_api_request(url)
return result.get('items', []) if result else []
def collect_player_data(limit=200, country_code="global"):
"""Collect detailed player data for the top global players"""
print(f"Fetching top {limit} players from {country_code}...")
# Get top players from specified country/global ranking
players = get_top_players(country_code, limit)
print(f"Found {len(players)} players")
player_details = []
# Get detailed information for each player
for i, player in enumerate(players):
tag = player['tag']
print(f"Fetching details for player {i+1}/{len(players)}: {tag}")
details = get_player_details(tag)
if details:
# Extract relevant player information
player_info = {
"player_tag": tag,
"player_name": details.get("name", "Unknown"),
"trophies": details.get("trophies", 0),
"highest_trophies": details.get("highestTrophies", 0),
"exp_level": details.get("expLevel", 0),
"exp_points": details.get("expPoints", 0),

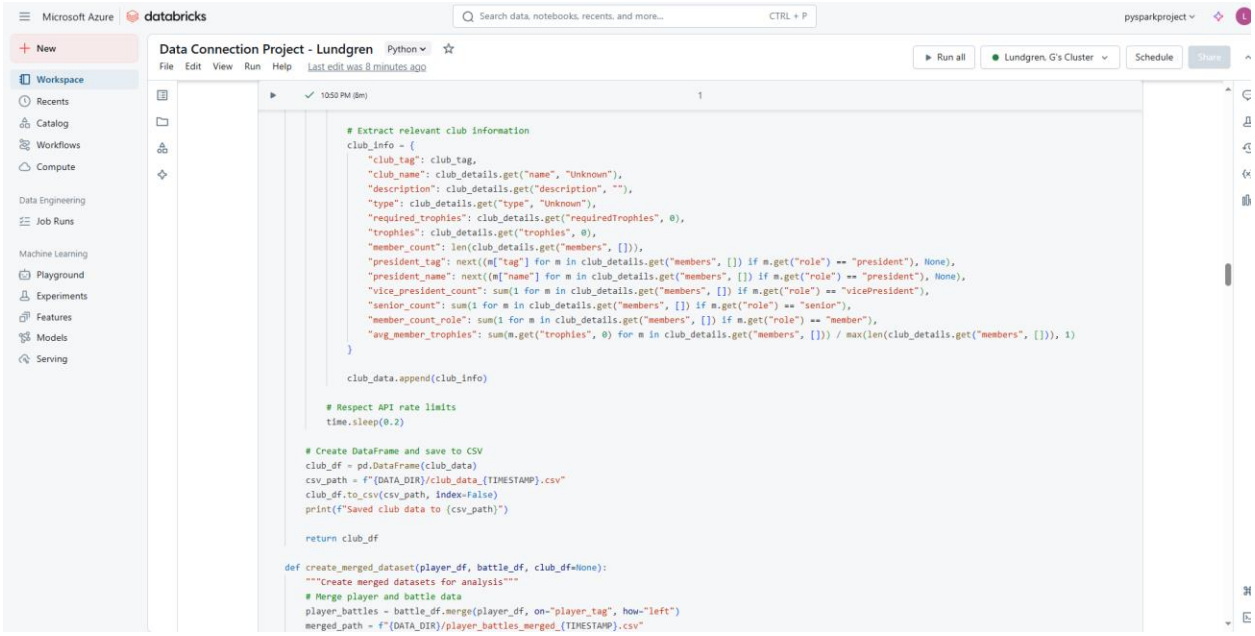

Microsoft Azure
databricks
Search data, notebooks, recents, and more... CTRL + P
pysparkproject
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 7 minutes ago
Run all
Lundgren, G's Cluster
Schedule
Share
10:50 PM (8m)
1
"trophies": details.get("trophies", 0),
"highest_trophies": details.get("highestTrophies", 0),
"exp_level": details.get("expLevel", 0),
"exp_points": details.get("expPoints", 0),
"is_qualified_from_championship_challenge": details.get("isQualifiedFromChampionshipChallenge", False),
"3v3_victories": details.get("3vs3Victories", 0),
"solo_victories": details.get("soloVictories", 0),
"duo_victories": details.get("duoVictories", 0),
"best_robo_rumble_time": details.get("bestRoboRumbleTime", 0),
"best_time_as_big_brawler": details.get("bestTimeAsBigBrawler", 0),
"club_tag": details.get("club", {}).get("tag", None),
"club_name": details.get("club", {}).get("name", None),
"brawler_count": len(details.get("brawlers", [])),
"max_brawler_trophies": max([b.get("trophies", 0) for b in details.get("brawlers", [])], default=0),
"avg_brawler_trophies": sum(b.get("trophies", 0) for b in details.get("brawlers", [])) / max(len(details.get("brawlers", [])), 1),
"gadget_count": sum(1 for b in details.get("brawlers", []) for g in b.get("gadgets", []) if g),
"star_power_count": sum(1 for b in details.get("brawlers", []) for s in b.get("starPowers", []) if s)
}
# Save raw player data for potential future use
with open(f"{DATA_DIR}/player_{tag.replace('#', '')}.json", 'w') as f:
json.dump(details, f)
player_details.append(player_info)
# Respect API rate limits
time.sleep(0.2)
# Create DataFrame and save to CSV
player_df = pd.DataFrame(player_details)
csv_path = f"{DATA_DIR}/player_profiles_{TIMESTAMP}.csv"
player_df.to_csv(csv_path, index=False)
print(f"Saved player data to {csv_path}")
return player_df

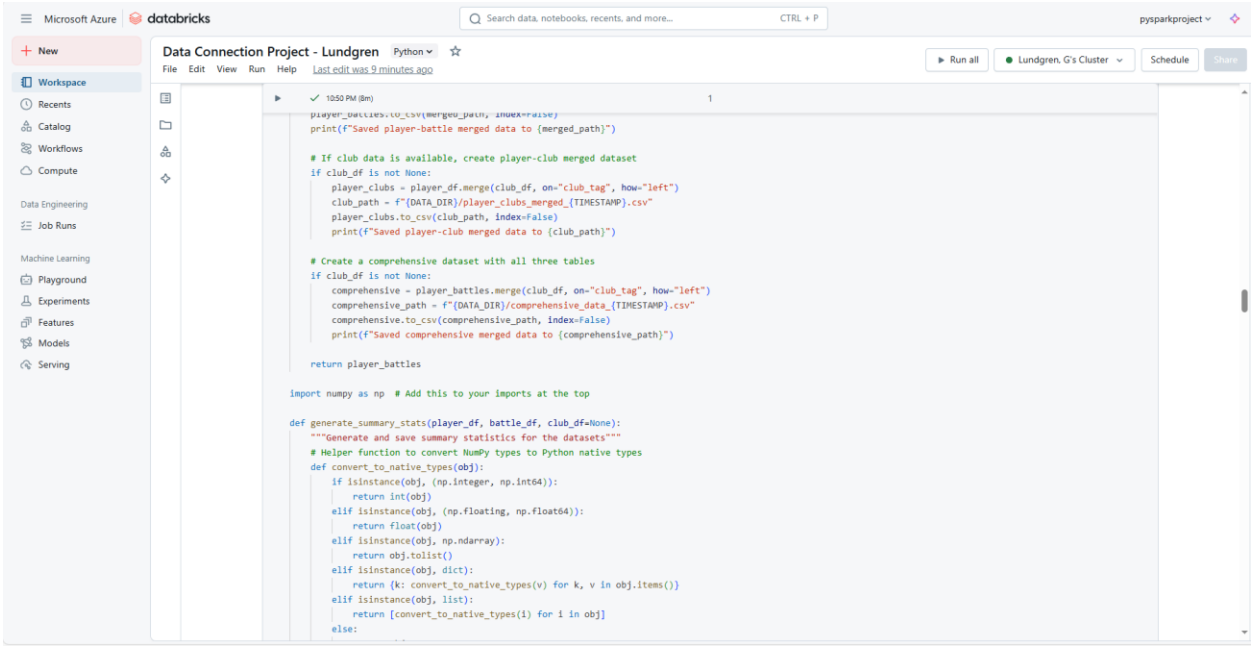

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 7 minutes ago
Run all
Lundgren, G's Cluster
Schedule
10:50 PM (8m)
def collect_battle_data(player_df, battles_per_player=20):
"""Collect battle data for players in the provided DataFrame"""
battle_data = []
total_players = len(player_df)
print(f"Collecting battle data for {total_players} players...")
for idx, (_, player) in enumerate(player_df.iterrows()):
tag = player["player_tag"]
print(f"Fetching battles for player {idx+1}/{total_players}: {tag}")
logs = get_battle_logs(tag)
# Save raw battle logs for potential future use
with open(f"{DATA_DIR}/battles_{tag.replace('#', '')}.json", 'w') as f:
json.dump(logs, f)
for battle_idx, battle in enumerate(logs[:battles_per_player]):
try:
# Check if battle data exists
if not battle or 'battle' not in battle:
continue
battle_info = battle['battle']
event_info = battle.get('event', {}) or {}
# Handle different battle types
battle_type = battle_info.get("type", "Unknown")
battle_mode = battle_info.get("mode", "Unknown")
# Find player in the battle
player_entry = None
team_number = None
outcome = "unknown"
# Handle team battles
https://adb-3524221631486125.5.azuredatabricks.net/ml/experiments?o=3524221631486125

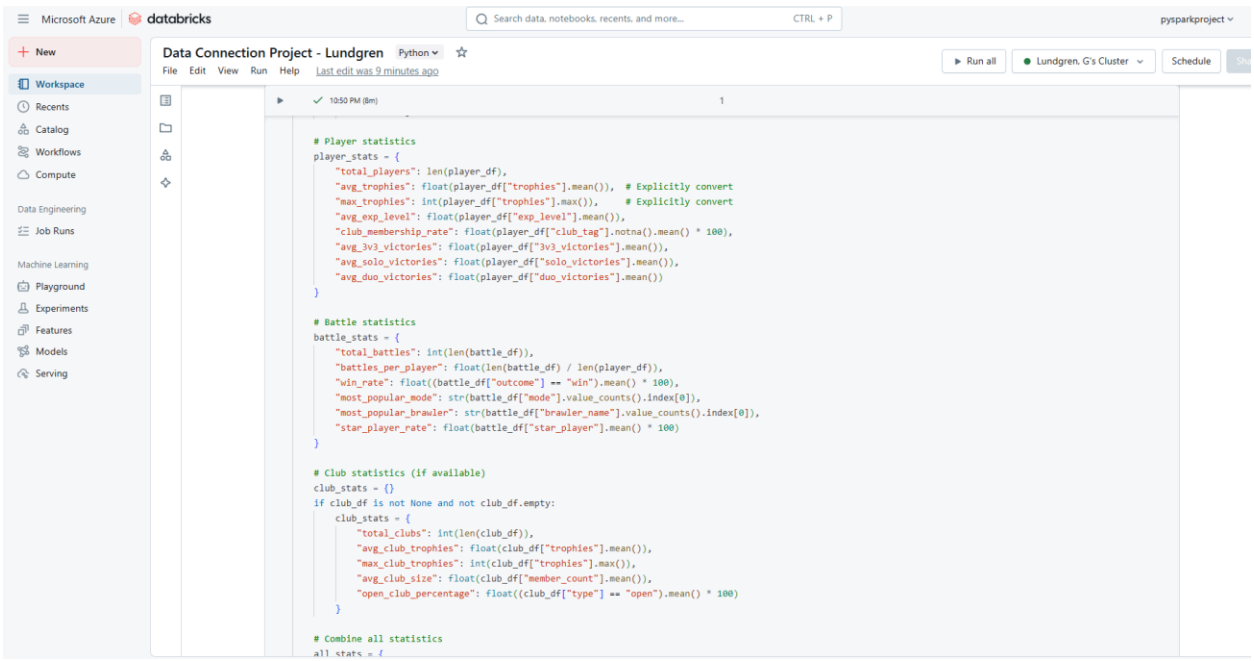

New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 7 minutes ago
Run all
Lundgren, G's Cluster
Schedule
10:50 PM (8m)
for team_idx, team in enumerate(battle_info.get('teams', [])):
for p in team:
if p.get('tag') == tag:
player_entry = p
team_number = team_idx
break
if player_entry:
break
if player_entry:
# Determine outcome
result = battle_info.get("result")
if result == "victory" or (team_number == 0 and result == "draw"):
outcome = "win"
elif result == "draw":
outcome = "draw"
else:
outcome = "loss"
# Handle showdown (solo/duo)
elif 'players' in battle_info:
player_entry = next((p for p in battle_info.get('players', []) if p.get('tag') == tag), None)
if player_entry:
rank = player_entry.get('rank', 0)
if battle_mode == "soloShowdown":
outcome = "win" if rank <= 4 else "loss" # Top 4 in solo is a win
elif battle_mode == "duoShowdown":
outcome = "win" if rank <= 2 else "loss" # Top 2 in duo is a win
# Skip if player not found in battle
if not player_entry:
continue
# Extract brawler information
brawler_name = "Unknown"

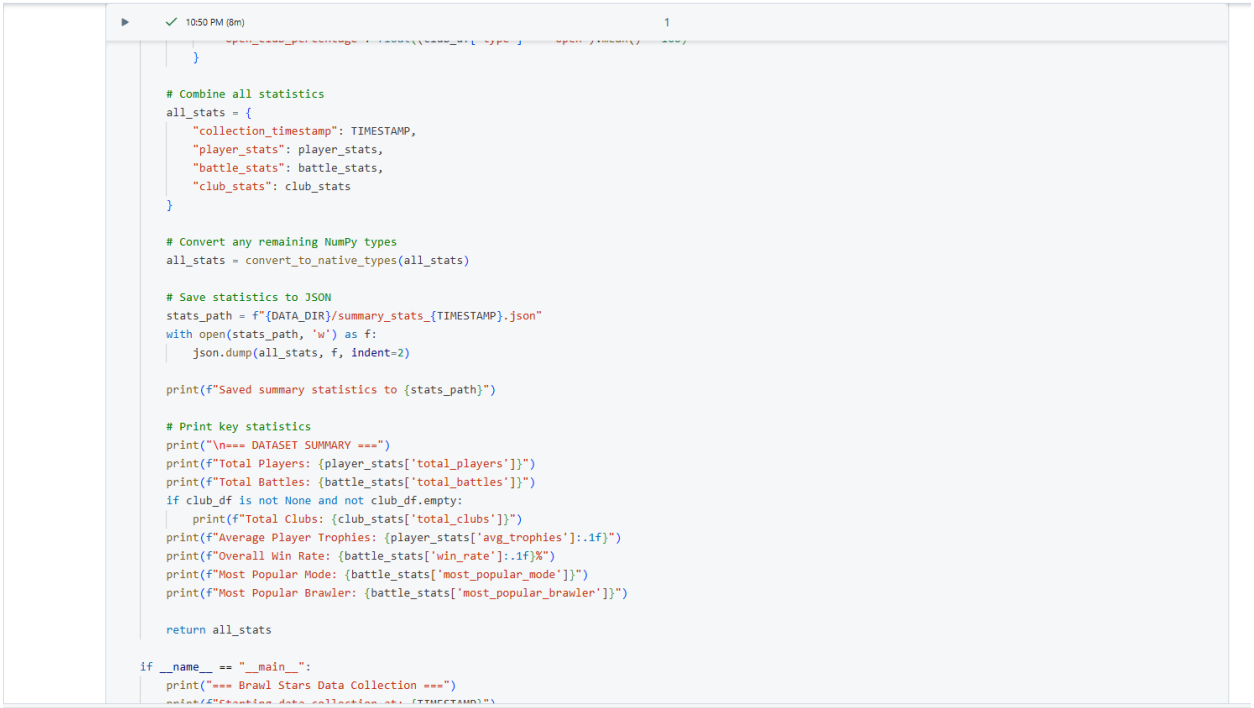

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 8 minutes ago
Run all
Lundgren, G's Cluster
10:50 PM (8m)
brawler_trophies = 0
if 'brawler' in player_entry:
brawler_name = player_entry['brawler'].get('name', 'Unknown')
brawler_power = player_entry['brawler'].get('power', 0)
brawler_trophies = player_entry['brawler'].get('trophies', 0)
# Create battle record
battle_record = {
"battle_id": f"{battle.get('battleTime', '')}_{tag.replace('#', '')}",
"player_tag": tag,
"timestamp": battle.get("battleTime"),
"mode": battle_mode,
"type": battle_type,
"map": event_info.get("map", "Unknown"),
"map_id": event_info.get("id", 0),
"brawler_name": brawler_name,
"brawler_power": brawler_power,
"brawler_trophies": brawler_trophies,
"outcome": outcome,
"duration": battle_info.get("duration", None),
"trophy_change": player_entry.get("trophyChange", 0),
"star_player": battle_info.get("starPlayer", {}).get("tag") == tag if battle_info.get("starPlayer") else False,
"team_size": len(battle_info.get("teams", [[]])[0]) if "teams" in battle_info and battle_info["teams"] else 1,
"rank": player_entry.get("rank", None) if "rank" in player_entry else None
}
battle_data.append(battle_record)
except Exception as e:
print(f"Error parsing battle {battle_idx} for {tag}: {str(e)}")
# Respect API rate limits
time.sleep(0.2)
# Create DataFrame and save to CSV


Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
pysparkproject
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 8 minutes ago
Run all
Lundgren, G's Cluster
Schedule
10:50 PM (8m)
# Respect API rate limits
time.sleep(0.2)
# Create DataFrame and save to CSV
battle_df = pd.DataFrame(battle_data)
csv_path = f"{DATA_DIR}/battle_data_{TIMESTAMP}.csv"
battle_df.to_csv(csv_path, index=False)
print(f"Saved battle data to {csv_path}")
return battle_df
def collect_club_data(player_df, additional_clubs=100):
"""Collect club data from player DataFrame and additional top clubs"""
# Extract unique clubs from player data
player_clubs = player_df[player_df["club_tag"].notna()][["club_tag", "club_name"]].drop_duplicates()
print(f"Found {len(player_clubs)} unique clubs from player data")
# Get additional top clubs
top_clubs = get_top_clubs(limit=additional_clubs)
print(f"Fetched {len(top_clubs)} additional top clubs")
# Combine club sources
all_club_tags = set(player_clubs["club_tag"].tolist() + [club["tag"] for club in top_clubs])
print(f"Total unique clubs to process: {len(all_club_tags)}")
club_data = []
# Process each club
for i, club_tag in enumerate(all_club_tags):
print(f"Fetching details for club {i+1}/{len(all_club_tags)}: {club_tag}")
club_details = get_club_details(club_tag)
if club_details:
# Save raw club data
with open(f"{DATA_DIR}/club_{club_tag.replace('#', '')}.json", 'w') as f:

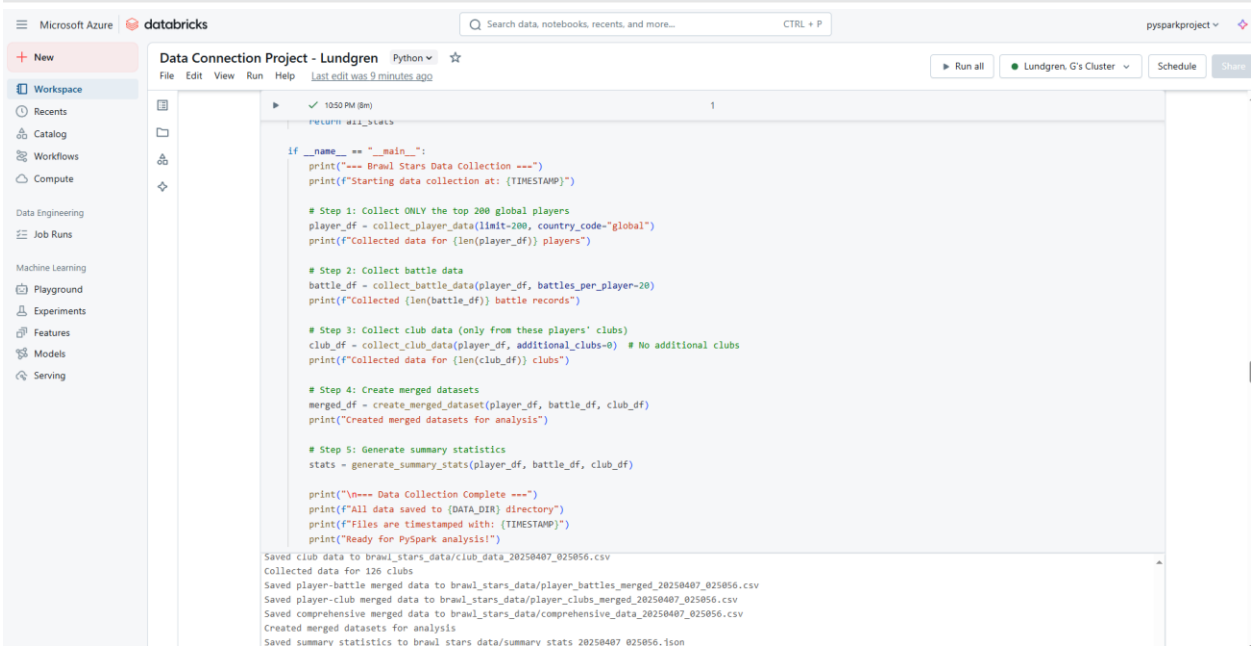

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
pysparkproject
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 8 minutes ago
Run all
Lundgren, G's Cluster
Schedule
Share
10:50 PM (8m)
1
# Extract relevant club information
club_info = {
"club_tag": club_tag,
"club_name": club_details.get("name", "Unknown"),
"description": club_details.get("description", ""),
"type": club_details.get("type", "Unknown"),
"required_trophies": club_details.get("requiredTrophies", 0),
"trophies": club_details.get("trophies", 0),
"member_count": len(club_details.get("members", [])),
"president_tag": next((m["tag"] for m in club_details.get("members", []) if m.get("role") == "president"), None),
"president_name": next((m["name"] for m in club_details.get("members", []) if m.get("role") == "president"), None),
"vice_president_count": sum(1 for m in club_details.get("members", []) if m.get("role") == "vicePresident"),
"senior_count": sum(1 for m in club_details.get("members", []) if m.get("role") == "senior"),
"member_count_role": sum(1 for m in club_details.get("members", []) if m.get("role") == "member"),
"avg_member_trophies": sum(m.get("trophies", 0) for m in club_details.get("members", [])) / max(len(club_details.get("members", [])), 1)
}
club_data.append(club_info)
# Respect API rate limits
time.sleep(0.2)
# Create DataFrame and save to CSV
club_df = pd.DataFrame(club_data)
csv_path = f"{DATA_DIR}/club_data_{TIMESTAMP}.csv"
club_df.to_csv(csv_path, index=False)
print(f"Saved club data to {csv_path}")
return club_df
def create_merged_dataset(player_df, battle_df, club_df=None):
"""Create merged datasets for analysis"""
# Merge player and battle data
player_battles = battle_df.merge(player_df, on="player_tag", how="left")
merged_path = f"{DATA_DIR}/player_battles_merged_{TIMESTAMP}.csv"


Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
pysparkproject
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 9 minutes ago
Run all
Lundgren, G's Cluster
Schedule
Share
10:50 PM (8m)
print(f"Saved player-battle merged data to {merged_path}")
# If club data is available, create player-club merged dataset
if club_df is not None:
player_clubs = player_df.merge(club_df, on="club_tag", how="left")
club_path = f"{DATA_DIR}/player_clubs_merged_{TIMESTAMP}.csv"
player_clubs.to_csv(club_path, index=False)
print(f"Saved player-club merged data to {club_path}")
# Create a comprehensive dataset with all three tables
if club_df is not None:
comprehensive = player_battles.merge(club_df, on="club_tag", how="left")
comprehensive_path = f"{DATA_DIR}/comprehensive_data_{TIMESTAMP}.csv"
comprehensive.to_csv(comprehensive_path, index=False)
print(f"Saved comprehensive merged data to {comprehensive_path}")
return player_battles
import numpy as np # Add this to your imports at the top
def generate_summary_stats(player_df, battle_df, club_df=None):
"""Generate and save summary statistics for the datasets"""
# Helper function to convert NumPy types to Python native types
def convert_to_native_types(obj):
if isinstance(obj, (np.integer, np.int64)):
return int(obj)
elif isinstance(obj, (np.floating, np.float64)):
return float(obj)
elif isinstance(obj, np.ndarray):
return obj.tolist()
elif isinstance(obj, dict):
return {k: convert_to_native_types(v) for k, v in obj.items()}
elif isinstance(obj, list):
return [convert_to_native_types(i) for i in obj]
else:


Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
pysparkproject
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 9 minutes ago
Run all
Lundgren, G's Cluster
Schedule
10:50 PM (8m)
# Player statistics
player_stats = {
"total_players": len(player_df),
"avg_trophies": float(player_df["trophies"].mean()), # Explicitly convert
"max_trophies": int(player_df["trophies"].max()), # Explicitly convert
"avg_exp_level": float(player_df["exp_level"].mean()),
"club_membership_rate": float(player_df["club_tag"].notna().mean() * 100),
"avg_3v3_victories": float(player_df["3v3_victories"].mean()),
"avg_solo_victories": float(player_df["solo_victories"].mean()),
"avg_duo_victories": float(player_df["duo_victories"].mean())
}
# Battle statistics
battle_stats = {
"total_battles": int(len(battle_df)),
"battles_per_player": float(len(battle_df) / len(player_df)),
"win_rate": float((battle_df["outcome"] == "win").mean() * 100),
"most_popular_mode": str(battle_df["mode"].value_counts().index[0]),
"most_popular_brawler": str(battle_df["brawler_name"].value_counts().index[0]),
"star_player_rate": float(battle_df["star_player"].mean() * 100)
}
# Club statistics (if available)
club_stats = {}
if club_df is not None and not club_df.empty:
club_stats = {
"total_clubs": int(len(club_df)),
"avg_club_trophies": float(club_df["trophies"].mean()),
"max_club_trophies": int(club_df["trophies"].max()),
"avg_club_size": float(club_df["member_count"].mean()),
"open_club_percentage": float((club_df["type"] == "open").mean() * 100)
}
# Combine all statistics
all_stats = {



```
    }

# Combine all statistics
all_stats = {
    "collection_timestamp": TIMESTAMP,
    "player_stats": player_stats,
    "battle_stats": battle_stats,
    "club_stats": club_stats
}

# Convert any remaining NumPy types
all_stats = convert_to_native_types(all_stats)

# Save statistics to JSON
stats_path = f"{DATA_DIR}/summary_stats_{TIMESTAMP}.json"
with open(stats_path, 'w') as f:
    json.dump(all_stats, f, indent=2)

print(f"Saved summary statistics to {stats_path}")

# Print key statistics
print("\n=== DATASET SUMMARY ===")
print(f"Total Players: {player_stats['total_players']}")
print(f"Total Battles: {battle_stats['total_battles']}")
if club_df is not None and not club_df.empty:
    print(f"Total Clubs: {club_stats['total_clubs']}")
print(f"Average Player Trophies: {player_stats['avg_trophies']:.1f}")
print(f"Overall Win Rate: {battle_stats['win_rate']:.1f}%")
print(f"Most Popular Mode: {battle_stats['most_popular_mode']}")
print(f"Most Popular Brawler: {battle_stats['most_popular_brawler']}")

return all_stats

if __name__ == "__main__":
    print("=== Brawl Stars Data Collection ===")
```



```
if __name__ == "__main__":
    print("=== Brawl Stars Data Collection ===")
    print(f"Starting data collection at: {TIMESTAMP}")

    # Step 1: Collect ONLY the top 200 global players
    player_df = collect_player_data(limit=200, country_code="global")
    print(f"Collected data for {len(player_df)} players")

    # Step 2: Collect battle data
    battle_df = collect_battle_data(player_df, battles_per_player=20)
    print(f"Collected {len(battle_df)} battle records")

    # Step 3: Collect club data (only from these players' clubs)
    club_df = collect_club_data(player_df, additional_clubs=0)  # No additional clubs
    print(f"Collected data for {len(club_df)} clubs")

    # Step 4: Create merged datasets
    merged_df = create_merged_dataset(player_df, battle_df, club_df)
    print("Created merged datasets for analysis")

    # Step 5: Generate summary statistics
    stats = generate_summary_stats(player_df, battle_df, club_df)

    print("\n=== Data Collection Complete ===")
    print(f"All data saved to {DATA_DIR} directory")
    print(f"Files are timestamped with: {TIMESTAMP}")
    print("Ready for PySpark analysis!")
```

```
Saved club data to brawl_stars_data/club_data_20250407_025056.csv
Collected data for 126 clubs
Saved player-battle merged data to brawl_stars_data/player_battles_merged_20250407_025056.csv
Saved player-club merged data to brawl_stars_data/player_clubs_merged_20250407_025056.csv
Saved comprehensive merged data to brawl_stars_data/comprehensive_data_20250407_025056.csv
Created merged datasets for analysis
Saved summary statistics to brawl_stars_data/summary_stats_20250407_025056.json
```

copied output:” `=== Brawl Stars Data Collection === Starting data collection at: 20250407_025056 Fetching top 200 players from global... Found 200 players Fetching details for player 1/200: #VLQPVPY Fetching details for player 2/200: #PR9U2JL Fetching details for player 3/200: #29L2JQPGL Fetching details for player 4/200: #LGVY0QGP9 Fetching details for player 5/200: #UJ29R022 Fetching details for player 6/200: #R2PG98JU Fetching details for player 7/200: #PQ28R9RQ Fetching details for player 8/200: #22GQU2LLP Fetching details for player 9/200: #808GPCQ0V Fetching details for player 10/200: #2C989CR0G Fetching details for player 11/200: #L880UPQJP Fetching details for player 12/200: #PVR9ULPRG Fetching details for player 13/200: #8JVV0P808 Fetching details for player 14/200: #RYJY2Y9UG Fetching details for player 15/200: #92QPYVLR Fetching`

details for player 16/200: #20QQYGV Fetching details for player 17/200: #2JUJP2LVR Fetching details for player 18/200: #92J80YL2J Fetching details for player 19/200: #92Y2U2LYV Fetching details for player 20/200: #VC9PYLVR Fetching details for player 21/200: #98P0LQL0G Fetching details for player 22/200: #Y0QGU2QG8 Fetching details for player 23/200: #LVQR8CLY0 Fetching details for player 24/200: #JL8YRCJ Fetching details for player 25/200: #2C9VQ0R0U Fetching details for player 26/200: #QGG2UUV0 Fetching details for player 27/200: #8VU00220C Fetching details for player 28/200: #8J8V2RUG0 Fetching details for player 29/200: #2UVUR0GJC Fetching details for player 30/200: #QLQ8UJRV Fetching details for player 31/200: #89UG2JP9L Fetching details for player 32/200: #VVLQ88JU Fetching details for player 33/200: #92YCG008G Fetching details for player 34/200: #208J8QL02 Fetching details for player 35/200: #YCQ8299GU Fetching details for player 36/200: #208QYQ022 Fetching details for player 37/200: #YV8U29VLY Fetching details for player 38/200: #PGQ9QQU00 Fetching details for player 39/200: #2PGC28UUG Fetching details for player 40/200: #80VRC2YUP Fetching details for player 41/200: #202CC8CJG Fetching details for player 42/200: #PJRC8G8V9 Fetching details for player 43/200: #9PUYUV Fetching details for player 44/200: #YGU92PYU9 Fetching details for player 45/200: #8GYC0R0UC Fetching details for player 46/200: #8PG80UVUU Fetching details for player 47/200: #G8RVPR2L Fetching details for player 48/200: #82U9GYQR9 Fetching details for player 49/200: #898929YL0 Fetching details for player 50/200: #QVVP09V Fetching details for player 51/200: #UJQ9CYC Fetching details for player 52/200: #2QL89GVLV Fetching details for player 53/200: #PCL8LGVP9 Fetching details for player 54/200: #2QUQVP0V8 Fetching details for player 55/200: #2CYCQJ00Y Fetching details for player 56/200: #P9Q2RLJLY Fetching details for player 57/200: #P2QQVUJJ9 Fetching details for player 58/200: #GYR0L0JG Fetching details for player 59/200: #8J9PPC880 Fetching details for player 60/200: #82GG8LL2 Fetching details for player 61/200: #LU2Q2L2RR Fetching details for player 62/200: #2G22CY0C9 Fetching details for player 63/200: #2R80CYL8L Fetching details for player 64/200: #YQQ2L2JV Fetching details for player 65/200: #2C2QV2VU0 Fetching details for player 66/200: #8PCQJJ8Y Fetching details for player 67/200: #9CRGVGQLR Fetching details for player 68/200: #2UJPGRR8R Fetching details for player 69/200: #282VGGG0R Fetching details for player 70/200: #9QPJ2YLRV Fetching details for player 71/200: #28YGY9PP Fetching details for player 72/200: #90YPRVRGP Fetching details for player 73/200: #8C2LC0G9R Fetching details for player 74/200: #2R8YCCJQ0 Fetching details for player 75/200: #9929P98UC Fetching details for player 76/200: #2VPYLGR88 Fetching details for player 77/200: #8PG0LC0VR Fetching details for player 78/200: #8QRL0VCPJ Fetching details for player 79/200: #Y898GQR2Q Fetching details for player 80/200: #9QYURP0L9 Fetching details for player 81/200: #Y9YVR082 Fetching details for player 82/200: #L2RYC9YPP Fetching details for player 83/200: #GVR2Y2GC Fetching details for player 84/200: #8PVJLLU8R Fetching details for player 85/200: #PQLJL9PU8 Fetching details for player 86/200: #2JJV98GQY Fetching details for player 87/200: #PRRP0J880 Fetching details for player 88/200: #9YG8UJQGL Fetching details for player 89/200: #2U9U8U00P Fetching details for player 90/200: #YCQ29Q0UG Fetching details for player 91/200: #2QCQJPLRC Fetching details for player 92/200: #2YUPURGU2 Fetching details for player 93/200: #8P20YCQ9R Fetching details for player 94/200: #8QRJY9JQ8 Fetching details for player 95/200: #8GR2L8V82 Fetching details for player 96/200: #928C2UPCJ Fetching details for player 97/200: #8QPU8JL2J Fetching details for player 98/200: #2LVP9Y9PL Fetching details for player 99/200: #90PU009PR Fetching details for player 100/200: #VYQUG9L2 Fetching details for player 101/200: #R9YC8LUQ Fetching details for player 102/200: #82CVLYC08 Fetching details for player 103/200: #RRJGCQVV Fetching details for player 104/200: #YQL20RGRC Fetching details for player 105/200: #PVVVRLJQ2 Fetching details for player 106/200: #9VJGQ999L Fetching details for player 107/200:

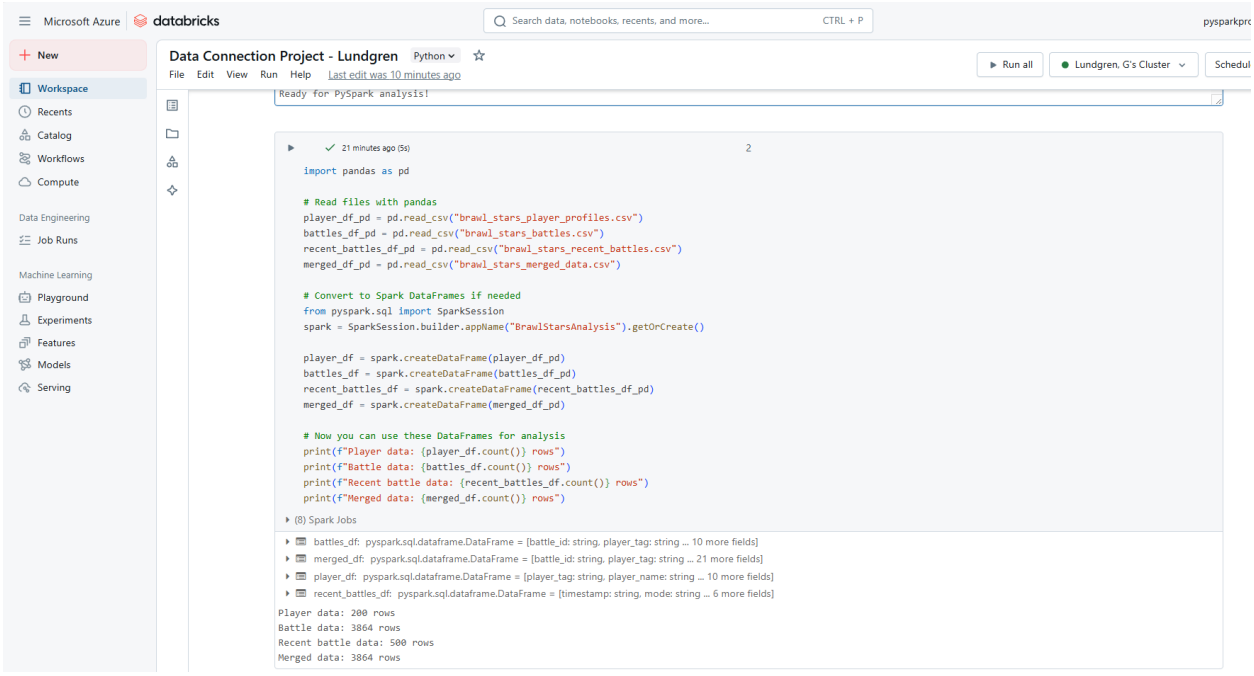
#9GVGY29QC Fetching details for player 108/200: #YU2Q2L8VJ Fetching details for player 109/200: #2L8Q82CQ Fetching details for player 110/200: #CYU9JRRL Fetching details for player 111/200: #2P8R2Y2PL Fetching details for player 112/200: #2YVP0RRGQ Fetching details for player 113/200: #RQQ08UV Fetching details for player 114/200: #LP82CYGQ Fetching details for player 115/200: #88YLJ0CRP Fetching details for player 116/200: #8Q2Q0Y8P Fetching details for player 117/200: #GYCQPUU0 Fetching details for player 118/200: #2PUY2QRCV Fetching details for player 119/200: #2JUYUJ9Q9 Fetching details for player 120/200: #8JCJCL0R9 Fetching details for player 121/200: #UCVQQ9Y Fetching details for player 122/200: #L9RLCG0L Fetching details for player 123/200: #98YYCYCUU Fetching details for player 124/200: #920J9G20V Fetching details for player 125/200: #9QUGVRJJG Fetching details for player 126/200: #J0PQ20LQ Fetching details for player 127/200: #8VCQCLJ22 Fetching details for player 128/200: #22URQ9YPU Fetching details for player 129/200: #9C8QUYC Fetching details for player 130/200: #9YJ8UJ88 Fetching details for player 131/200: #2QRV8YP8Q Fetching details for player 132/200: #2UV2UVPVR Fetching details for player 133/200: #9CJ90LP02 Fetching details for player 134/200: #CCUL0CRR Fetching details for player 135/200: #80QJPP92C Fetching details for player 136/200: #9GQY0P0L Fetching details for player 137/200: #80V2UC0P2 Fetching details for player 138/200: #PP8Q28JP8 Fetching details for player 139/200: #P28J29P22 Fetching details for player 140/200: #8U02LJRVY Fetching details for player 141/200: #8RGQRQGUQ Fetching details for player 142/200: #8YGYU2QYR Fetching details for player 143/200: #VLG0JJP0 Fetching details for player 144/200: #92QJGV2G0 Fetching details for player 145/200: #20JGG89YU Fetching details for player 146/200: #9VCL9G22 Fetching details for player 147/200: #2JPV2C89R Fetching details for player 148/200: #8GUJC2QR8 Fetching details for player 149/200: #2Q9Q9G8P2 Fetching details for player 150/200: #LGVP9QLRQ Fetching details for player 151/200: #2QC0RJ09 Fetching details for player 152/200: #9JJ8Y0RJ Fetching details for player 153/200: #8J29GUQCC Fetching details for player 154/200: #808U2JPUR Fetching details for player 155/200: #8R8PUG0L Fetching details for player 156/200: #Y892UQ82U Fetching details for player 157/200: #2YCPG0U0 Fetching details for player 158/200: #L00LCGCG Fetching details for player 159/200: #Q0P20Q2LJ Fetching details for player 160/200: #2G8GJ80GG Fetching details for player 161/200: #PQ2Y9LGY9 Fetching details for player 162/200: #8RVYUU9PG Fetching details for player 163/200: #2P029RC2L Fetching details for player 164/200: #9JQP0V98R Fetching details for player 165/200: #L899Q22 Fetching details for player 166/200: #2Q9RR8LLY Fetching details for player 167/200: #2UJYUP0R8 Fetching details for player 168/200: #9QPU9UU29 Fetching details for player 169/200: #YJUY029C8 Fetching details for player 170/200: #GR20QUU2 Fetching details for player 171/200: #8QR2UCCYU Fetching details for player 172/200: #P92J028L8 Fetching details for player 173/200: #P0P822VJ2 Fetching details for player 174/200: #288PVYL0P Fetching details for player 175/200: #UL2CJ8R Fetching details for player 176/200: #2JL8R08UC Fetching details for player 177/200: #9U0G2UGP0 Fetching details for player 178/200: #890PY0GJC Fetching details for player 179/200: #8Q2YJJRPP Fetching details for player 180/200: #22RJ98UUU Fetching details for player 181/200: #PRCPVJRLV Fetching details for player 182/200: #8Y9GLYQR2 Fetching details for player 183/200: #R02URJVJQ Fetching details for player 184/200: #8JQQ0QP88 Fetching details for player 185/200: #QPUCQVJ Fetching details for player 186/200: #Y9RGPRULP Fetching details for player 187/200: #9GJCPCGU2 Fetching details for player 188/200: #P2282QGYJ Fetching details for player 189/200: #2PUPU2RP9 Fetching details for player 190/200: #22PY0VQQC Fetching details for player 191/200: #9YUUVCRG9 Fetching details for player 192/200: #JVCC9VQ9 Fetching details for player 193/200: #P82URG298 Fetching details for player 194/200: #YL8JLQ2JR Fetching details for player 195/200: #289RC2JL0 Fetching details for player 196/200: #2R0L8PJCY Fetching details for player 197/200: #2QUGCYVYP Fetching

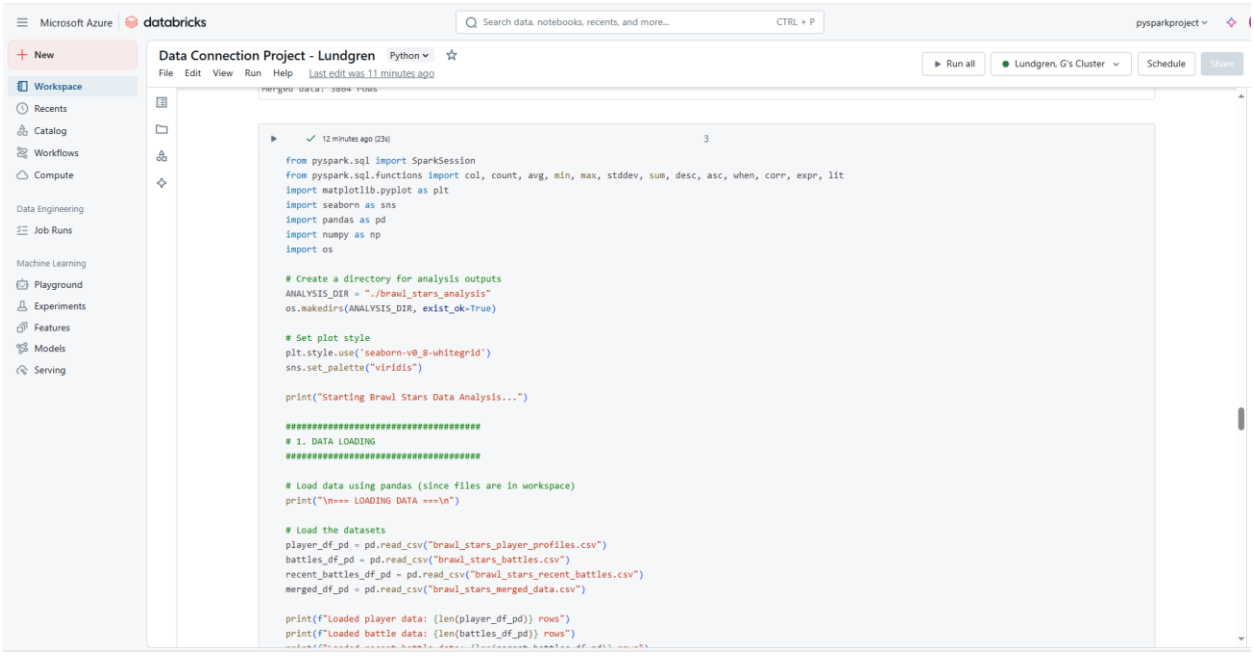
details for player 198/200: #88CC0R2GC Fetching details for player 199/200: #98Y9VYYR Fetching details for player 200/200: #LGC09YGQU Saved player data to brawl_stars_data/player_profiles_20250407_025056.csv Collected data for 200 players Collecting battle data for 200 players... Fetching battles for player 1/200: #VLQPVPY Fetching battles for player 2/200: #PR9U2JL Fetching battles for player 3/200: #29L2JQPGL Fetching battles for player 4/200: #LGVY0QGP9 Fetching battles for player 5/200: #UJ29R022 Fetching battles for player 6/200: #R2PG98JU Fetching battles for player 7/200: #PQ28R9RQ Fetching battles for player 8/200: #22GQU2LLP Fetching battles for player 9/200: #808GPCQ0V Fetching battles for player 10/200: #2C989CR0G Fetching battles for player 11/200: #L880UPQJP Fetching battles for player 12/200: #PVR9ULPRG Fetching battles for player 13/200: #8JVV0P808 Fetching battles for player 14/200: #RYJY2Y9UG Fetching battles for player 15/200: #92QPYVLR Fetching battles for player 16/200: #20QQYGV Fetching battles for player 17/200: #2JUJP2LVR Fetching battles for player 18/200: #92J80YL2J Fetching battles for player 19/200: #92Y2U2LYV Fetching battles for player 20/200: #VC9PYLVR Fetching battles for player 21/200: #98P0LQL0G Fetching battles for player 22/200: #Y0QGU2QG8 Fetching battles for player 23/200: #LVQR8CLY0 Fetching battles for player 24/200: #JL8YRCJ Fetching battles for player 25/200: #2C9VQ0R0U Fetching battles for player 26/200: #QGG2UUV0 Fetching battles for player 27/200: #8VU00220C Fetching battles for player 28/200: #8J8V2RUG0 Fetching battles for player 29/200: #2UVUR0GJC Fetching battles for player 30/200: #QLQ8UJRV Fetching battles for player 31/200: #89UG2JP9L Fetching battles for player 32/200: #VVLQ88JU Fetching battles for player 33/200: #92YCG008G Fetching battles for player 34/200: #208J8QL02 Fetching battles for player 35/200: #YCQ8299GU Fetching battles for player 36/200: #208QYQ022 Fetching battles for player 37/200: #YV8U29VLY Fetching battles for player 38/200: #PGQ9QQU00 Fetching battles for player 39/200: #2PGC28UUG Fetching battles for player 40/200: #80VRC2YUP Fetching battles for player 41/200: #202CC8CJG Fetching battles for player 42/200: #PJRC8G8V9 Fetching battles for player 43/200: #9PUYUV Fetching battles for player 44/200: #YGU92PYU9 Fetching battles for player 45/200: #8GYC0R0UC Fetching battles for player 46/200: #8PG80UVUU Fetching battles for player 47/200: #G8RVPR2L Fetching battles for player 48/200: #82U9GYQR9 Fetching battles for player 49/200: #898929YL0 Fetching battles for player 50/200: #QVVP09V Fetching battles for player 51/200: #UJQ9CYC Fetching battles for player 52/200: #2QL89GVLV Fetching battles for player 53/200: #PCL8LGVP9 Fetching battles for player 54/200: #2QUQVP0V8 Fetching battles for player 55/200: #2CYCQJ00Y Fetching battles for player 56/200: #P9Q2RLJLY Fetching battles for player 57/200: #P2QQVUJJ9 Fetching battles for player 58/200: #GYR0L0JG Fetching battles for player 59/200: #8J9PPC880 Fetching battles for player 60/200: #82GG8LL2 Fetching battles for player 61/200: #LU2Q2L2RR Fetching battles for player 62/200: #2G22CY0C9 Fetching battles for player 63/200: #2R80CYL8L Fetching battles for player 64/200: #YQQ2L2JV Fetching battles for player 65/200: #2C2QV2VU0 Fetching battles for player 66/200: #8PCQJJ8Y Fetching battles for player 67/200: #9CRGVGQLR Fetching battles for player 68/200: #2UJPGRR8R Fetching battles for player 69/200: #282VGGG0R Fetching battles for player 70/200: #9QPJ2YLRV Fetching battles for player 71/200: #28YGY9PP Fetching battles for player 72/200: #90YPRVRGP Fetching battles for player 73/200: #8C2LC0G9R Fetching battles for player 74/200: #2R8YCCJQ0 Fetching battles for player 75/200: #9929P98UC Fetching battles for player 76/200: #2VPYLGR88 Fetching battles for player 77/200: #8PG0LC0VR Fetching battles for player 78/200: #8QRL0VCPJ Fetching battles for player 79/200: #Y898GQR2Q Fetching battles for player 80/200: #9QYURP0L9 Fetching battles for player 81/200: #Y9YVR082 Fetching battles for player 82/200: #L2RYC9YPP Fetching battles for player 83/200: #GVR2Y2GC Fetching battles for player 84/200: #8PVJLLU8R Fetching battles for player 85/200: #PQLJL9PU8 Fetching battles for player 86/200:

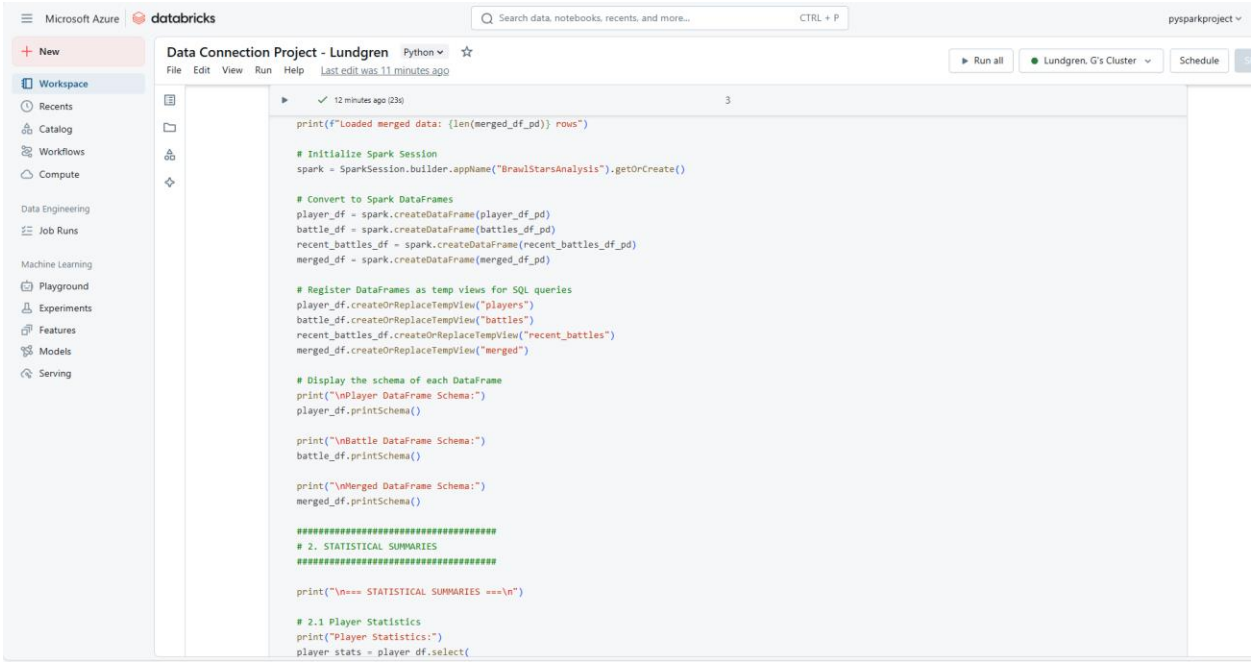
#2JJV98GQY Fetching battles for player 87/200: #PRRP0J880 Fetching battles for player 88/200: #9YG8UJQGL Fetching battles for player 89/200: #2U9U8U00P Fetching battles for player 90/200: #YCQ29Q0UG Fetching battles for player 91/200: #2QCQJPLRC Fetching battles for player 92/200: #2YUPURGU2 Fetching battles for player 93/200: #8P20YCQ9R Fetching battles for player 94/200: #8QRJY9JQ8 Fetching battles for player 95/200: #8GR2L8V82 Fetching battles for player 96/200: #928C2UPCJ Fetching battles for player 97/200: #8QPU8JL2J Fetching battles for player 98/200: #2LVP9Y9PL Fetching battles for player 99/200: #90PU009PR Fetching battles for player 100/200: #VYQUG9L2 Fetching battles for player 101/200: #R9YC8LUQ Fetching battles for player 102/200: #82CVLYC08 Fetching battles for player 103/200: #RRJGCQVV Fetching battles for player 104/200: #YQL20RGRC Fetching battles for player 105/200: #PVVVRLJQ2 Fetching battles for player 106/200: #9VJGQ999L Fetching battles for player 107/200: #9GVGY29QC Fetching battles for player 108/200: #YU2Q2L8VJ Fetching battles for player 109/200: #2L8Q82CQ Fetching battles for player 110/200: #CYU9JRRL Fetching battles for player 111/200: #2P8R2Y2PL Fetching battles for player 112/200: #2YVP0RRGQ Fetching battles for player 113/200: #RQQ08UV Fetching battles for player 114/200: #LP82CYGQ Fetching battles for player 115/200: #88YLJ0CRP Fetching battles for player 116/200: #8Q2Q0Y8P Fetching battles for player 117/200: #GYCQPUU0 Fetching battles for player 118/200: #2PUY2QRCV Fetching battles for player 119/200: #2JUYUJ9Q9 Fetching battles for player 120/200: #8JCJCL0R9 Fetching battles for player 121/200: #UCVQQ9Y Fetching battles for player 122/200: #L9RLCG0L Fetching battles for player 123/200: #98YYCYCUU Fetching battles for player 124/200: #920J9G20V Fetching battles for player 125/200: #9QUGVRJJG Fetching battles for player 126/200: #J0PQ20LQ Fetching battles for player 127/200: #8VCQCLJ22 Fetching battles for player 128/200: #22URQ9YPU Fetching battles for player 129/200: #9C8QUYC Fetching battles for player 130/200: #9YJ8UJ88 Fetching battles for player 131/200: #2QRV8YP8Q Fetching battles for player 132/200: #2UV2UVPVR Fetching battles for player 133/200: #9CJ90LP02 Fetching battles for player 134/200: #CCUL0CRR Fetching battles for player 135/200: #80QJPP92C Fetching battles for player 136/200: #9GQY0P0L Fetching battles for player 137/200: #80V2UC0P2 Fetching battles for player 138/200: #PP8Q28JP8 Fetching battles for player 139/200: #P28J29P22 Fetching battles for player 140/200: #8U02LJRVY Fetching battles for player 141/200: #8RGQRQGUQ Fetching battles for player 142/200: #8YGYU2QYR Fetching battles for player 143/200: #VLG0JJP0 Fetching battles for player 144/200: #92QJGV2G0 Fetching battles for player 145/200: #20JGG89YU Fetching battles for player 146/200: #9VCL9G22 Fetching battles for player 147/200: #2JPV2C89R Fetching battles for player 148/200: #8GUJC2QR8 Fetching battles for player 149/200: #2Q9Q9G8P2 Fetching battles for player 150/200: #LGVP9QLRQ Fetching battles for player 151/200: #2QC0RJ09 Fetching battles for player 152/200: #9JJ8Y0RJ Fetching battles for player 153/200: #8J29GUQCC Fetching battles for player 154/200: #808U2JPUR Fetching battles for player 155/200: #8R8PUG0L Fetching battles for player 156/200: #Y892UQ82U Fetching battles for player 157/200: #2YCPG0U0 Fetching battles for player 158/200: #L00LCGCG Fetching battles for player 159/200: #Q0P20Q2LJ Fetching battles for player 160/200: #2G8GJ80GG Fetching battles for player 161/200: #PQ2Y9LGY9 Fetching battles for player 162/200: #8RVYUU9PG Fetching battles for player 163/200: #2P029RC2L Fetching battles for player 164/200: #9JQP0V98R Fetching battles for player 165/200: #L899Q22 Fetching battles for player 166/200: #2Q9RR8LLY Fetching battles for player 167/200: #2UJYUP0R8 Fetching battles for player 168/200: #9QPU9UU29 Fetching battles for player 169/200: #YJUY029C8 Fetching battles for player 170/200: #GR20QUU2 Fetching battles for player 171/200: #8QR2UCCYU Fetching battles for player 172/200: #P92J028L8 Fetching battles for player 173/200: #P0P822VJ2 Fetching battles for player 174/200: #288PVYL0P Fetching battles for player 175/200: #UL2CJ8R Fetching battles for player 176/200: #2JL8R08UC Fetching battles for

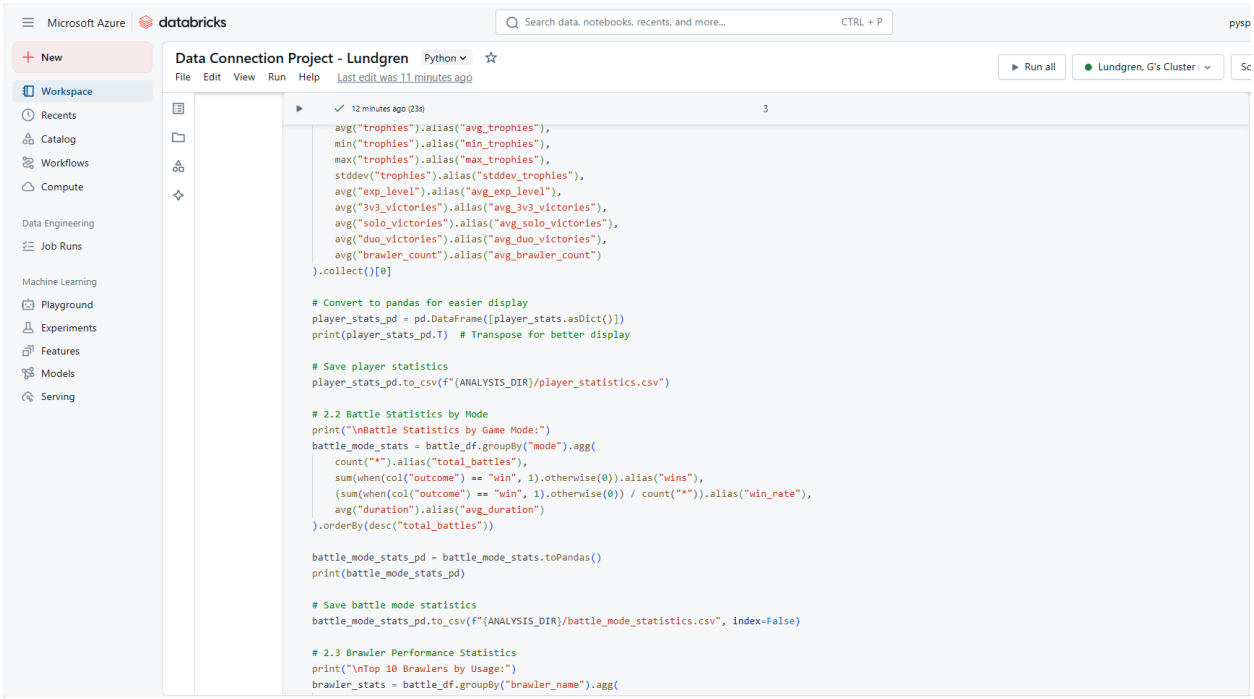
player 177/200: #9U0G2UGP0 Fetching battles for player 178/200: #890PY0GJC Fetching battles for player 179/200: #8Q2YJJRPP Fetching battles for player 180/200: #22RJ98UUU Fetching battles for player 181/200: #PRCPVJRLV Fetching battles for player 182/200: #8Y9GLYQR2 Fetching battles for player 183/200: #R02URJVJQ Fetching battles for player 184/200: #8JQQ0QP88 Fetching battles for player 185/200: #QPUCQVJ Fetching battles for player 186/200: #Y9RGPRULP Fetching battles for player 187/200: #9GJCPCGU2 Fetching battles for player 188/200: #P2282QGYJ Fetching battles for player 189/200: #2PUPU2RP9 Fetching battles for player 190/200: #22PY0VQQC Fetching battles for player 191/200: #9YUUVCRG9 Fetching battles for player 192/200: #JVCC9VQ9 Fetching battles for player 193/200: #P82URG298 Fetching battles for player 194/200: #YL8JLQ2JR Fetching battles for player 195/200: #289RC2JL0 Fetching battles for player 196/200: #2R0L8PJCY Fetching battles for player 197/200: #2QUGCYVYP Fetching battles for player 198/200: #88CC0R2GC Fetching battles for player 199/200: #98Y9VYYR Fetching battles for player 200/200: #LGC09YGQU Saved battle data to brawl_stars_data/battle_data_20250407_025056.csv Collected 4000 battle records Found 126 unique clubs from player data API Error: 400 - {"reason":"badRequest","message":"Invalid 'limit' parameter used in the request"} Fetched 0 additional top clubs Total unique clubs to process: 126 Fetching details for club 1/126: #JYLYV9CU Fetching details for club 2/126: #2JV2VJ8QP Fetching details for club 3/126: #2R0CUGGPV Fetching details for club 4/126: #8080RV82G Fetching details for club 5/126: #2VLGLR8CQ Fetching details for club 6/126: #2VGLJ0L0R Fetching details for club 7/126: #2VRQQJQ8R Fetching details for club 8/126: #80882L982 Fetching details for club 9/126: #2G8QUG28Y Fetching details for club 10/126: #20CQP2VPG Fetching details for club 11/126: #2UU0GCC9C Fetching details for club 12/126: #2PC8GPYLR Fetching details for club 13/126: #2VCY8PU9J Fetching details for club 14/126: #2VY99098G Fetching details for club 15/126: #2U9CQV8L2 Fetching details for club 16/126: #2YGU0LL90 Fetching details for club 17/126: #2JCUCLUC2 Fetching details for club 18/126: #2VCLV88VJ Fetching details for club 19/126: #2VJ898VUY Fetching details for club 20/126: #JY899QQL Fetching details for club 21/126: #8082VRRGJ Fetching details for club 22/126: #2JUQLGL98 Fetching details for club 23/126: #2UJQ2828Y Fetching details for club 24/126: #2VJJYGLC2 Fetching details for club 25/126: #2L98VPGVJ Fetching details for club 26/126: #2LPQ0RVRY Fetching details for club 27/126: #2RCJVJV2U Fetching details for club 28/126: #CRJRLPJY Fetching details for club 29/126: #2J0P9VV28 Fetching details for club 30/126: #2YJLUJ9JR Fetching details for club 31/126: #2GGQU0RPV Fetching details for club 32/126: #2ULJGRPV8 Fetching details for club 33/126: #2JVQ09YCJ Fetching details for club 34/126: #2LY89JJ22 Fetching details for club 35/126: #2GJ2GUPGV Fetching details for club 36/126: #QRU9C9GG Fetching details for club 37/126: #RQPU082P Fetching details for club 38/126: #2VJ9QRYCC Fetching details for club 39/126: #809PLYCQ8 Fetching details for club 40/126: #2VRQY202U Fetching details for club 41/126: #80P00R9PC Fetching details for club 42/126: #2YCGQVQUQ Fetching details for club 43/126: #JY8YYVPR Fetching details for club 44/126: #JYPJ0L8R Fetching details for club 45/126: #80P2QPUCC Fetching details for club 46/126: #J2G208PU Fetching details for club 47/126: #2J28UV28J Fetching details for club 48/126: #29VU2J0JJ Fetching details for club 49/126: #2GPJQU2QU Fetching details for club 50/126: #2LPYCVUGQ Fetching details for club 51/126: #2YY9UQ28G Fetching details for club 52/126: #JPL80URV Fetching details for club 53/126: #2RJ8JJPQP Fetching details for club 54/126: #2L9VJCYJ2 Fetching details for club 55/126: #2VCYY0P0R Fetching details for club 56/126: #2L8LVVG0C Fetching details for club 57/126: #802UR0J22 Fetching details for club 58/126: #2UG0JP0RP Fetching details for club 59/126: #2VQ2JJP09 Fetching details for club 60/126: #2LYLULYUJ Fetching details for club 61/126: #2LP8QQQGU Fetching details for club 62/126: #JY288RRC Fetching details for club 63/126: #RPJU0RP2 Fetching details for club 64/126: #809PYY9JG Fetching details for club 65/126: #2VC8CVV0G Fetching details for club 66/126: #P00PP2Y Fetching details for club

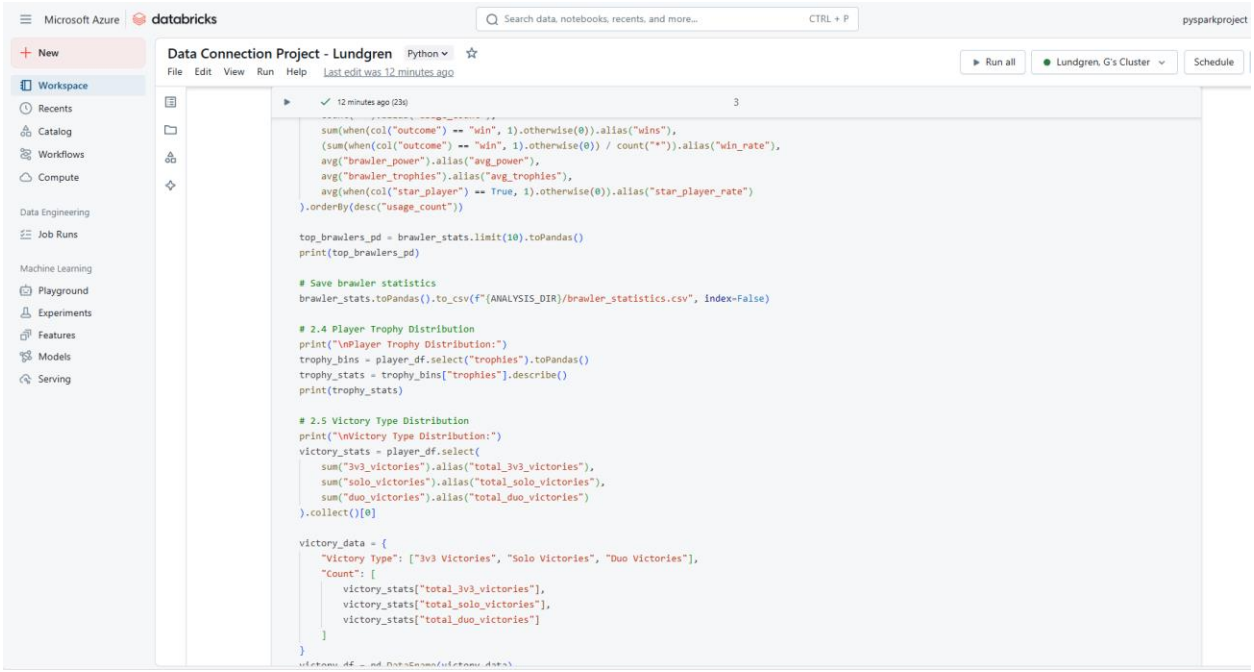
67/126: #80P00RULC Fetching details for club 68/126: #80P00R80Y Fetching details for club 69/126: #JYQ0UCUL Fetching details for club 70/126: #2U9LGRLGQ Fetching details for club 71/126: #2PL0QCLCJ Fetching details for club 72/126: #2VUQ8822Q Fetching details for club 73/126: #800RR2CG9 Fetching details for club 74/126: #2JUUJPCCP Fetching details for club 75/126: #2Q28PQ0P0 Fetching details for club 76/126: #80P002909 Fetching details for club 77/126: #2LPUU90UL Fetching details for club 78/126: #2VPPY0CPU Fetching details for club 79/126: #2YUGP8GVR Fetching details for club 80/126: #2JUP8QGPU Fetching details for club 81/126: #2L9QPPCCR Fetching details for club 82/126: #2L8VC88RR Fetching details for club 83/126: #2VU9QP80V Fetching details for club 84/126: #2L9YUGUY0 Fetching details for club 85/126: #2JQPPC0CQ Fetching details for club 86/126: #2UGGYRU0U Fetching details for club 87/126: #800LG2YVQ Fetching details for club 88/126: #2VCJ0YYJJ Fetching details for club 89/126: #2VJ9GRVGY Fetching details for club 90/126: #2VPPQ29J2 Fetching details for club 91/126: #80PJ8PJCP Fetching details for club 92/126: #2UJL9JQRR Fetching details for club 93/126: #80P8PU0Q9 Fetching details for club 94/126: #2RQQU2YLV Fetching details for club 95/126: #2LY8JVCUY Fetching details for club 96/126: #2G8V09JYJ Fetching details for club 97/126: #228U8YPVL Fetching details for club 98/126: #80P0YP229 Fetching details for club 99/126: #2L2CQL98U Fetching details for club 100/126: #2LPGCGL0R Fetching details for club 101/126: #2RPYGU8RQ Fetching details for club 102/126: #800U02CLV Fetching details for club 103/126: #2JJYJRCU0 Fetching details for club 104/126: #809U0GJLC Fetching details for club 105/126: #J2CP0Y8C Fetching details for club 106/126: #J9V2V9PU Fetching details for club 107/126: #28U982 Fetching details for club 108/126: #2VGPUYUGR Fetching details for club 109/126: #2L8YLJ0CC Fetching details for club 110/126: #2LLPYLC0J Fetching details for club 111/126: #8098JGP2J Fetching details for club 112/126: #2LPGPU8CV Fetching details for club 113/126: #JY8Y8LVJ Fetching details for club 114/126: #2VCQQYL08 Fetching details for club 115/126: #J8LVR02Y Fetching details for club 116/126: #2JUV8U9CL Fetching details for club 117/126: #802UULP8U Fetching details for club 118/126: #2RVLUQ8UR Fetching details for club 119/126: #20U9YJR0L Fetching details for club 120/126: #9QYJ9L22 Fetching details for club 121/126: #2JUQPVYRP Fetching details for club 122/126: #2JQGUQPGY Fetching details for club 123/126: #2UQJ0GJ88 Fetching details for club 124/126: #2VLG9YP98 Fetching details for club 125/126: #2JCR2GGGU Fetching details for club 126/126: #2L80JRPQV Saved club data to brawl_stars_data/club_data_20250407_025056.csv Collected data for 126 clubs Saved player-battle merged data to brawl_stars_data/player_battles_merged_20250407_025056.csv Saved player-club merged data to brawl_stars_data/player_clubs_merged_20250407_025056.csv Saved comprehensive merged data to brawl_stars_data/comprehensive_data_20250407_025056.csv Created merged datasets for analysis Saved summary statistics to brawl_stars_data/summary_stats_20250407_025056.json === DATASET SUMMARY === Total Players: 200 Total Battles: 4000 Total Clubs: 126 Average Player Trophies: 98399.1 Overall Win Rate: 76.9% Most Popular Mode: brawlBall Most Popular Brawler: BEA === Data Collection Complete === All data saved to brawl_stars_data directory Files are timestamped with: 20250407_025056 Ready for PySpark analysis!”

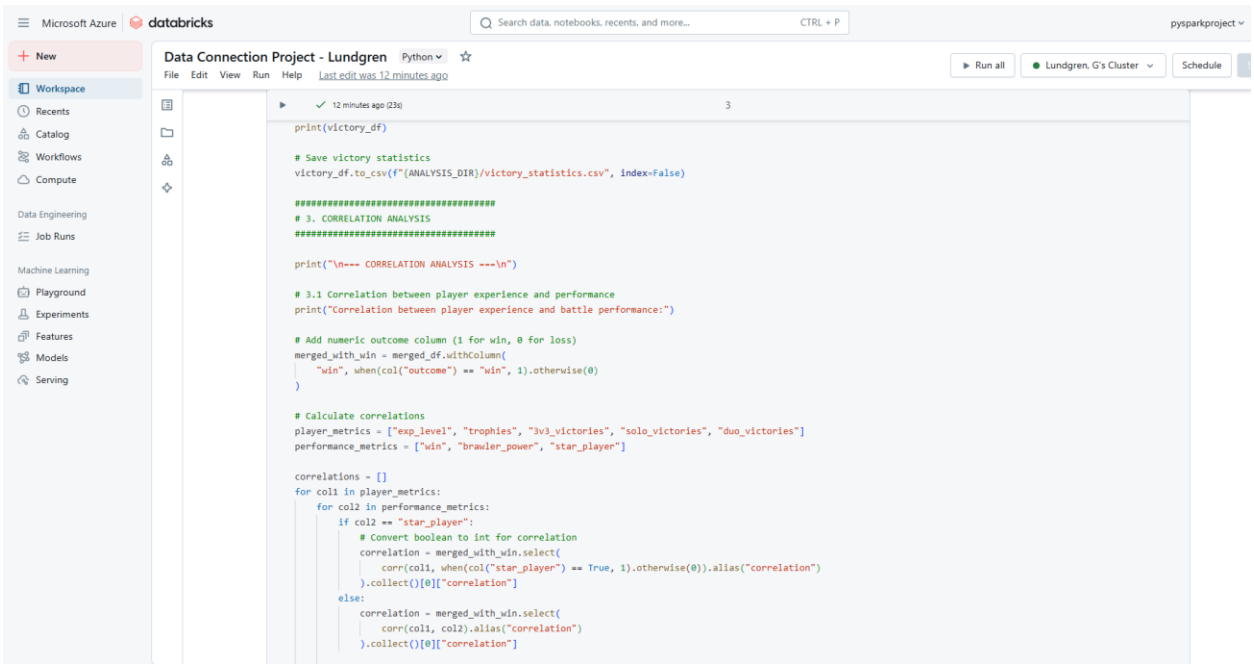

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
pysparkproj
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 10 minutes ago
Run all
Lundgren, G's Cluster
Ready for PySpark analysis!
21 minutes ago (5s)
2
import pandas as pd
# Read files with pandas
player_df_pd = pd.read_csv("brawl_stars_player_profiles.csv")
battles_df_pd = pd.read_csv("brawl_stars_battles.csv")
recent_battles_df_pd = pd.read_csv("brawl_stars_recent_battles.csv")
merged_df_pd = pd.read_csv("brawl_stars_merged_data.csv")
# Convert to Spark DataFrames if needed
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("BrawlStarsAnalysis").getOrCreate()
player_df = spark.createDataFrame(player_df_pd)
battles_df = spark.createDataFrame(battles_df_pd)
recent_battles_df = spark.createDataFrame(recent_battles_df_pd)
merged_df = spark.createDataFrame(merged_df_pd)
# Now you can use these DataFrames for analysis
print(f"Player data: {player_df.count()} rows")
print(f"Battle data: {battles_df.count()} rows")
print(f"Recent battle data: {recent_battles_df.count()} rows")
print(f"Merged data: {merged_df.count()} rows")
(8) Spark Jobs
battles_df: pyspark.sql.dataframe.DataFrame = [battle_id: string, player_tag: string ... 10 more fields]
merged_df: pyspark.sql.dataframe.DataFrame = [battle_id: string, player_tag: string ... 21 more fields]
player_df: pyspark.sql.dataframe.DataFrame = [player_tag: string, player_name: string ... 10 more fields]
recent_battles_df: pyspark.sql.dataframe.DataFrame = [timestamp: string, mode: string ... 6 more fields]
Player data: 200 rows
Battle data: 3864 rows
Recent battle data: 500 rows
Merged data: 3864 rows

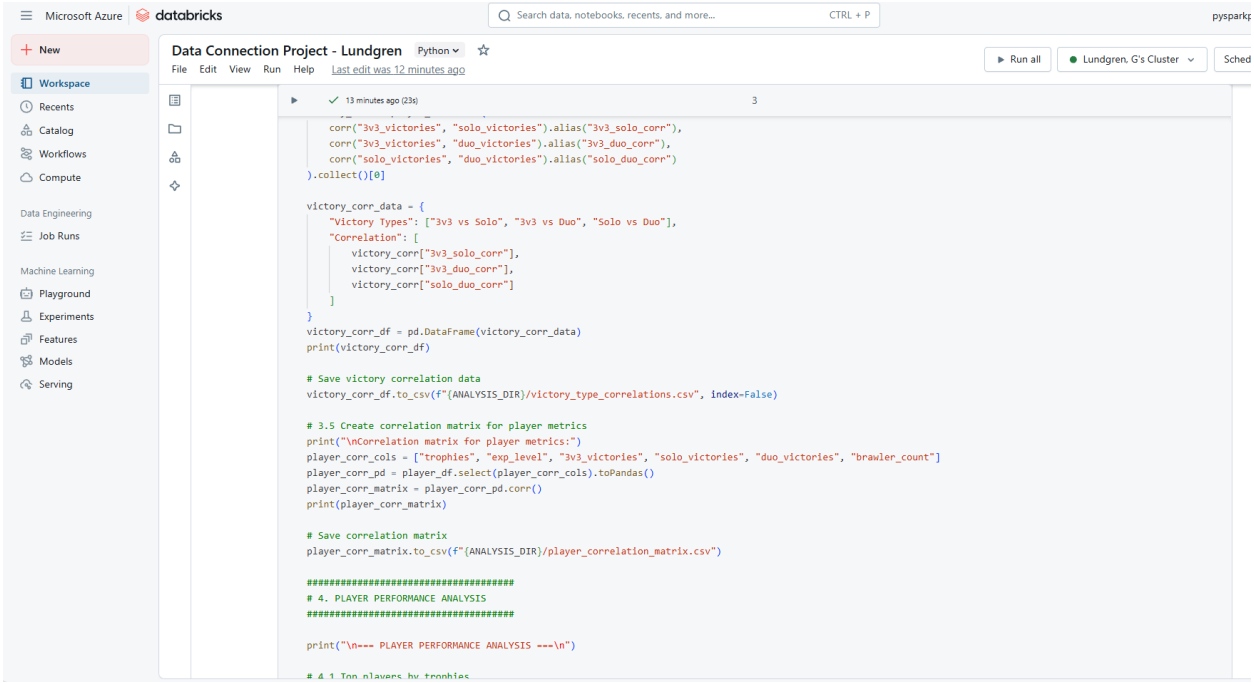

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
pysparkproject
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 11 minutes ago
Run all
Lundgren, G's Cluster
Schedule
Share
12 minutes ago (23s)
3
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, count, avg, min, max, stddev, sum, desc, asc, when, corr, expr, lit
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
import os
# Create a directory for analysis outputs
ANALYSIS_DIR = "./brawl_stars_analysis"
os.makedirs(ANALYSIS_DIR, exist_ok=True)
# Set plot style
plt.style.use('seaborn-v0_8-whitegrid')
sns.set_palette("viridis")
print("Starting Brawl Stars Data Analysis...")
#####################################
1. DATA LOADING
#####################################
# Load data using pandas (since files are in workspace)
print("\n=== LOADING DATA ===\n")
# Load the datasets
player_df_pd = pd.read_csv("brawl_stars_player_profiles.csv")
battles_df_pd = pd.read_csv("brawl_stars_battles.csv")
recent_battles_df_pd = pd.read_csv("brawl_stars_recent_battles.csv")
merged_df_pd = pd.read_csv("brawl_stars_merged_data.csv")
print(f"Loaded player data: {len(player_df_pd)} rows")
print(f"Loaded battle data: {len(battles_df_pd)} rows")

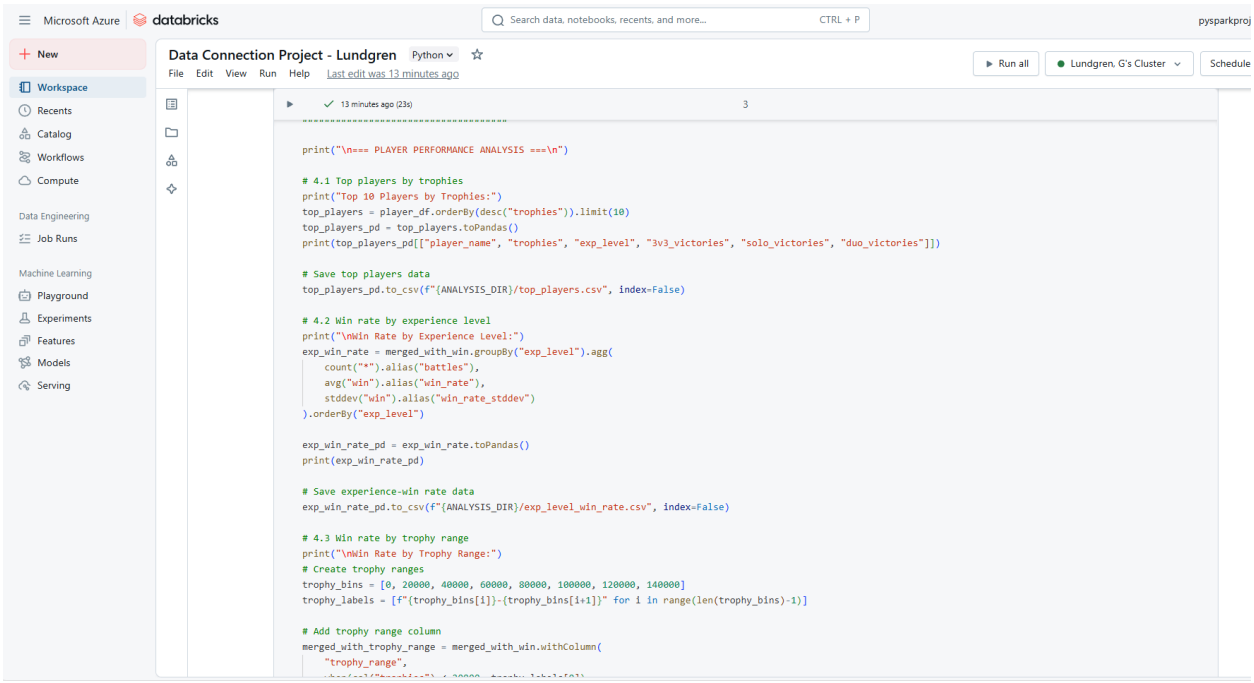

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
pysparkproject
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 11 minutes ago
Run all
Lundgren, G's Cluster
Schedule
12 minutes ago (23s)
3
print(f"Loaded merged data: {len(merged_df_pd)} rows")
# Initialize Spark Session
spark = SparkSession.builder.appName("BrawlStarsAnalysis").getOrCreate()
# Convert to Spark DataFrames
player_df = spark.createDataFrame(player_df_pd)
battle_df = spark.createDataFrame(battles_df_pd)
recent_battles_df = spark.createDataFrame(recent_battles_df_pd)
merged_df = spark.createDataFrame(merged_df_pd)
# Register DataFrames as temp views for SQL queries
player_df.createOrReplaceTempView("players")
battle_df.createOrReplaceTempView("battles")
recent_battles_df.createOrReplaceTempView("recent_battles")
merged_df.createOrReplaceTempView("merged")
# Display the schema of each DataFrame
print("\nPlayer DataFrame Schema:")
player_df.printSchema()
print("\nBattle DataFrame Schema:")
battle_df.printSchema()
print("\nMerged DataFrame Schema:")
merged_df.printSchema()
####################################
2. STATISTICAL SUMMARIES
####################################
print("\n=== STATISTICAL SUMMARIES ===\n")
# 2.1 Player Statistics
print("Player Statistics:")
player_stats = player_df.select(

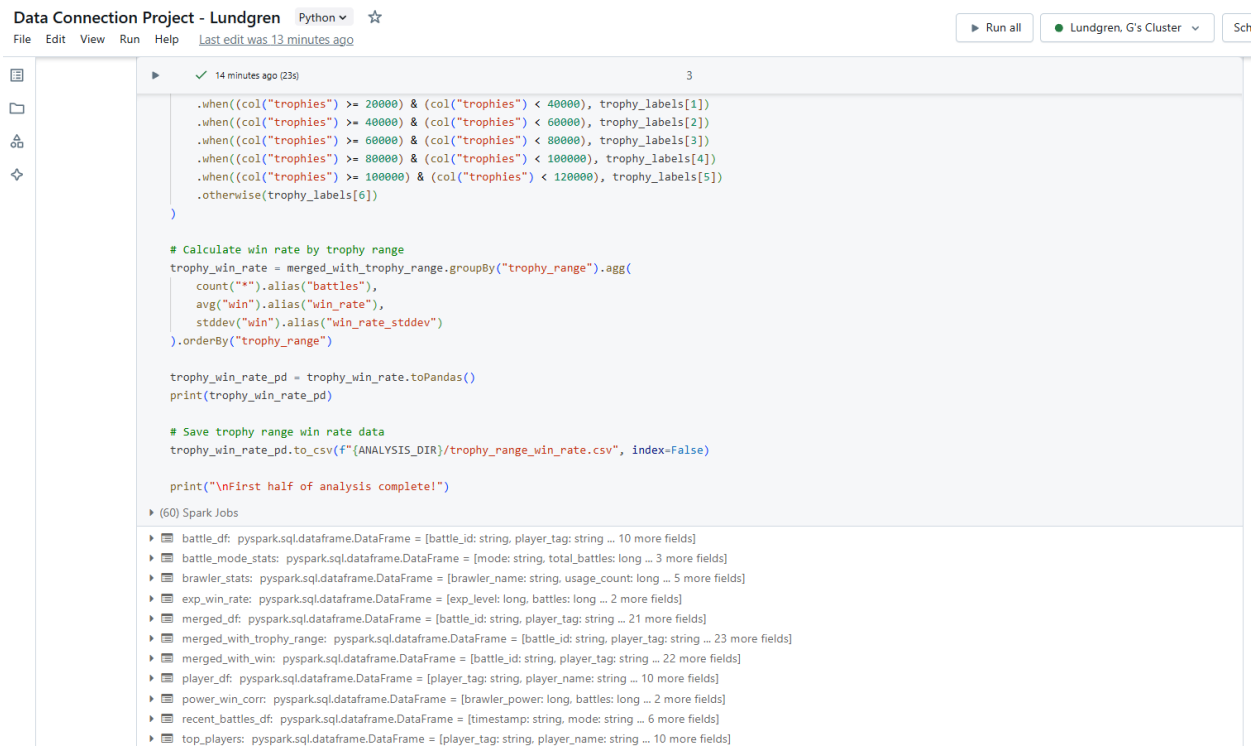

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 11 minutes ago
Run all
Lundgren, G's Cluster
12 minutes ago (23s)
3
avg("trophies").alias("avg_trophies"),
min("trophies").alias("min_trophies"),
max("trophies").alias("max_trophies"),
stddev("trophies").alias("stddev_trophies"),
avg("exp_level").alias("avg_exp_level"),
avg("3v3_victories").alias("avg_3v3_victories"),
avg("solo_victories").alias("avg_solo_victories"),
avg("duo_victories").alias("avg_duo_victories"),
avg("brawler_count").alias("avg_brawler_count")
).collect()[0]
# Convert to pandas for easier display
player_stats_pd = pd.DataFrame([player_stats.asDict()])
print(player_stats_pd.T) # Transpose for better display
# Save player statistics
player_stats_pd.to_csv(f"{ANALYSIS_DIR}/player_statistics.csv")
# 2.2 Battle Statistics by Mode
print("\nBattle Statistics by Game Mode:")
battle_mode_stats = battle_df.groupBy("mode").agg(
count("*").alias("total_battles"),
sum(when(col("outcome") == "win", 1).otherwise(0)).alias("wins"),
(sum(when(col("outcome") == "win", 1).otherwise(0)) / count("*")).alias("win_rate"),
avg("duration").alias("avg_duration")
).orderBy(desc("total_battles"))
battle_mode_stats_pd = battle_mode_stats.toPandas()
print(battle_mode_stats_pd)
# Save battle mode statistics
battle_mode_stats_pd.to_csv(f"{ANALYSIS_DIR}/battle_mode_statistics.csv", index=False)
# 2.3 Brawler Performance Statistics
print("\nTop 10 Brawlers by Usage:")
brawler_stats = battle_df.groupBy("brawler_name").agg(


Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
pysparkproject
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 12 minutes ago
Run all
Lundgren, G's Cluster
Schedule
12 minutes ago (23s)
3
sum(when(col("outcome") == "win", 1).otherwise(0)).alias("wins"),
(sum(when(col("outcome") == "win", 1).otherwise(0)) / count("*")).alias("win_rate"),
avg("brawler_power").alias("avg_power"),
avg("brawler_trophies").alias("avg_trophies"),
avg(when(col("star_player") == True, 1).otherwise(0)).alias("star_player_rate")
).orderBy(desc("usage_count"))
top_brawlers_pd = brawler_stats.limit(10).toPandas()
print(top_brawlers_pd)
# Save brawler statistics
brawler_stats.toPandas().to_csv(f"{ANALYSIS_DIR}/brawler_statistics.csv", index=False)
# 2.4 Player Trophy Distribution
print("\nPlayer Trophy Distribution:")
trophy_bins = player_df.select("trophies").toPandas()
trophy_stats = trophy_bins["trophies"].describe()
print(trophy_stats)
# 2.5 Victory Type Distribution
print("\nVictory Type Distribution:")
victory_stats = player_df.select(
sum("3v3_victories").alias("total_3v3_victories"),
sum("solo_victories").alias("total_solo_victories"),
sum("duo_victories").alias("total_duo_victories")
).collect()[0]
victory_data = {
"Victory Type": ["3v3 Victories", "Solo Victories", "Duo Victories"],
"Count": [
victory_stats["total_3v3_victories"],
victory_stats["total_solo_victories"],
victory_stats["total_duo_victories"]
]
}


Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
pysparkproject
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 12 minutes ago
Run all
Lundgren, G's Cluster
Schedule
12 minutes ago (23s)
3
print(victory_df)
# Save victory statistics
victory_df.to_csv(f"{ANALYSIS_DIR}/victory_statistics.csv", index=False)
#####################################
3. CORRELATION ANALYSIS
#####################################
print("\n=== CORRELATION ANALYSIS ===\n")
# 3.1 Correlation between player experience and performance
print("Correlation between player experience and battle performance:")
# Add numeric outcome column (1 for win, 0 for loss)
merged_with_win = merged_df.withColumn(
"win", when(col("outcome") == "win", 1).otherwise(0)
)
# Calculate correlations
player_metrics = ["exp_level", "trophies", "3v3_victories", "solo_victories", "duo_victories"]
performance_metrics = ["win", "brawler_power", "star_player"]
correlations = []
for col1 in player_metrics:
for col2 in performance_metrics:
if col2 == "star_player":
# Convert boolean to int for correlation
correlation = merged_with_win.select(
corr(col1, when(col("star_player") == True, 1).otherwise(0)).alias("correlation")
).collect()[0]["correlation"]
else:
correlation = merged_with_win.select(
corr(col1, col2).alias("correlation")
).collect()[0]["correlation"]

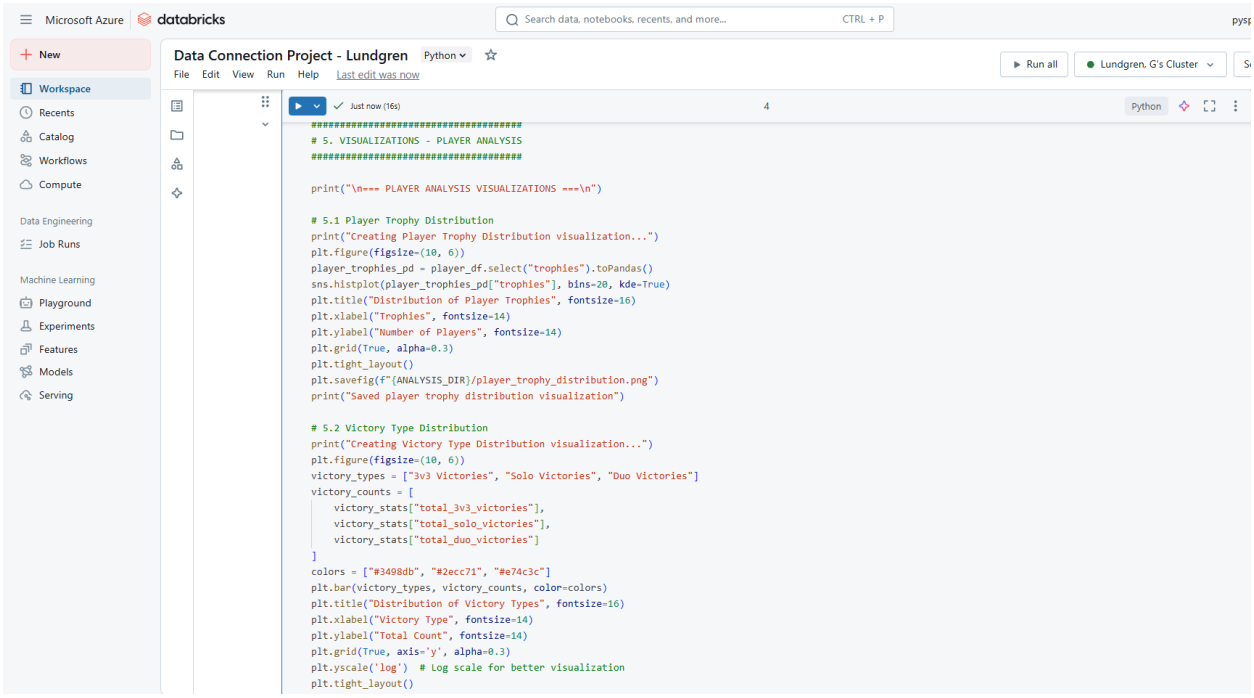

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 12 minutes ago
Run all
Lundgren, G's Cluster
13 minutes ago (23s)
corr("3v3_victories", "solo_victories").alias("3v3_solo_corr"),
corr("3v3_victories", "duo_victories").alias("3v3_duo_corr"),
corr("solo_victories", "duo_victories").alias("solo_duo_corr")
).collect()[0]
victory_corr_data = {
"Victory Types": ["3v3 vs Solo", "3v3 vs Duo", "Solo vs Duo"],
"Correlation": [
victory_corr["3v3_solo_corr"],
victory_corr["3v3_duo_corr"],
victory_corr["solo_duo_corr"]
]
}
victory_corr_df = pd.DataFrame(victory_corr_data)
print(victory_corr_df)
# Save victory correlation data
victory_corr_df.to_csv(f"{ANALYSIS_DIR}/victory_type_correlations.csv", index=False)
# 3.5 Create correlation matrix for player metrics
print("\nCorrelation matrix for player metrics:")
player_corr_cols = ["trophies", "exp_level", "3v3_victories", "solo_victories", "duo_victories", "brawler_count"]
player_corr_pd = player_df.select(player_corr_cols).toPandas()
player_corr_matrix = player_corr_pd.corr()
print(player_corr_matrix)
# Save correlation matrix
player_corr_matrix.to_csv(f"{ANALYSIS_DIR}/player_correlation_matrix.csv")
#####################################
4. PLAYER PERFORMANCE ANALYSIS
#####################################
print("\n=== PLAYER PERFORMANCE ANALYSIS ===\n")

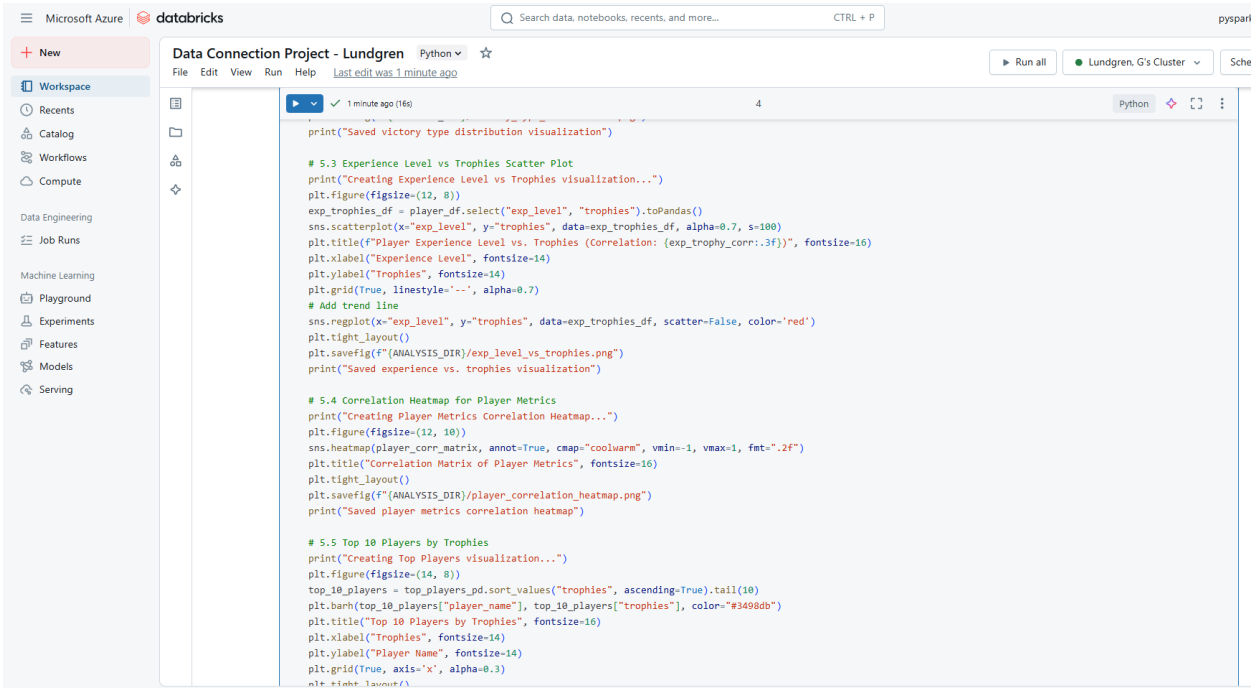

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 13 minutes ago
Run all
Lundgren, G's Cluster
13 minutes ago (23s)
print("\n=== PLAYER PERFORMANCE ANALYSIS ===\n")
# 4.1 Top players by trophies
print("Top 10 Players by Trophies:")
top_players = player_df.orderBy(desc("trophies")).limit(10)
top_players_pd = top_players.toPandas()
print(top_players_pd[["player_name", "trophies", "exp_level", "3v3_victories", "solo_victories", "duo_victories"]])
# Save top players data
top_players_pd.to_csv(f"{ANALYSIS_DIR}/top_players.csv", index=False)
# 4.2 Win rate by experience level
print("\nWin Rate by Experience Level:")
exp_win_rate = merged_with_win.groupBy("exp_level").agg(
count("*").alias("battles"),
avg("win").alias("win_rate"),
stddev("win").alias("win_rate_stddev")
).orderBy("exp_level")
exp_win_rate_pd = exp_win_rate.toPandas()
print(exp_win_rate_pd)
# Save experience-win rate data
exp_win_rate_pd.to_csv(f"{ANALYSIS_DIR}/exp_level_win_rate.csv", index=False)
# 4.3 Win rate by trophy range
print("\nWin Rate by Trophy Range:")
# Create trophy ranges
trophy_bins = [0, 20000, 40000, 60000, 80000, 100000, 120000, 140000]
trophy_labels = [f"{trophy_bins[i]}-{trophy_bins[i+1]}" for i in range(len(trophy_bins)-1)]
# Add trophy range column
merged_with_trophy_range = merged_with_win.withColumn(
"trophy_range",

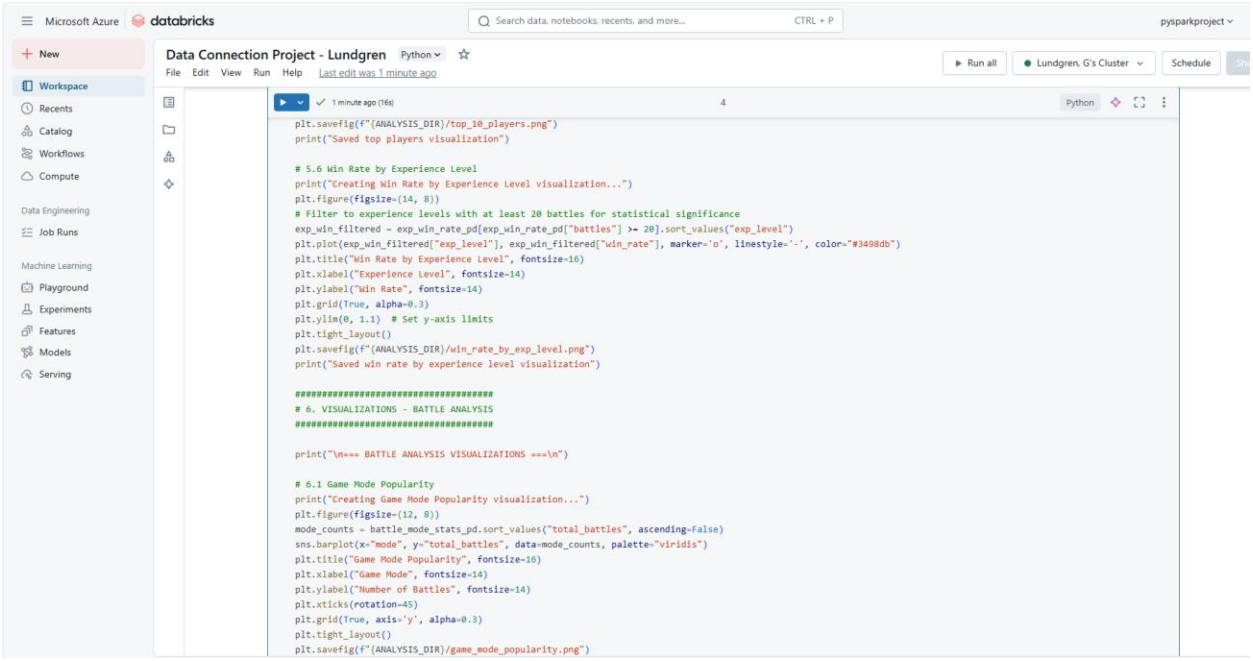


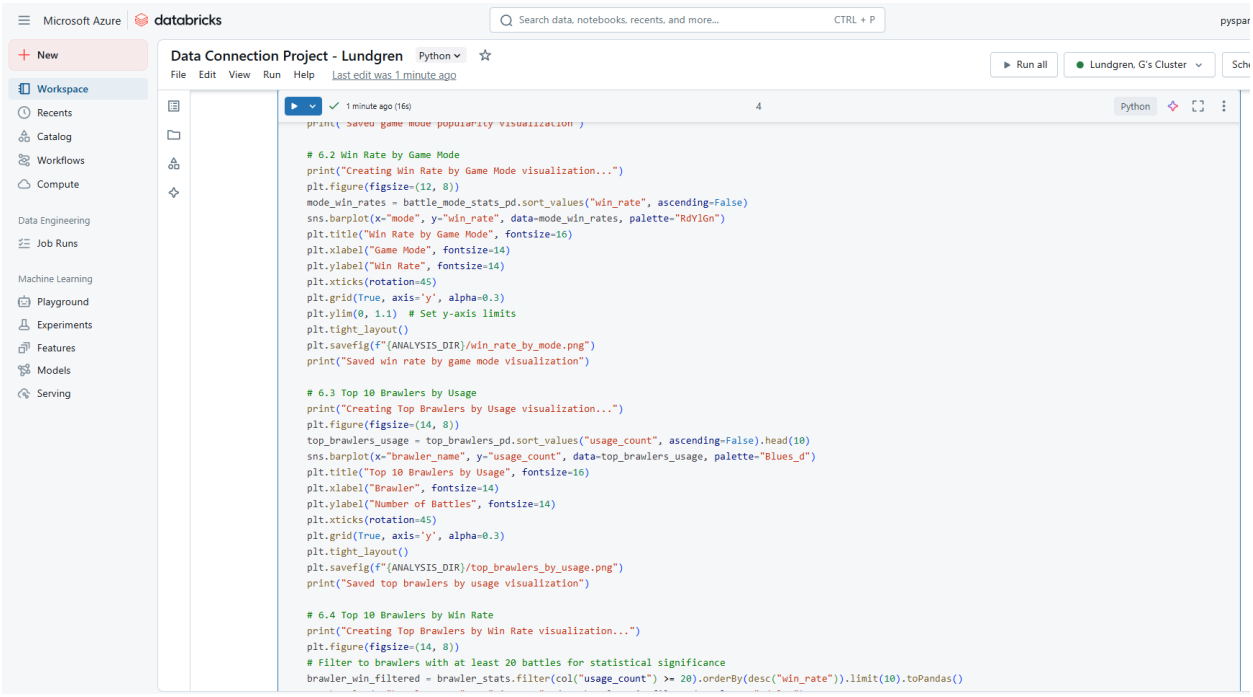
Copied output:” Starting Brawl Stars Data Analysis... === LOADING DATA === Loaded player data: 200 rows Loaded battle data: 3864 rows Loaded recent battle data: 500 rows Loaded merged data: 3864 rows Player DataFrame Schema: root |-- player_tag: string (nullable = true) |-- player_name: string (nullable = true) |-- trophies: long (nullable = true) |-- highest_trophies: long (nullable = true) |-- exp_level: long (nullable = true) |-- 3v3_victories: long (nullable = true) |-- solo_victories: long (nullable = true) |-- duo_victories: long (nullable = true) |-- club_tag: string (nullable = true) |-- club_name: string (nullable = true) |-- brawler_count: long (nullable = true) |-- avg_brawler_trophies: double (nullable = true) Battle DataFrame Schema: root |-- battle_id: string (nullable = true) |-- player_tag: string (nullable = true) |-- timestamp: string (nullable = true) |-- mode: string (nullable = true) |-- map: string (nullable = true) |-- type: string (nullable = true) |-- brawler_name: string (nullable = true) |-- brawler_power: long (nullable = true) |-- brawler_trophies: long (nullable = true) |-- outcome: string (nullable = true) |-- duration: double (nullable = true) |-- star_player: boolean (nullable = true) Merged DataFrame Schema: root |-- battle_id: string (nullable = true) |-- player_tag: string (nullable = true) |-- timestamp: string (nullable = true) |-- mode: string (nullable = true) |-- map: string (nullable = true) |-- type: string (nullable = true) |-- brawler_name: string (nullable = true) |-- brawler_power: long (nullable = true) |-- brawler_trophies: long (nullable = true) |-- outcome: string (nullable = true) |-- duration: double (nullable = true) |-- star_player: boolean (nullable = true) |-- player_name: string (nullable = true) |-- trophies: long (nullable = true) |-- highest_trophies: long (nullable = true) |-- exp_level: long (nullable = true) |-- 3v3_victories: long (nullable = true) |-- solo_victories: long (nullable = true) |-- duo_victories: long (nullable = true) |-- club_tag: string (nullable = true) |-- club_name: string (nullable = true) |-- brawler_count: long (nullable = true) |-- avg_brawler_trophies: double (nullable = true) === STATISTICAL SUMMARIES === Player Statistics: 0 avg_trophies 98128.255000 min_trophies 95657.000000

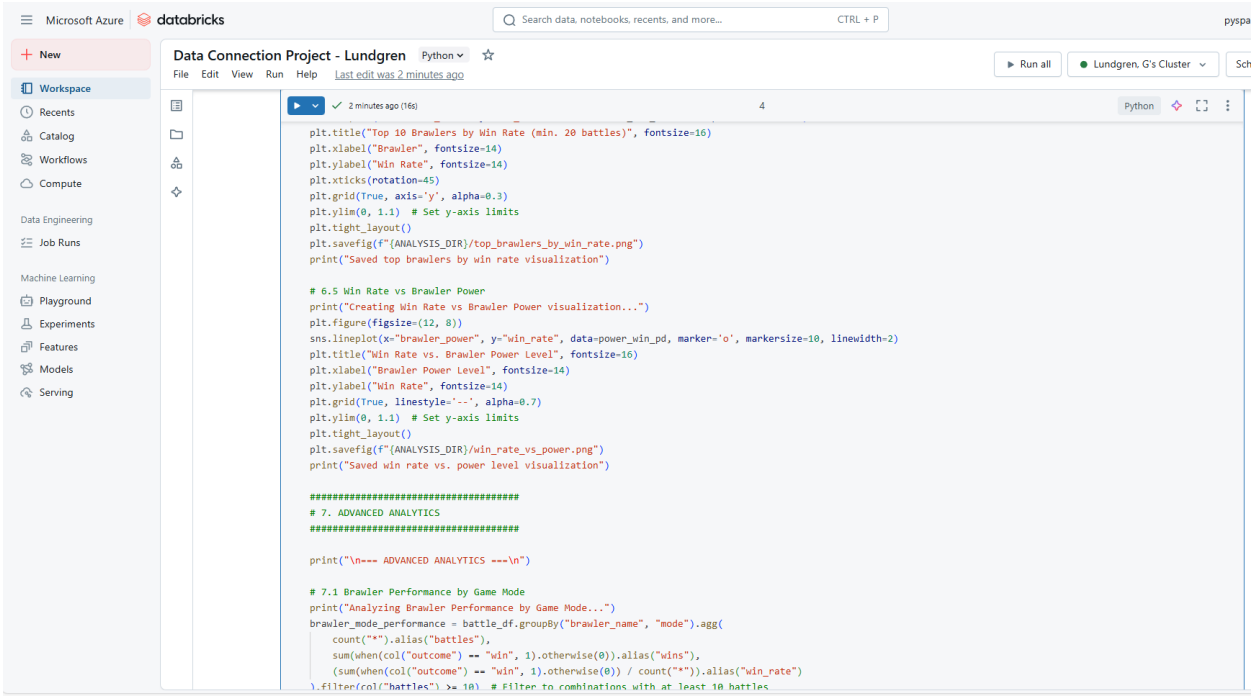
max_trophies 118643.000000 stddev_trophies 2881.581273 avg_exp_level 339.780000 avg_3v3_victories 33763.510000 avg_solo_victories 3485.950000 avg_duo_victories 4604.160000 avg_brawler_count 89.945000 Battle Statistics by Game Mode: mode total_battles wins win_rate avg_duration 0 brawlBall 2560 2090 0.816406 84.060156 1 knockout 347 262 0.755043 105.291066 2 duoShowdown 208 0 0.000000 NaN 3 hotZone 182 142 0.780220 101.565934 4 siege 179 145 0.810056 85.307263 5 bounty 133 121 0.909774 96.150376 6 wipeout 115 100 0.869565 121.747826 7 gemGrab 105 86 0.819048 103.847619 8 heist 35 29 0.828571 60.971429 Top 10 Brawlers by Usage: brawler_name usage_count wins ... avg_power avg_trophies star_player_rate 0 BEA 145 120 ... 10.572414 1083.462069 0.289655 1 LOLA 111 79 ... 10.801802 1177.423423 0.216216 2 MAX 103 75 ... 10.883495 1216.252427 0.145631 3 AMBER 102 75 ... 10.911765 1083.509804 0.254902 4 MANDY 99 75 ... 10.878788 1158.363636 0.212121 5 TARA 98 86 ... 10.795918 1117.091837 0.306122 6 PIPER 97 60 ... 10.628866 1061.855670 0.278351 7 NANI 96 81 ... 10.947917 1111.239583 0.395833 8 MAISIE 92 73 ... 11.000000 1151.673913 0.239130 9 STU 91 78 ... 10.868132 1255.142857 0.230769 [10 rows x 7 columns] Player Trophy Distribution: count 200.000000 mean 98128.255000 std 2881.581273 min 95657.000000 25% 96319.500000 50% 97167.000000 75% 98860.750000 max 118643.000000 Name: trophies, dtype: float64 Victory Type Distribution: Victory Type Count 0 3v3 Victories 6752702 1 Solo Victories 697190 2 Duo Victories 920832 === CORRELATION ANALYSIS === Correlation between player experience and battle performance: player_attribute performance_metric correlation 0 exp_level win 0.081727 1 exp_level brawler_power 0.058628 2 exp_level star_player 0.040156 3 trophies win 0.102992 4 trophies brawler_power -0.044690 5 trophies star_player 0.048095 6 3v3_victories win 0.156622 7 3v3_victories brawler_power 0.028047 8 3v3_victories star_player 0.064362 9 solo_victories win -0.069086 10 solo_victories brawler_power -0.055103 11 solo_victories star_player -0.020906 12 duo_victories win -0.091441 13 duo_victories brawler_power -0.001578 14 duo_victories star_player -0.037741 Correlation between brawler power and win rate: brawler_power battles wins win_rate 0 -1 16 12 0.750000 1 6 1 1 1.000000 2 9 107 67 0.626168 3 10 382 239 0.625654 4 11 3358 2656 0.790947 Correlation between experience level and trophies: Correlation coefficient: 0.1929 Correlations between different victory types: Victory Types Correlation 0 3v3 vs Solo -0.112228 1 3v3 vs Duo -0.073113 2 Solo vs Duo 0.344894 Correlation matrix for player metrics: trophies exp_level ... duo_victories brawler_count trophies 1.000000 0.192857 ... -0.106971 0.073873 exp_level 0.192857 1.000000 ... 0.361772 -0.111473 3v3_victories 0.444466 0.774828 ... -0.073113 -0.070948 solo_victories -0.001602 0.287182 ... 0.344894 -0.104715 duo_victories -0.106971 0.361772 ... 1.000000 -0.061546 brawler_count 0.073873 -0.111473 ... -0.061546 1.000000 [6 rows x 6 columns] === PLAYER PERFORMANCE ANALYSIS === Top 10 Players by Trophies: player_name trophies ... solo_victories duo_victories 0 Hyra 118643 ... 6501 2159 1 ELV|JuanCarlos 110036 ... 7887 3332 2 YT: DetroBS 107568 ... 274 846 3 Maxel🫧 106158 ... 1621 2341 4 Elox 106025 ... 2629 1487 5 YT Cry Spirit 105637 ... 1512 1831 6 makmay_ 105426 ... 1986 1145 7 ItzFire 105129 ... 3122 3472 8 Efecan✨~Shelly🖤 104402 ... 5198 5253 9 Krozyy♡ 104077 ... 8790 6204 [10 rows x 6 columns] Win Rate by Experience Level: exp_level battles win_rate win_rate_stddev 0 214 20 0.050 0.223607 1 218 20 0.750 0.444262 2 224 40 0.800 0.405096 3 225 20 0.700 0.470162 4 226 20 0.850 0.366348 .. .. 131 463 40 0.825 0.384808 132 472 15 1.000 0.000000 133 476 20 0.950 0.223607 134 480 20 0.950 0.223607 135 500 40 0.850 0.361620 [136 rows x 4 columns] Win Rate by Trophy Range: trophy_range battles win_rate win_rate_stddev 0 100000-120000 673 0.833581 0.372733 1 80000-100000 3191 0.756503 0.429260 First half of analysis complete!”

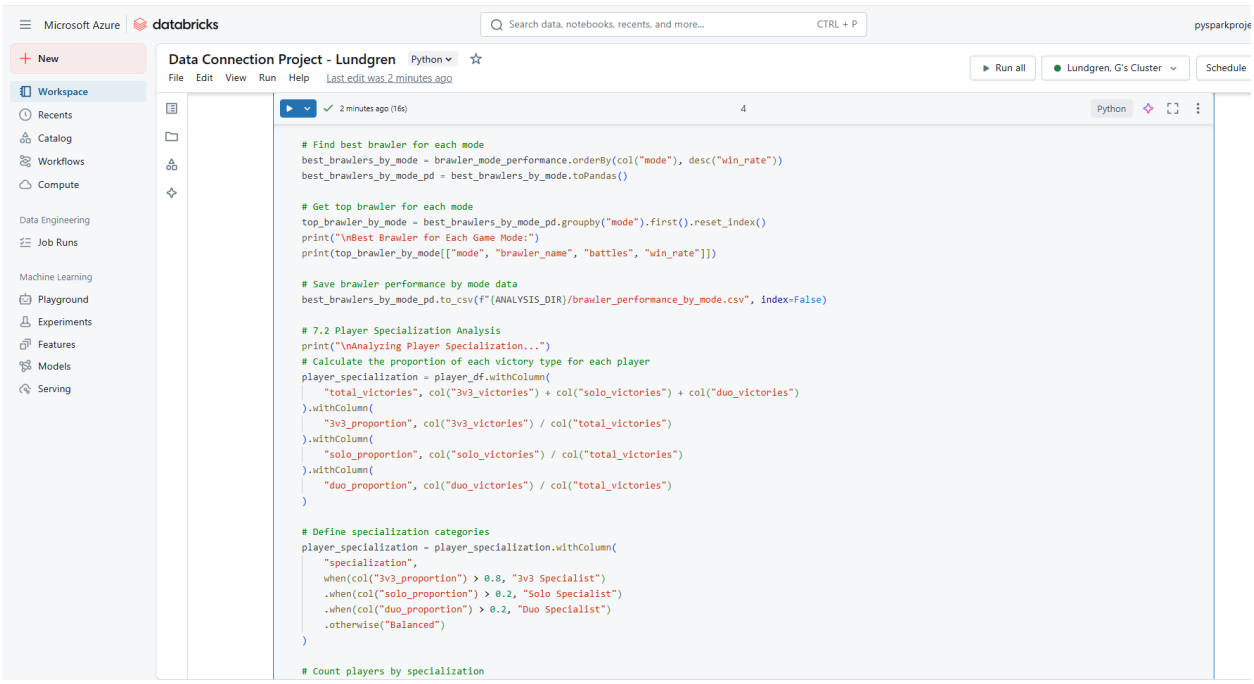

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was now
Run all
Lundgren, G's Cluster
Just now (16s)
4
Python
# 5. VISUALIZATIONS - PLAYER ANALYSIS
print("\n=== PLAYER ANALYSIS VISUALIZATIONS ===\n")
# 5.1 Player Trophy Distribution
print("Creating Player Trophy Distribution visualization...")
plt.figure(figsize=(10, 6))
player_trophies_pd = player_df.select("trophies").toPandas()
sns.histplot(player_trophies_pd["trophies"], bins=20, kde=True)
plt.title("Distribution of Player Trophies", fontsize=16)
plt.xlabel("Trophies", fontsize=14)
plt.ylabel("Number of Players", fontsize=14)
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.savefig(f"{ANALYSIS_DIR}/player_trophy_distribution.png")
print("Saved player trophy distribution visualization")
# 5.2 Victory Type Distribution
print("Creating Victory Type Distribution visualization...")
plt.figure(figsize=(10, 6))
victory_types = ["3v3 Victories", "Solo Victories", "Duo Victories"]
victory_counts = [
victory_stats["total_3v3_victories"],
victory_stats["total_solo_victories"],
victory_stats["total_duo_victories"]
]
colors = ["#3498db", "#2ecc71", "#e74c3c"]
plt.bar(victory_types, victory_counts, color=colors)
plt.title("Distribution of Victory Types", fontsize=16)
plt.xlabel("Victory Type", fontsize=14)
plt.ylabel("Total Count", fontsize=14)
plt.grid(True, axis='y', alpha=0.3)
plt.yscale('log') # Log scale for better visualization
plt.tight_layout()

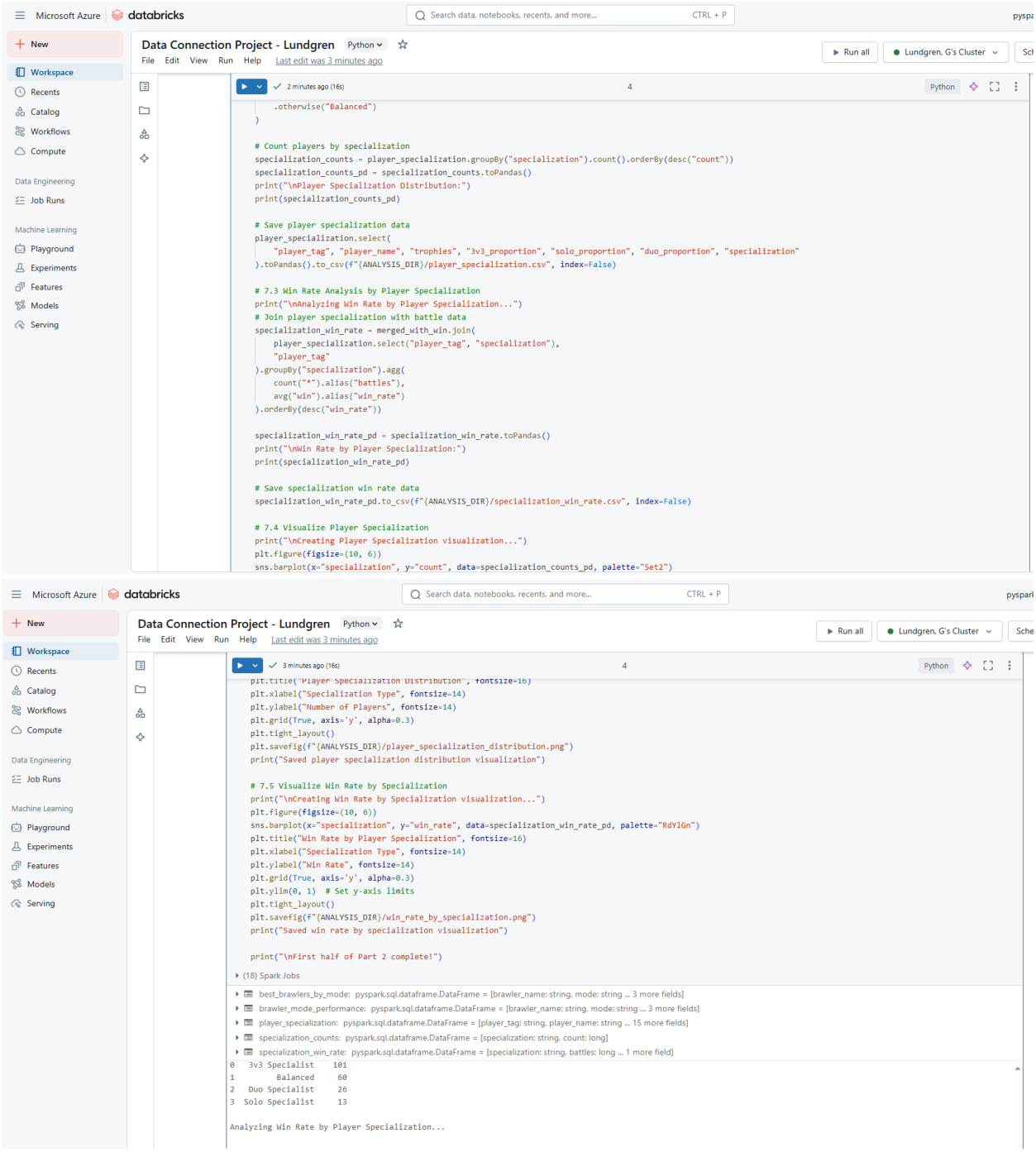

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 1 minute ago
Run all
Lundgren, G's Cluster
1 minute ago (16s)
4
Python
print("Saved victory type distribution visualization")
# 5.3 Experience Level vs Trophies Scatter Plot
print("Creating Experience Level vs Trophies visualization...")
plt.figure(figsize=(12, 8))
exp_trophies_df = player_df.select("exp_level", "trophies").toPandas()
sns.scatterplot(x="exp_level", y="trophies", data=exp_trophies_df, alpha=0.7, s=100)
plt.title(f"Player Experience Level vs. Trophies (Correlation: {exp_trophy_corr:.3f})", fontsize=16)
plt.xlabel("Experience Level", fontsize=14)
plt.ylabel("Trophies", fontsize=14)
plt.grid(True, linestyle='--', alpha=0.7)
# Add trend line
sns.regplot(x="exp_level", y="trophies", data=exp_trophies_df, scatter=False, color='red')
plt.tight_layout()
plt.savefig(f"{ANALYSIS_DIR}/exp_level_vs_trophies.png")
print("Saved experience vs. trophies visualization")
# 5.4 Correlation Heatmap for Player Metrics
print("Creating Player Metrics Correlation Heatmap...")
plt.figure(figsize=(12, 10))
sns.heatmap(player_corr_matrix, annot=True, cmap="coolwarm", vmin=-1, vmax=1, fmt=".2f")
plt.title("Correlation Matrix of Player Metrics", fontsize=16)
plt.tight_layout()
plt.savefig(f"{ANALYSIS_DIR}/player_correlation_heatmap.png")
print("Saved player metrics correlation heatmap")
# 5.5 Top 10 Players by Trophies
print("Creating Top Players visualization...")
plt.figure(figsize=(14, 8))
top_10_players = top_players_pd.sort_values("trophies", ascending=True).tail(10)
plt.barh(top_10_players["player_name"], top_10_players["trophies"], color="#3498db")
plt.title("Top 10 Players by Trophies", fontsize=16)
plt.xlabel("Trophies", fontsize=14)
plt.ylabel("Player Name", fontsize=14)
plt.grid(True, axis='x', alpha=0.3)


Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
pysparkproject
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 1 minute ago
Run all
Lundgren, G's Cluster
Schedule
1 minute ago (16s)
4
Python
plt.savefig(f"{ANALYSIS_DIR}/top_10_players.png")
print("Saved top players visualization")
# 5.6 Win Rate by Experience Level
print("Creating Win Rate by Experience Level visualization...")
plt.figure(figsize=(14, 8))
# Filter to experience levels with at least 20 battles for statistical significance
exp_win_filtered = exp_win_rate_pd[exp_win_rate_pd["battles"] >= 20].sort_values("exp_level")
plt.plot(exp_win_filtered["exp_level"], exp_win_filtered["win_rate"], marker='o', linestyle='-', color="#3498db")
plt.title("Win Rate by Experience Level", fontsize=16)
plt.xlabel("Experience Level", fontsize=14)
plt.ylabel("Win Rate", fontsize=14)
plt.grid(True, alpha=0.3)
plt.ylim(0, 1.1) # Set y-axis limits
plt.tight_layout()
plt.savefig(f"{ANALYSIS_DIR}/win_rate_by_exp_level.png")
print("Saved win rate by experience level visualization")
####################################
6. VISUALIZATIONS - BATTLE ANALYSIS
####################################
print("\n=== BATTLE ANALYSIS VISUALIZATIONS ===\n")
# 6.1 Game Mode Popularity
print("Creating Game Mode Popularity visualization...")
plt.figure(figsize=(12, 8))
mode_counts = battle_mode_stats_pd.sort_values("total_battles", ascending=False)
sns.barplot(x="mode", y="total_battles", data=mode_counts, palette="viridis")
plt.title("Game Mode Popularity", fontsize=16)
plt.xlabel("Game Mode", fontsize=14)
plt.ylabel("Number of Battles", fontsize=14)
plt.xticks(rotation=45)
plt.grid(True, axis='y', alpha=0.3)
plt.tight_layout()
plt.savefig(f"{ANALYSIS_DIR}/game_mode_popularity.png")


Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 1 minute ago
Run all
Lundgren, G's Cluster
1 minute ago (16s)
4
Python
# 6.2 Win Rate by Game Mode
print("Creating Win Rate by Game Mode visualization...")
plt.figure(figsize=(12, 8))
mode_win_rates = battle_mode_stats_pd.sort_values("win_rate", ascending=False)
sns.barplot(x="mode", y="win_rate", data=mode_win_rates, palette="RdYlGn")
plt.title("Win Rate by Game Mode", fontsize=16)
plt.xlabel("Game Mode", fontsize=14)
plt.ylabel("Win Rate", fontsize=14)
plt.xticks(rotation=45)
plt.grid(True, axis='y', alpha=0.3)
plt.ylim(0, 1.1) # Set y-axis limits
plt.tight_layout()
plt.savefig(f"{ANALYSIS_DIR}/win_rate_by_mode.png")
print("Saved win rate by game mode visualization")
# 6.3 Top 10 Brawlers by Usage
print("Creating Top Brawlers by Usage visualization...")
plt.figure(figsize=(14, 8))
top_brawlers_usage = top_brawlers_pd.sort_values("usage_count", ascending=False).head(10)
sns.barplot(x="brawler_name", y="usage_count", data=top_brawlers_usage, palette="Blues_d")
plt.title("Top 10 Brawlers by Usage", fontsize=16)
plt.xlabel("Brawler", fontsize=14)
plt.ylabel("Number of Battles", fontsize=14)
plt.xticks(rotation=45)
plt.grid(True, axis='y', alpha=0.3)
plt.tight_layout()
plt.savefig(f"{ANALYSIS_DIR}/top_brawlers_by_usage.png")
print("Saved top brawlers by usage visualization")
# 6.4 Top 10 Brawlers by Win Rate
print("Creating Top Brawlers by Win Rate visualization...")
plt.figure(figsize=(14, 8))
# Filter to brawlers with at least 20 battles for statistical significance
brawler_win_filtered = brawler_stats.filter(col("usage_count") >= 20).orderBy(desc("win_rate")).limit(10).toPandas()


Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 2 minutes ago
Run all
Lundgren, G's Cluster
2 minutes ago (16s)
4
Python
plt.title("Top 10 Brawlers by Win Rate (min. 20 battles)", fontsize=16)
plt.xlabel("Brawler", fontsize=14)
plt.ylabel("Win Rate", fontsize=14)
plt.xticks(rotation=45)
plt.grid(True, axis='y', alpha=0.3)
plt.ylim(0, 1.1) # Set y-axis limits
plt.tight_layout()
plt.savefig(f"{ANALYSIS_DIR}/top_brawlers_by_win_rate.png")
print("Saved top brawlers by win rate visualization")
# 6.5 Win Rate vs Brawler Power
print("Creating Win Rate vs Brawler Power visualization...")
plt.figure(figsize=(12, 8))
sns.lineplot(x="brawler_power", y="win_rate", data=power_win_pd, marker='o', markersize=10, linewidth=2)
plt.title("Win Rate vs. Brawler Power Level", fontsize=16)
plt.xlabel("Brawler Power Level", fontsize=14)
plt.ylabel("Win Rate", fontsize=14)
plt.grid(True, linestyle='--', alpha=0.7)
plt.ylim(0, 1.1) # Set y-axis limits
plt.tight_layout()
plt.savefig(f"{ANALYSIS_DIR}/win_rate_vs_power.png")
print("Saved win rate vs. power level visualization")
####################################
7. ADVANCED ANALYTICS
####################################
print("\n=== ADVANCED ANALYTICS ===\n")
# 7.1 Brawler Performance by Game Mode
print("Analyzing Brawler Performance by Game Mode...")
brawler_mode_performance = battle_df.groupBy("brawler_name", "mode").agg(
count("*").alias("battles"),
sum(when(col("outcome") == "win", 1).otherwise(0)).alias("wins"),
(sum(when(col("outcome") == "win", 1).otherwise(0)) / count("*")).alias("win_rate")
).filter(col("battles") >= 10) # Filter to combinations with at least 10 battles

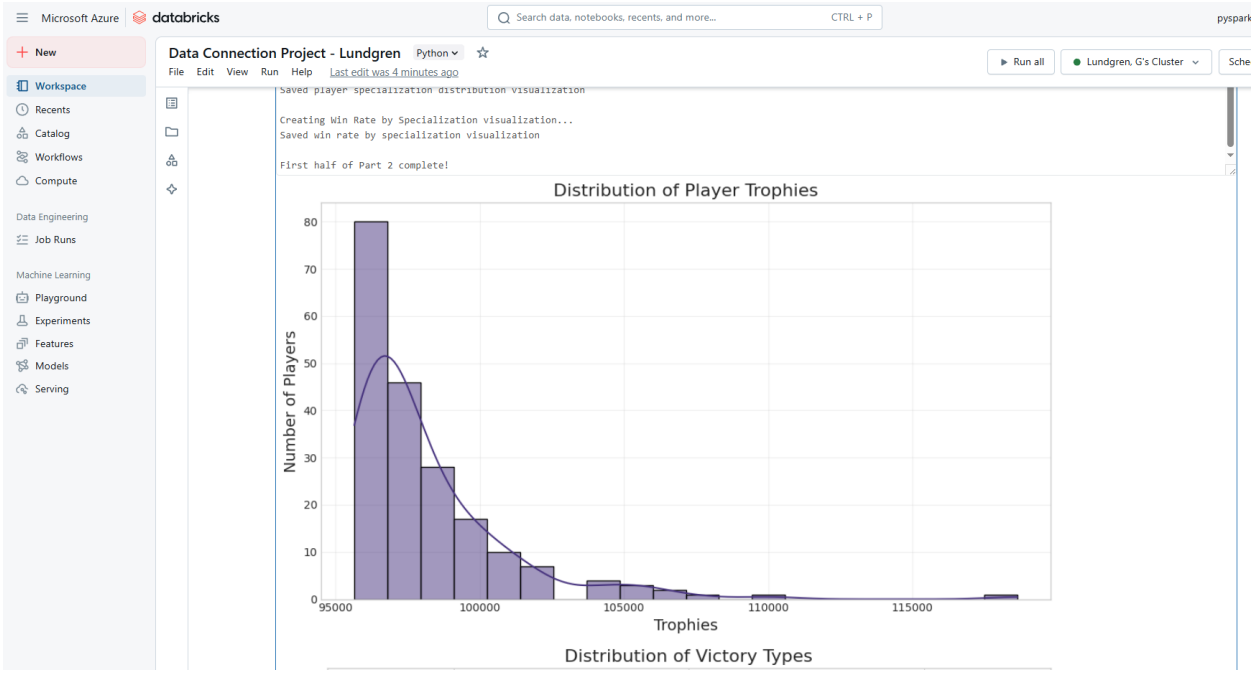

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 2 minutes ago
Run all
Lundgren, G's Cluster
Schedule
2 minutes ago (16s)
4
Python
# Find best brawler for each mode
best_brawlers_by_mode = brawler_mode_performance.orderBy(col("mode"), desc("win_rate"))
best_brawlers_by_mode_pd = best_brawlers_by_mode.toPandas()
# Get top brawler for each mode
top_brawler_by_mode = best_brawlers_by_mode_pd.groupby("mode").first().reset_index()
print("\nBest Brawler for Each Game Mode:")
print(top_brawler_by_mode[["mode", "brawler_name", "battles", "win_rate"]])
# Save brawler performance by mode data
best_brawlers_by_mode_pd.to_csv(f"{ANALYSIS_DIR}/brawler_performance_by_mode.csv", index=False)
# 7.2 Player Specialization Analysis
print("\nAnalyzing Player Specialization...")
# Calculate the proportion of each victory type for each player
player_specialization = player_df.withColumn(
"total_victories", col("3v3_victories") + col("solo_victories") + col("duo_victories")
).withColumn(
"3v3_proportion", col("3v3_victories") / col("total_victories")
).withColumn(
"solo_proportion", col("solo_victories") / col("total_victories")
).withColumn(
"duo_proportion", col("duo_victories") / col("total_victories")
)
# Define specialization categories
player_specialization = player_specialization.withColumn(
"specialization",
when(col("3v3_proportion") > 0.8, "3v3 Specialist")
.when(col("solo_proportion") > 0.2, "Solo Specialist")
.when(col("duo_proportion") > 0.2, "Duo Specialist")
.otherwise("Balanced")
)
Count players by specialization

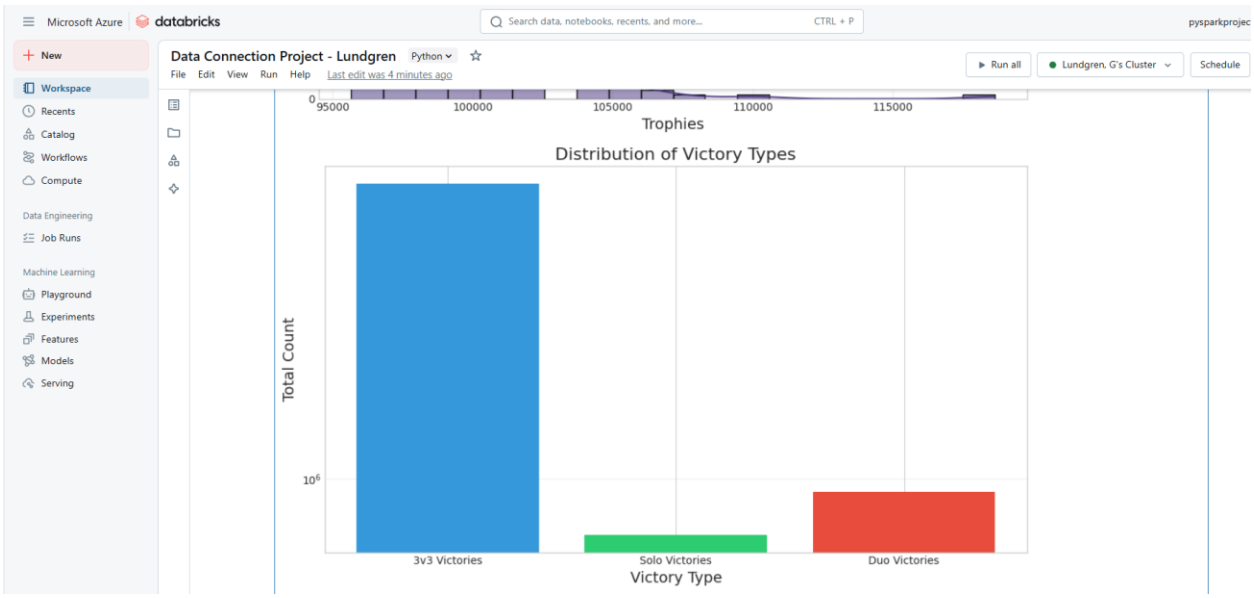


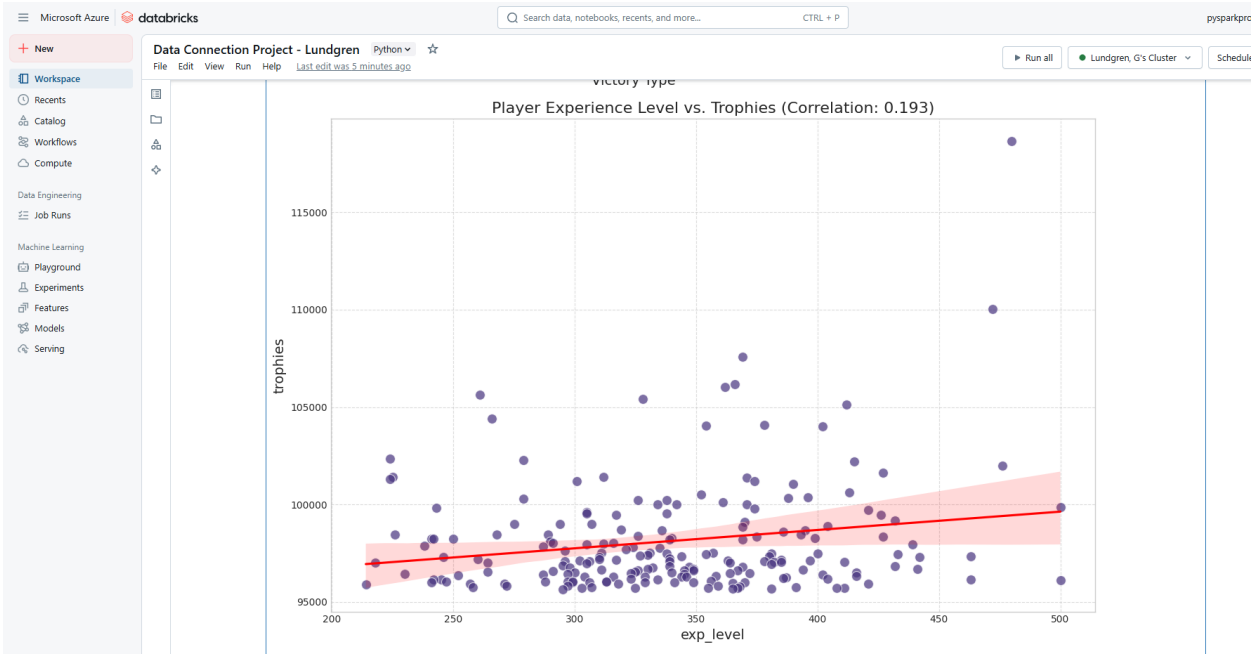
copied output:”

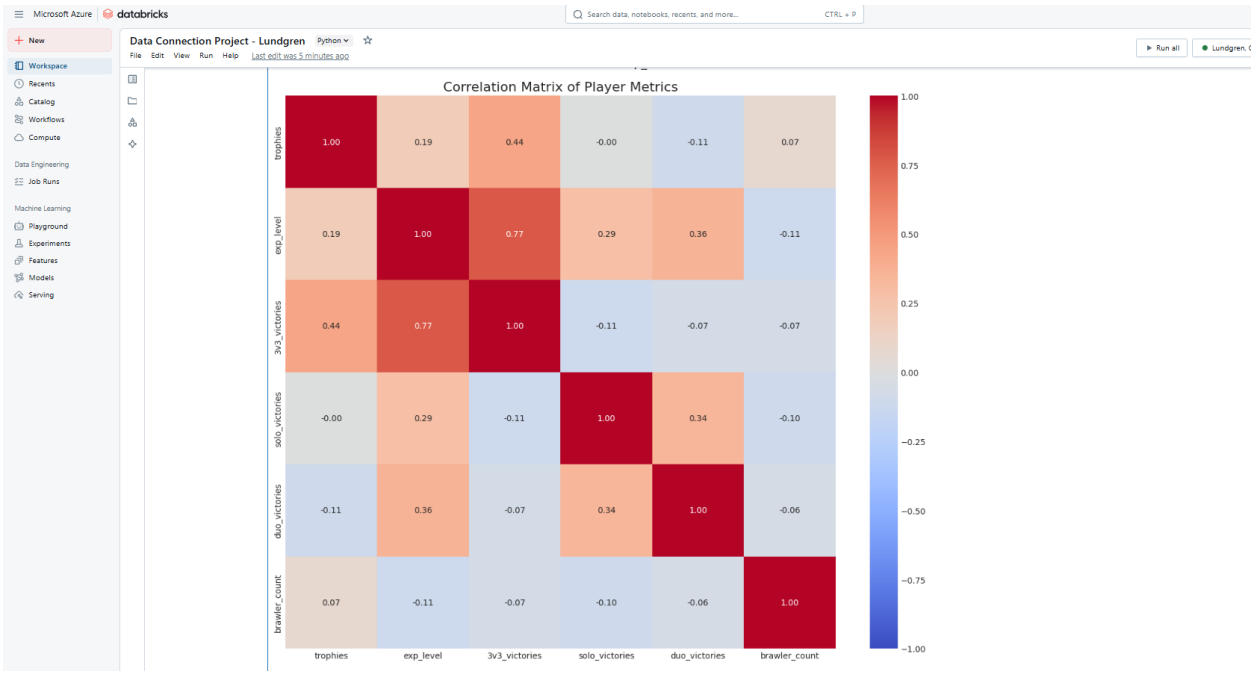
```
=== PLAYER ANALYSIS VISUALIZATIONS === Creating Player Trophy Distribution visualization... Saved player trophy distribution visualization Creating Victory Type Distribution visualization... Saved victory type distribution visualization Creating Experience Level vs Trophies visualization... Saved experience vs. trophies visualization Creating Player Metrics Correlation Heatmap... Saved player metrics correlation heatmap Creating Top Players visualization... /root/.ipykernel/869/command-7739057420998076-3231994127:73: UserWarning: Glyph 129293 (\N{WHITE HEART}) missing from current font.
```

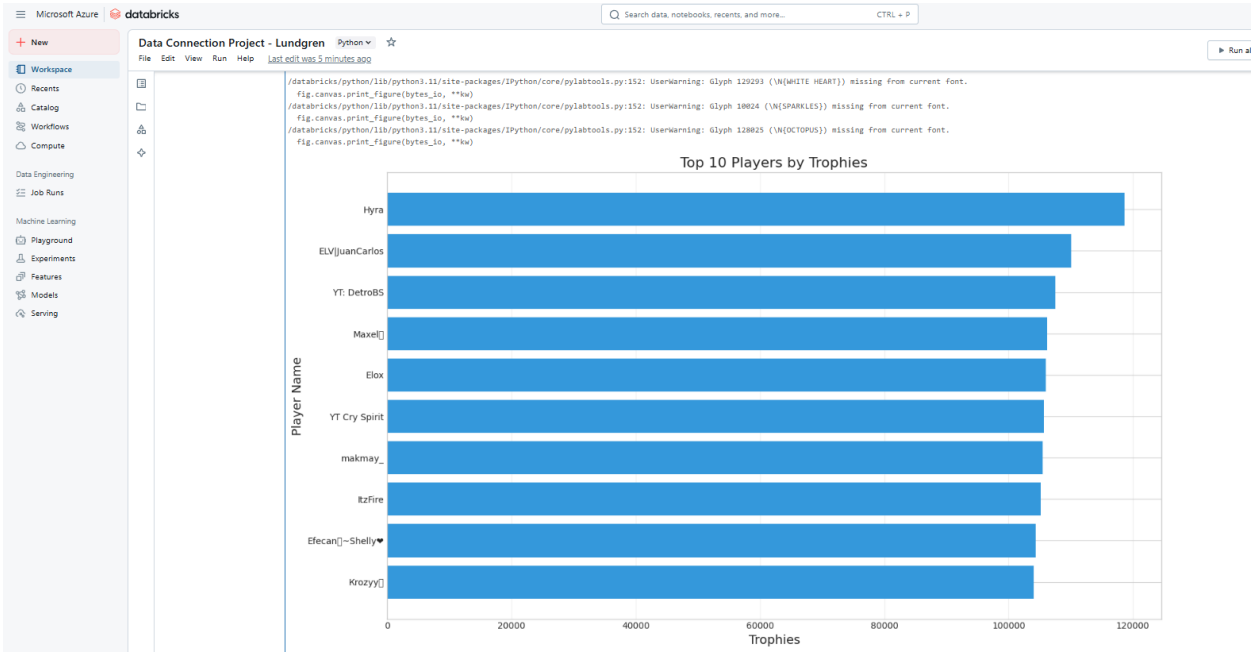
plt.tight_layout() /root/.ipykernel/869/command-7739057420998076-3231994127:73: UserWarning: Glyph 10024 (\N{SPARKLES}) missing from current font. plt.tight_layout() /root/.ipykernel/869/command-7739057420998076-3231994127:73: UserWarning: Glyph 128025 (\N{OCTOPUS}) missing from current font. plt.tight_layout() /root/.ipykernel/869/command-7739057420998076-3231994127:74: UserWarning: Glyph 129293 (\N{WHITE HEART}) missing from current font. plt.savefig(f"{ANALYSIS_DIR}/top_10_players.png") /root/.ipykernel/869/command-7739057420998076-3231994127:74: UserWarning: Glyph 10024 (\N{SPARKLES}) missing from current font. plt.savefig(f"{ANALYSIS_DIR}/top_10_players.png") /root/.ipykernel/869/command-7739057420998076-3231994127:74: UserWarning: Glyph 128025 (\N{OCTOPUS}) missing from current font. plt.savefig(f"{ANALYSIS_DIR}/top_10_players.png") Saved top players visualization Creating Win Rate by Experience Level visualization... Saved win rate by experience level visualization === BATTLE ANALYSIS VISUALIZATIONS === Creating Game Mode Popularity visualization... Saved game mode popularity visualization Creating Win Rate by Game Mode visualization... Saved win rate by game mode visualization Creating Top Brawlers by Usage visualization... Saved top brawlers by usage visualization Creating Top Brawlers by Win Rate visualization... Saved top brawlers by win rate visualization Creating Win Rate vs Brawler Power visualization... Saved win rate vs. power level visualization === ADVANCED ANALYTICS === Analyzing Brawler Performance by Game Mode... Best Brawler for Each Game Mode: mode brawler_name battles win_rate 0 bounty MR. P 10 1.000000 1 brawlBall PEARL 23 0.956522 2 duoShowdown PIPER 15 0.000000 3 gemGrab TARA 12 1.000000 4 heist CARL 10 0.900000 5 hotZone JESSIE 15 1.000000 6 knockout 8-BIT 28 0.928571 7 siege MORTIS 11 1.000000 8 wipeout 8-BIT 32 0.968750 Analyzing Player Specialization... Player Specialization Distribution: specialization count 0 3v3 Specialist 101 1 Balanced 60 2 Duo Specialist 26 3 Solo Specialist 13 Analyzing Win Rate by Player Specialization... Win Rate by Player Specialization: specialization battles win_rate 0 3v3 Specialist 1989 0.828557 1 Balanced 1155 0.737662 2 Solo Specialist 228 0.688596 3 Duo Specialist 492 0.646341 Creating Player Specialization visualization... Saved player specialization distribution visualization Creating Win Rate by Specialization visualization... Saved win rate by specialization visualization First half of Part 2 complete!”

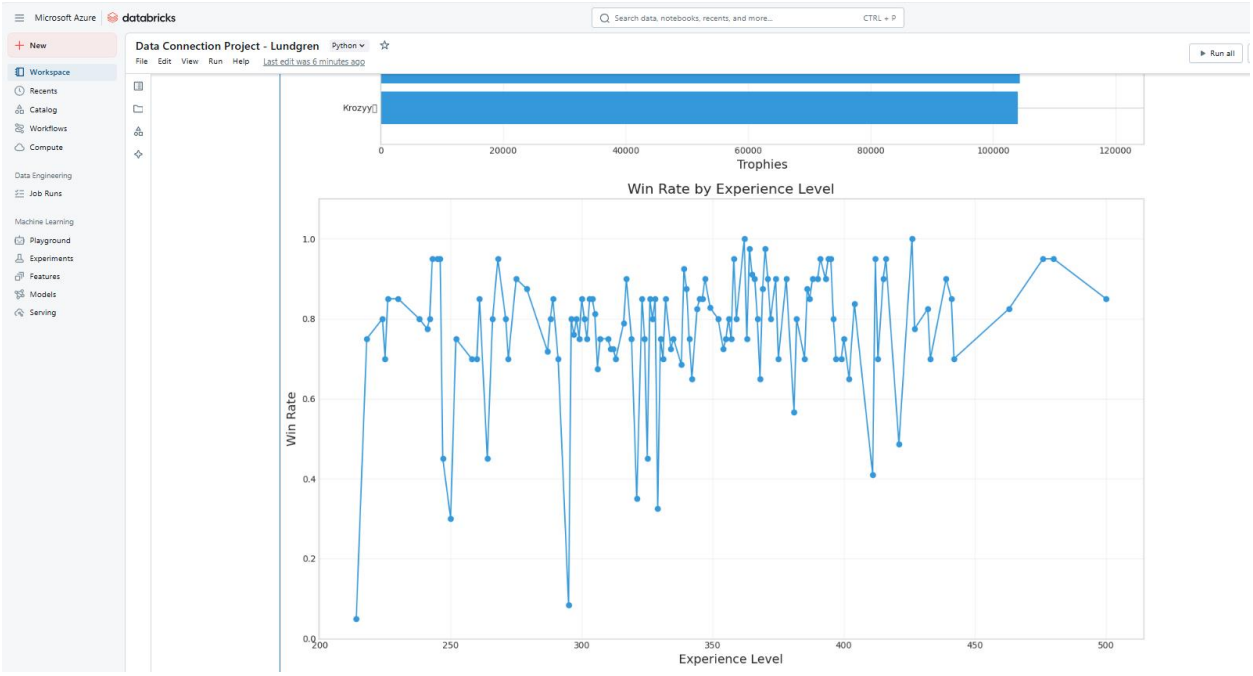

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
pyspark
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 4 minutes ago
Run all
Lundgren, G's Cluster
Sche
Saved player specialization distribution visualization
Creating Win Rate by Specialization visualization...
Saved win rate by specialization visualization
First half of Part 2 complete!
Distribution of Player Trophies
Number of Players
Trophies
0
10
20
30
40
50
60
70
80
95000
100000
105000
110000
115000
Distribution of Victory Types

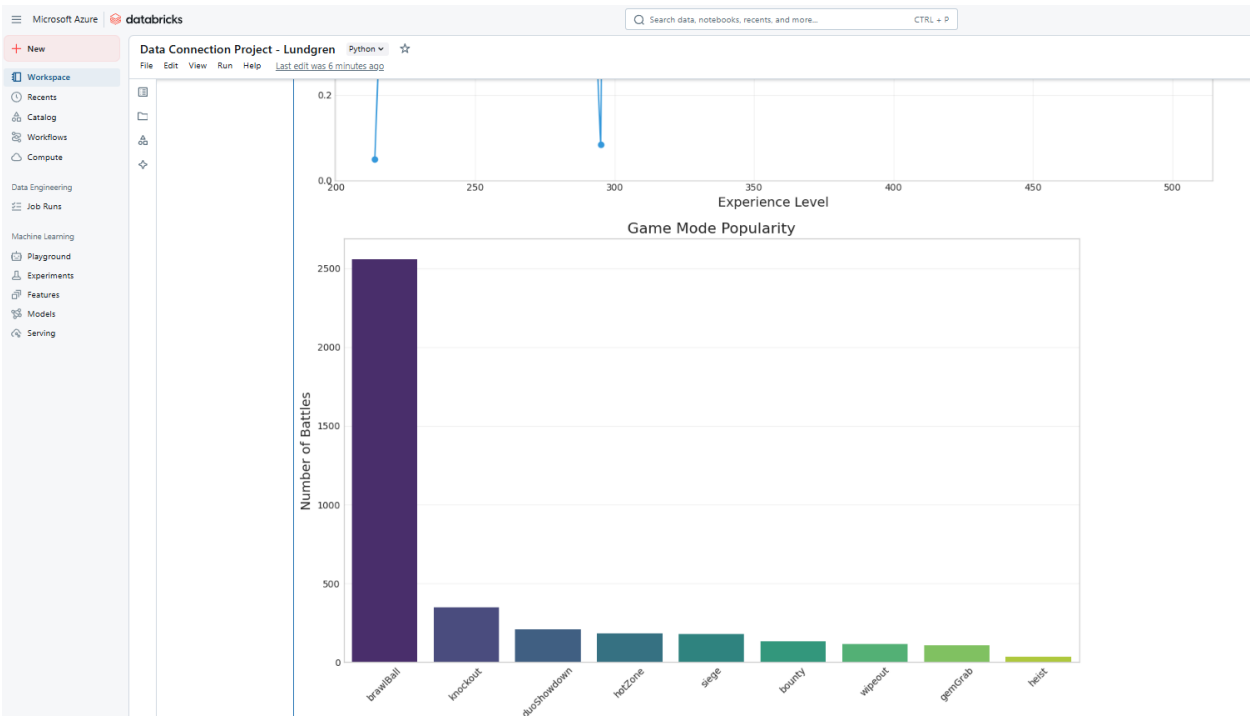

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
pysparkprojec
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 4 minutes ago
Run all
Lundgren, G's Cluster
Schedule
0
95000
100000
105000
110000
115000
Trophies
Distribution of Victory Types
Total Count
10^6
3v3 Victories
Solo Victories
Duo Victories
Victory Type

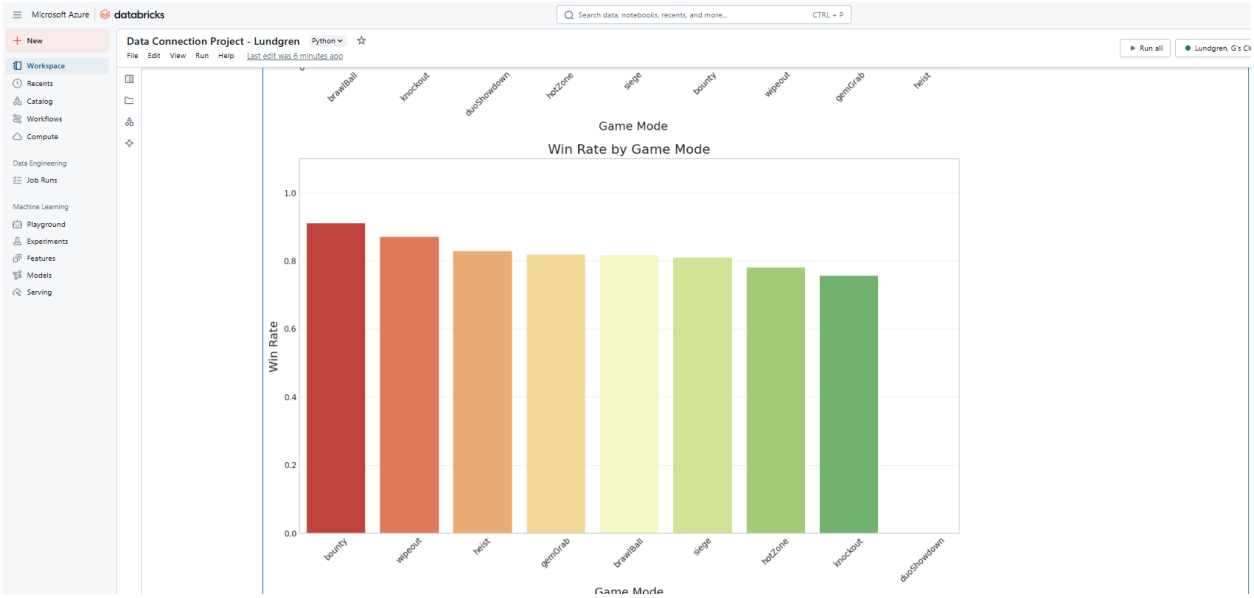

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
pysparkpro
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 5 minutes ago
Run all
Lundgren, G's Cluster
Victory Type
Player Experience Level vs. Trophies (Correlation: 0.193)
trophies
115000
110000
105000
100000
95000
200
250
300
350
400
450
500
exp_level

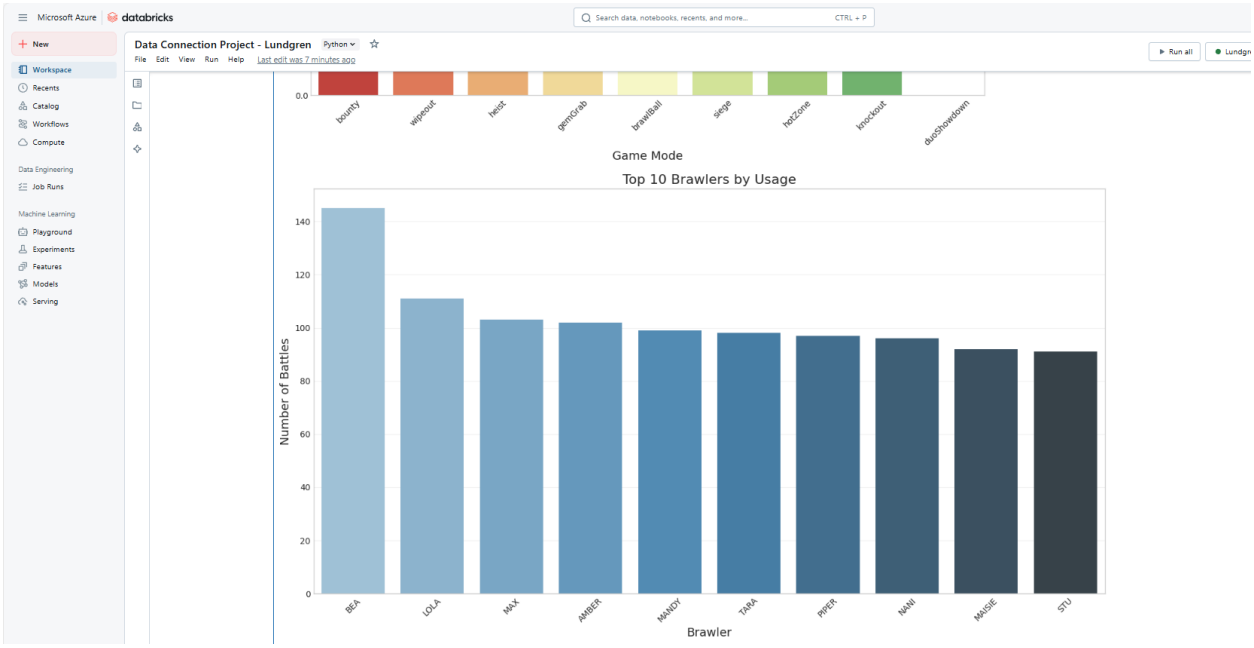

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 5 minutes ago
Run all
Correlation Matrix of Player Metrics
trophies
exp_level
3v3_victories
solo_victories
duo_victories
brawler_count
1.00 0.19 0.44 -0.00 -0.11 0.07
0.19 1.00 0.77 0.29 0.36 -0.11
0.44 0.77 1.00 -0.11 -0.07 -0.07
-0.00 0.29 -0.11 1.00 0.34 -0.10
-0.11 0.36 -0.07 0.34 1.00 -0.06
0.07 -0.11 -0.07 -0.10 -0.06 1.00
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00


Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 5 minutes ago
/databricks/python/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: UserWarning: Glyph 129293 (\N{WHITE HEART}) missing from current font.
fig.canvas.print_figure(bytes_io, **kw)
/databricks/python/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: UserWarning: Glyph 10024 (\N{SPARKLES}) missing from current font.
fig.canvas.print_figure(bytes_io, **kw)
/databricks/python/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: UserWarning: Glyph 128025 (\N{OCTOPUS}) missing from current font.
fig.canvas.print_figure(bytes_io, **kw)
Top 10 Players by Trophies
Player Name
Hyra
ELVIJuanCarlos
YT: DetroBS
Maxel
Elox
YT Cry Spirit
makmay_
ItzFire
Efecan~Shelly
Krozyy
0
20000
40000
60000
80000
100000
120000
Trophies


Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 6 minutes ago
Run all
Krozyy
0
20000
40000
60000
80000
100000
120000
Trophies
Win Rate by Experience Level
Win Rate
0.0
0.2
0.4
0.6
0.8
1.0
200
250
300
350
400
450
500
Experience Level


Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 6 minutes ago
0.0
0.2
200
250
300
350
400
450
500
Experience Level
Game Mode Popularity
Number of Battles
0
500
1000
1500
2000
2500
brawlBall
knockout
duoShowdown
hotZone
siege
bounty
wipeout
gemGrab
heist


Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 6 minutes ago
Run all
brawlBall
knockout
duoShowdown
hotZone
siege
bounty
wipeout
gemGrab
heist
Game Mode
Win Rate by Game Mode
Win Rate
1.0
0.8
0.6
0.4
0.2
0.0
bounty
wipeout
heist
gemGrab
brawlBall
siege
hotZone
knockout
duoShowdown
Game Mode


Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
New
Workspace
Recents
Catalog
Workflows
Compute
Data Engineering
Job Runs
Machine Learning
Playground
Experiments
Features
Models
Serving
Data Connection Project - Lundgren
Python
File Edit View Run Help
Last edit was 7 minutes ago
Run all
0.0
bounty
wipeout
heist
gemGrab
brawlBall
siege
hotZone
knockout
duoShowdown
Game Mode
Top 10 Brawlers by Usage
Number of Battles
140
120
100
80
60
40
20
0
BEA
LOLA
MAX
AMBER
MANDY
TARA
PIPER
NANI
MAISIE
STU
Brawler

This analysis provides valuable insights into the factors that contribute to success in Brawl Stars at the highest competitive level. The data clearly shows that specialization in 3v3 game modes, strategic

brawler selection for specific modes, and maximizing brawler power levels are key strategies employed by top players. The negative correlation between solo/duo victories and win rate suggests that the skills developed in team play may be more valuable for overall success than those developed in solo competition.

These findings could be valuable for players looking to improve their performance, game developers balancing game mechanics, and the broader gaming community interested in understanding competitive dynamics in Brawl Stars.

As someone in the top .1% (ish) of players, I would say these findings hold true to my own personal beliefs. The competitive esports scene only has team game modes, the solo and duo showdown modes are not even taken into account.