Report On

Title of the Course Project

Submitted in partial fulfillment of the requirements of the Course project in
Semester V of Third Year Computer Engineering

by
Shruti Jagdish Borhade (Roll No. 20)
Anushka Kailas Chavan (Roll No. 26)
Tanishka Jayesh Das (Roll No. 36)

Supervisor
Prof. Swapna Borde

University of Mumbai

Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

(2023-24)

Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

CERTIFICATE

This is to certify that the project entitled “Title of the project” is a bonafide work of "Shruti Jagdish Borhade (Roll No. 20),Anushka Kailas Chavan (Roll No. 26),Tanishka Jayesh Das (Roll No. 36)" submitted to the University of Mumbai in partial fulfillment of the requirement for the Course project in semester VI of Third Year Computer Engineering.

Supervisor

Prof. Swapna Borde

Internal Examiner

External Examiner

Dr Megha Trivedi
Head of Department

Dr. H.V. Vankudre
Principal

Contents

1. **Section One: Overview of the Valorant Stats Analysis Project**

   The Valorant Stats Analysis project aims to perform a comprehensive data analysis on the popular first-person shooter game, Valorant, using data sourced from Kaggle. Valorant is a competitive, team-based tactical shooter where players choose from a roster of agents, each with distinct abilities, and play on various maps in 5v5 match formats. With an emphasis on understanding player behaviour, agent effectiveness, map influence, and other in-game dynamics, this project employs a variety of data science methods to uncover insights that can help players, teams, and analysts optimize strategies and improve performance.

The dataset comprises a wealth of information across multiple dimensions of the game, such as agents, player statistics, maps, teams, and weapon usage. By leveraging data science techniques like statistical analysis, machine learning, and visualization, the project aims to identify key patterns, predict outcomes, and uncover correlations between various game features. This analysis not only enhances the understanding of game mechanics but also provides actionable insights for gameplay improvement, competitive strategies, and meta-analysis.

1.1 Subsection One of Section One: Project Structure

The project is organized into a series of well-defined components, each responsible for analysing specific aspects of the dataset. These components ensure a systematic approach to data preprocessing, exploration, modelling, and visualization. The structure helps to focus the analysis on extracting meaningful insights and answering critical questions about Valorant gameplay.

1.1.1 Dataset Overview

The dataset is organized into several tables, each providing different facets of game data that can be analysed in isolation or combined for more in-depth insights:

- **Agents Data**: Contains information on the agents (characters), including their unique abilities, roles (e.g., Due list, Initiator), and frequency of use in matches. This data can help identify popular agents and those with underutilized potential.
- **Maps Data**: Includes details about the different maps used in gameplay, with data points capturing the map's layout, common choke points, site locations, and any influence on strategic gameplay. Analysing map data can reveal the effectiveness of specific agents or strategies on different maps.
- **Players Data**: Provides player-specific statistics such as kills, deaths, assists, headshot percentage, accuracy, damage per round, and overall score. This information is crucial for understanding individual and team performance.
- **Teams Data**: Captures team-level information, including the agents selected by each team, match results (wins/losses), and team-based performance metrics like average kill-death ratio and teamwork efficiency.
- **Weapons Data**: Contains information on weapon usage, including weapon types (e.g., rifles, SMGs, snipers), damage statistics, accuracy, kill ratios, and player preferences. Analysing this data can help determine the most effective weapons in different situations or for different playstyles.

1.1.2 Analysis Methods

To achieve the project's goals, a range of data analysis and machine learning methods are applied. Each technique is selected for its ability to answer specific research questions about the game:

- **Identifying the Top 10 Most Popular Agents**: This analysis ranks agents based on their frequency of selection across matches, which can shed light on the current meta and player preferences. Understanding agent popularity also aids in predicting which agents are likely to appear in competitive matches.
- **Linear Regression Analysis**: Regression models are used to find relationships between game variables, such as predicting player performance (e.g., kills, accuracy) based on other factors like the selected agent, map, or team composition. These models can help quantify the impact of specific variables on player success.
- **Decision Tree Classification**: A decision tree is built to predict match outcomes based on features like agent choice, map played, and player statistics. This approach provides insights into key factors that influence the likelihood of winning, helping players and teams refine their strategies.
- **Heatmap Visualizations**: Correlation heatmaps are generated to visualize relationships between variables such as agent selection, map characteristics, weapon preferences, and player performance metrics. These visualizations highlight significant patterns that might not be apparent from raw data.
- **K-Means Clustering**: Players, agents, and teams are grouped into clusters based on statistical similarities. For example, agents can be clustered according to their performance metrics on different maps,

revealing which agents are better suited for certain playstyles or strategies.

- **Outlier Detection**: This technique is used to detect exceptional performances or anomalies in the data. For instance, identifying players who consistently outperform others on specific maps or with certain agents can highlight unique skills or strategies.

1.2 Subsection Two of Section One: Project Goals

The main objectives of the Valorant Stats Analysis project are centered around understanding game dynamics and answering key questions that can benefit players, teams, and analysts. The goals of the project are:

1.2.1 Top 10 Most Popular Agents

The analysis seeks to rank agents based on their frequency of selection in matches, revealing player preferences and reflecting the in-game meta. This information helps players choose agents strategically and anticipate common compositions during matches.

1.2.2 Linear Regression Analysis

By performing linear regression, the project explores relationships between performance metrics (e.g., kills, accuracy, deaths) and external factors like map choice or agent selection. This analysis aims to predict how changing one variable, such as playing on a different map or using a specific agent, might impact a player's performance.

1.2.3 Decision Tree Classifier

The decision tree model is used to classify players or predict match outcomes based on various features such as agent choice, team composition, and individual performance metrics. This approach can identify important decision points that significantly influence match results, providing data-driven insights for competitive play.

1.2.4 Heatmap Visualizations

Heatmaps provide an intuitive way to understand correlations between different features. For example, heatmaps may reveal that certain agents perform better on specific maps or that players using certain weapons tend to have higher accuracy. These insights can guide decision-making in gameplay.

1.2.5 K-Means Clustering

The clustering analysis aims to identify common trends in playstyles or strategies by grouping players, teams, or agents based on their statistics. For instance, clusters may show that some agents are more frequently used by players who prefer aggressive playstyles, while others are favoured in defensive setups.

1.2.6 Outlier Detection

Detecting outliers helps identify players or agents with exceptional or unusual performance. Such anomalies might indicate high skill, unique strategies, or areas where game balance could be improved.

1.3 Subsection Three of Section One: Tech Stack and Execution

The project utilizes a range of tools and libraries tailored for data analysis, machine learning, and visualization:

- **Python**: The programming language used for data analysis and machine learning implementations.
- **Pandas and NumPy**: Libraries for data handling and preprocessing, including data cleaning, manipulation, and transformation.
- **Scikit-learn**: Used for implementing machine learning models, including linear regression, decision trees, clustering, and outlier detection.

- **Seaborn and Matplotlib**: Visualization libraries used to create heatmaps, scatter plots, and other graphs for data interpretation.

1.3.1 How to Run the Project

To run the project, follow these steps:

1. **Clone the Repository**: Clone the project repository from GitHub using:

bash

git clone https://github.com/your-username/val-stats-analysis.git

2. **Install Dependencies**: Install dependencies listed in the requirements.txt file:

bash

pip install -r requirements.txt

3. **Run the Jupiter Notebooks or Python Scripts**: Execute the main analysis notebook (ValoPlayer'sStats.ipynb) or scripts to view results.

This expanded content provides a thorough description, ensuring the reader has a clear understanding of the project, its goals, and execution.

2. **Section Two: Machine Learning and Data Analysis Techniques**

   This section delves into the machine learning models and data analysis techniques utilized in the Valorant Stats Analysis project. These methods are employed to interpret the dataset and uncover valuable insights into player behaviours, agent effectiveness, map strategies, and overall gameplay trends. By applying a range of statistical and machine learning approaches, the project is able to go beyond basic data exploration and uncover deeper patterns and relationships within the game data.

2.1 Subsection One of Section Two: Machine Learning Models and Techniques

The project leverages several machine learning models and techniques, each designed to address specific analytical goals. The methods vary from supervised learning, where the goal is to predict outcomes based on labelled data, to unsupervised learning, which aims to find inherent patterns without pre-defined categories. These models are essential for answering questions about performance, strategy, and the dynamics of the game.

2.1.1 Linear Regression

Linear regression is one of the simplest yet powerful techniques used in this project to understand the relationships between different variables. It is applied to predict player performance metrics, such as kills, deaths, or assists, based on other features like agent selection, map, or weapon choice. The model fits a linear equation to the observed data, allowing us to predict outcomes based on the influence of these variables.

For example, linear regression can help answer questions like: "Does playing on a particular map tend to increase a player's kill-death ratio?" or "How does the choice of agent affect a player's accuracy in landing headshots?" By analysing the coefficients, we can also quantify the impact of individual factors on the target variable. The simplicity of linear regression makes it a good starting point for understanding the data, while more complex relationships might require other techniques.

2.1.2 Decision Tree Classifier

A decision tree classifier is used to predict the outcomes of matches or categorize players based on various game features. The decision tree model splits the data into branches based on decision rules derived from the features, such as agent choice, map played, or player performance statistics. This hierarchical structure of decisions breaks down the prediction process into a series of simpler questions, making the decision tree model particularly useful for interpreting how different factors contribute to match outcomes.

For instance, the decision tree can reveal which agents are most likely to lead to victory on certain maps or highlight key performance metrics (like accuracy or number of kills) that strongly predict match results. Decision trees are valuable not only for their predictive power but also for their interpretability, as they provide a clear visual representation of the decision-making process. Moreover, decision trees can be fine-tuned to improve accuracy, and techniques like pruning can be used to avoid overfitting by simplifying the model.

2.1.3 K-Means Clustering

K-Means clustering is an unsupervised learning technique that groups similar data points into clusters. In the context of this project, K-Means is applied to cluster players, teams, or agents based on statistical similarities. The technique helps in identifying common trends, such as groups of players who share similar playstyles (e.g., aggressive versus defensive players) or teams that perform exceptionally well on certain maps.

By analysing these clusters, we can better understand how different factors influence player or team behaviours and uncover latent structures in the data. For example, clusters may show that certain agents tend to be favoured by players who excel in head-to-head confrontations, while other agents might cluster with players who focus on supporting their teammates. This information can guide players and teams in selecting strategies that align with their playstyle or counter the tactics of opponents. K-Means clustering can also be used to segment matches into different types based on factors like average round length, weapon usage patterns, or team compositions, revealing deeper gameplay dynamics.

2.1.4 Outlier Detection

Outlier detection is an essential technique for identifying data points that deviate significantly from the norm. In this project, outlier detection is used to find unusual performances or rare events in the dataset. Outliers may represent exceptionally high player scores, unexpected match outcomes, or uncharacteristic agent choices that differ from typical trends. Detecting these anomalies can provide insights into unique strategies or highlight players with exceptional skills.

For example, if a player consistently achieves higher accuracy than others when using a specific weapon, this could indicate a level of expertise or a novel playstyle that could be emulated. Similarly, identifying matches where certain underutilized agents were key to a team's victory can suggest untapped strategic potential in the game. Outlier detection methods can also help improve game balance by flagging agents, weapons, or maps that may be disproportionately strong or weak.

2.1.5 Support Vector Machines (SVM)

In addition to the techniques mentioned above, the project can incorporate Support Vector Machines (SVM) for classification tasks. SVM is a powerful supervised learning model used to separate data points into different classes by finding the hyperplane that best divides the dataset. In the context of Valorant analysis, SVM could be used to classify players into skill tiers based on their performance metrics or to predict whether a match will result in a win or loss based on pre-match statistics.

The advantage of SVM lies in its ability to handle high-dimensional data and its robustness in cases where the classes are not linearly separable, through the use of kernel functions. This makes it a suitable technique for dealing with complex relationships between game variables. The model's ability to provide a clear boundary between classes adds to its interpretability, particularly when visualizing the decision surface.

2.1.6 Principal Component Analysis (PCA)

PCA is a dimensionality reduction technique used to reduce the complexity of the dataset by transforming it into a set of orthogonal components. In this project, PCA can be applied to simplify the dataset by reducing the number of features while retaining most of the variance in the data. This is especially useful for visualizing high-dimensional data, such as player statistics, and for improving the performance of other machine learning models by removing noise.

For example, PCA can help identify the most significant features that contribute to player performance, such as certain combinations of accuracy, kills, and map choices. Reducing the dataset to a smaller set of principal components makes it easier to perform further analysis or apply clustering techniques like K-Means.

2.2 Subsection Two of Section Two: Data Analysis Techniques

Data analysis techniques complement machine learning methods by enabling a deeper exploration of the dataset. Techniques such as data cleaning, exploratory data analysis (EDA), and feature engineering are critical in preparing the data for modelling and ensuring that meaningful insights can be derived.

2.2.1 Data Cleaning and Preprocessing

Before applying any machine learning algorithms, it is crucial to clean and preprocess the data. This involves handling missing values, removing duplicate entries, and transforming data into a suitable format for analysis. For instance, converting categorical data such as agent names into numerical representations using one-hot encoding or label encoding is a common preprocessing step.

2.2.2 Exploratory Data Analysis (EDA)

EDA involves using statistical methods and visualizations to summarize the main characteristics of the data. Techniques such as plotting histograms, scatter plots, and correlation matrices can reveal underlying patterns, distributions, and relationships in the data. EDA helps in identifying which variables are most important for further analysis and can guide the selection of appropriate machine learning techniques.

2.2.3 Feature Engineering

Feature engineering is the process of creating new features from the existing data to improve model performance. In this project, features such as "average kills per map," "headshot accuracy ratio," or "agent usage frequency" can be derived from raw data. Creating these new features can enhance the predictive power of the models by providing more relevant input data.

This expanded content provides a deeper dive into the various machine learning and data analysis techniques used in the project, explaining their roles and potential benefits in extracting insights from the dataset.

3. Section Three: Conclusions and Future Work

The Valorant Stats Analysis project successfully utilized the Kaggle dataset to extract valuable insights into various facets of the game, including player behaviours, agent effectiveness, map strategies, and weapon preferences. Through a combination of data analysis techniques and machine learning models, the project uncovered meaningful patterns that provide a deeper understanding of the game's dynamics. These findings not only offer players and teams data-driven strategies for improving their gameplay but also serve as a valuable resource for game developers and analysts interested in balancing the game or studying evolving player trends. The results highlight the potential of data-driven approaches in the competitive gaming landscape, setting a foundation for further explorations into more complex analytical models.

3.1 Key Findings

- **Popular Agents:** The analysis successfully identified the top 10 most frequently selected agents in Valorant, offering a snapshot of the current in-game meta and reflecting player preferences. Understanding which agents are most popular helps inform game balance changes, as frequently selected agents may be seen as overpowered or simply fitting well within current strategic trends. This information could also be used by players looking to adjust their own strategies to counter or incorporate these popular agents into their team compositions effectively. In addition, analysing how agent popularity changes over time can provide insights into how game patches, updates, or new agent releases impact the metagame.

- **Performance Predictors:** The project uncovered significant relationships between player performance metrics (such as kill-death ratios, accuracy, and assists) and game variables like agent selection and map choice using linear regression models. These relationships suggest that certain agents or maps may favour specific playstyles or strategies, influencing overall player effectiveness. For instance, some maps might promote more aggressive tactics, leading to higher average kills, while others might encourage defensive play. This understanding of performance predictors can help players make more informed decisions about agent selection and map strategies before entering matches, while developers can use these insights to tweak map layouts or adjust agent abilities for balanced gameplay.
- **Predictive Modelling:** The decision tree models demonstrated the ability to predict game outcomes based on key features such as agent selection, player performance metrics, and map data. These predictive models provide a systematic approach to strategic planning, offering players a tool to anticipate match results and refine their gameplay accordingly. By identifying critical decision points, such as the importance of specific agents on certain maps, players can prioritize their choices to increase their chances of winning. Additionally, further refinement of these models, potentially through ensemble methods like random forests or boosting, could enhance prediction accuracy, making the models even more valuable for competitive play or automated coaching systems.
- **Pattern Discovery:** The application of K-Means clustering revealed groups of players, teams, or agents that share similar characteristics, aiding in understanding common strategies and playstyles. These clusters can help identify typical player behaviours, such as aggressive, passive, or balanced playstyles, based on performance statistics. For

example, clustering may show that certain agents are more commonly used by players with higher accuracy or higher engagement rates, suggesting an association between agent abilities and player playstyle. Such insights can be useful for teams developing specific strategies or for analysts looking to understand shifts in the competitive landscape, such as how certain tactics become more prevalent over time.

- **Anomaly Detection:** The project's outlier detection techniques successfully identified data points that deviated significantly from normal gameplay patterns. These outliers could indicate exceptional player performances, unusual agent choices, or unexpected match outcomes that stand out from the typical gameplay data. Highlighting these anomalies can bring attention to unique strategies that deviate from standard meta trends, showcasing innovative approaches used by top players or teams. Furthermore, anomaly detection can also help identify potential bugs, exploits, or data quality issues in the game that developers may need to address, ensuring the game's fairness and integrity.

3.2 Future Work

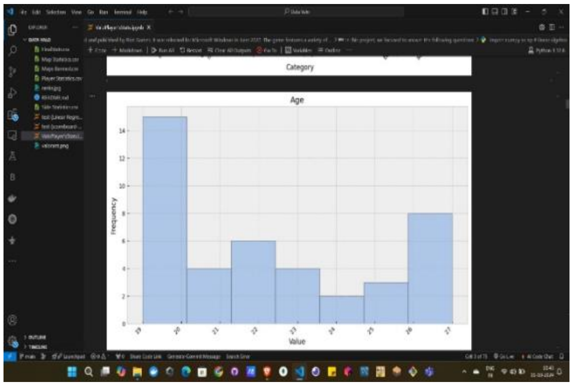
- **Deep Learning Models:** The next step in enhancing the predictive capabilities of the project could involve implementing more advanced models such as neural networks. Deep learning models, particularly convolutional neural networks (CNNs) or recurrent neural networks (RNNs), could capture complex, non-linear relationships within the data, potentially improving prediction accuracy for match outcomes or player performance. Incorporating deep learning could also allow for more sophisticated feature extraction from the data, automatically identifying important patterns that might not be apparent through traditional methods, thus opening new possibilities for high-level strategic insights.

- **Real-Time Analysis:** Implementing the current analyses within a real-time framework could significantly enhance the project's applicability, allowing for live analysis of ongoing games or streams. Real-time data processing could provide instant feedback to players and viewers, such as predicting the likelihood of winning based on current in-game metrics or suggesting agent or strategy adjustments on the fly. This could be integrated into live broadcasts for more engaging content or be used by professional teams to adjust strategies during matches, making the insights not only informative but also actionable in real-time.
- **User Interface:** To make the insights more accessible to a broader audience, developing a user-friendly interface or dashboard is essential. A web-based dashboard could present the analysis results through interactive visualizations, such as dynamic heatmaps, performance graphs, and predictive outcome probabilities. This would allow players, analysts, and coaches to explore the data intuitively without needing a data science background. Additionally, implementing features like custom query capabilities or personalized performance analysis could tailor the experience to individual users, making the platform versatile for both casual players and professional teams.

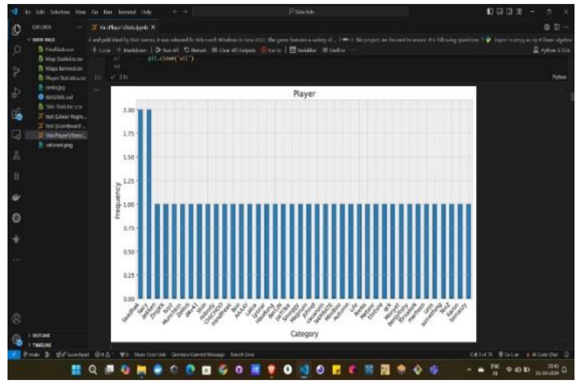
3.3 Contributing to the Project

The Valorant Stats Analysis project is open for collaboration and welcomes contributions from the community. There are several ways to get involved:

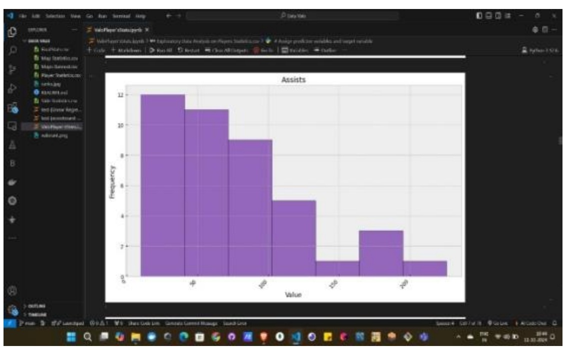
- **Fork the Repository:** The project repository can be forked from GitHub, allowing contributors to make enhancements or address existing issues. Interested developers can add new features, improve the current analysis methods, or explore other datasets that could be integrated into the project for more comprehensive insights. Contributors are encouraged to document their changes clearly and provide detailed explanations for any new algorithms or techniques introduced.
- **Install Dependencies:** Before contributing, ensure that all the necessary dependencies are installed. These can be easily set up by running the provided requirements.txt file. Setting up a consistent development environment helps prevent compatibility issues and ensures that the project runs smoothly across different systems. Additionally, contributors can help by updating the documentation and adding any newly required dependencies for future enhancements.
- **Run the Main Program:** To replicate the analyses and experiment with the models, run the primary Jupiter notebook, ValoPlayer'sStats.ipynb. Contributors can use this notebook as a starting point for testing new models, adding new data visualizations, or optimizing existing code for better performance. Users are encouraged to share their insights and findings, which could help refine the project's methodologies or inspire new avenues for analysis.
- **Submit Pull Requests:** Once modifications or enhancements are made, contributors can submit pull requests to share their improvements with the community. The project maintainers will review the submissions, providing feedback or suggestions where necessary, and eventually merge accepted changes, helping the project grow and evolve with community input.



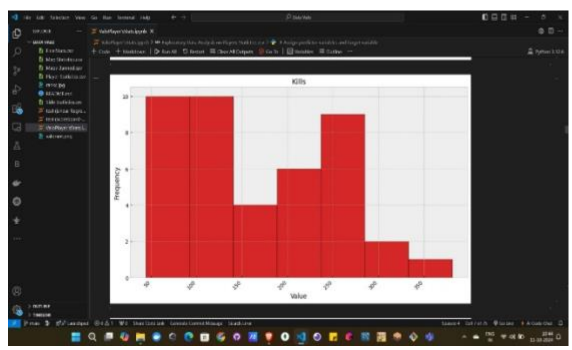
1.Age



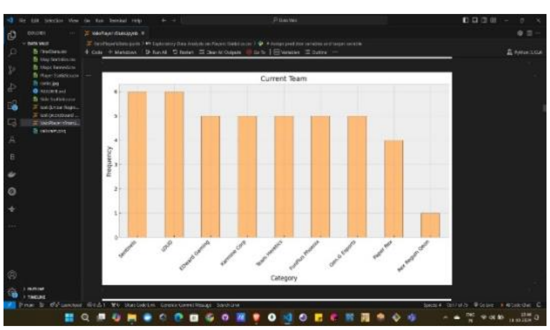
2.Player



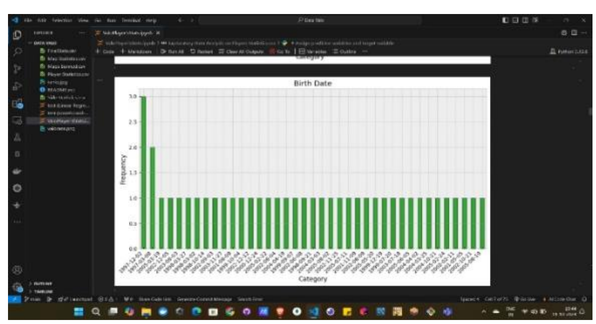
3.Assists



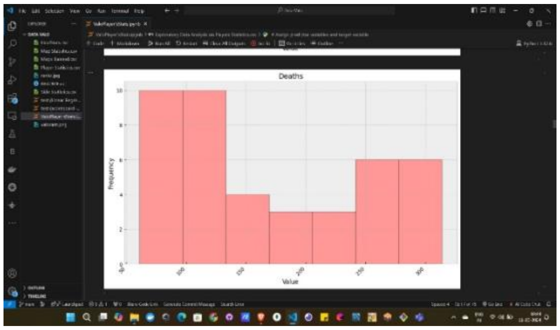
4.Kills



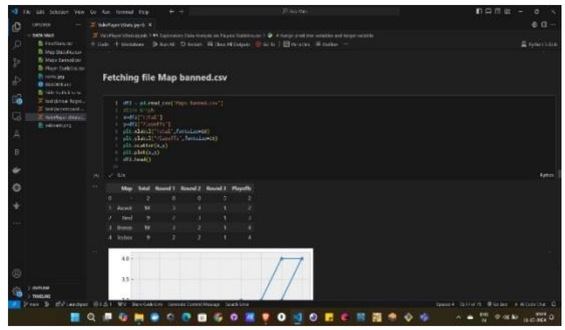
5.Current Team



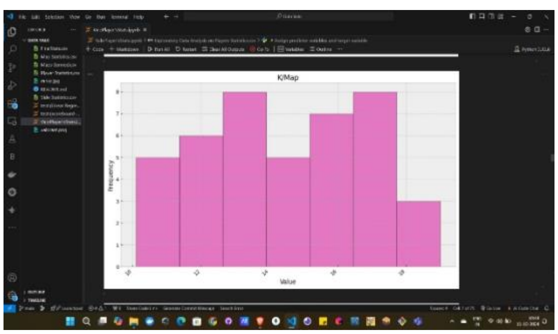
6.Birth Date



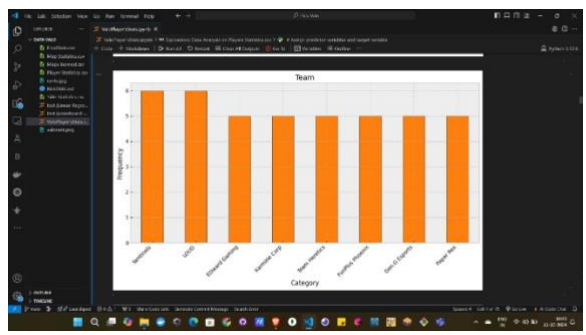
7.Deaths



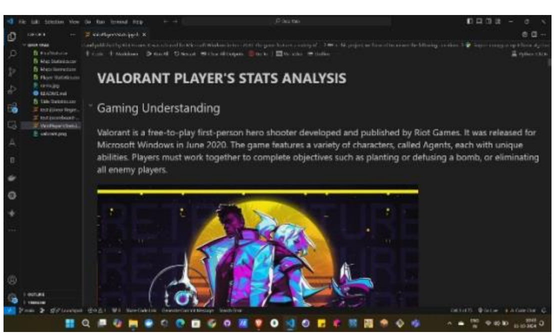
8.Fetching file map banned



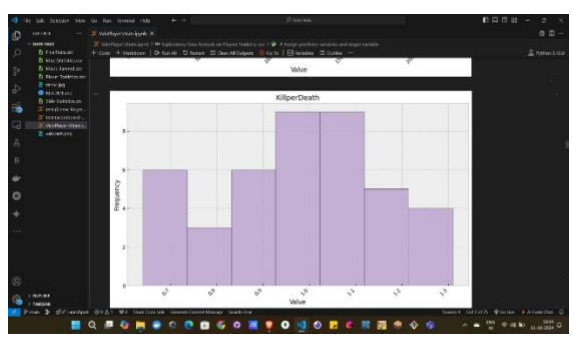
9.K-map



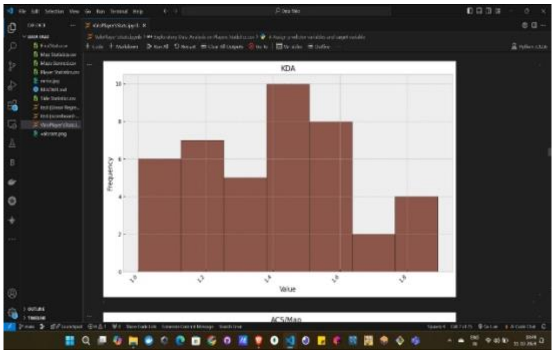
10. Team



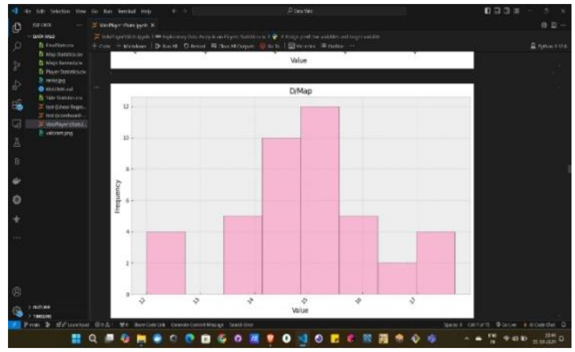
11.Gaming Understanding



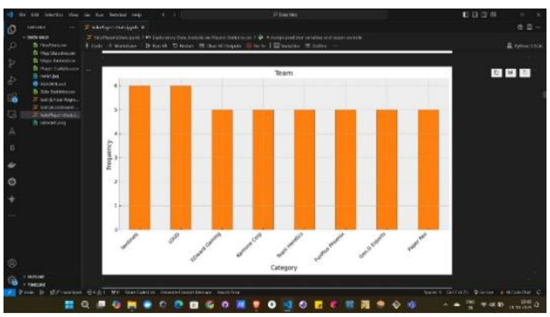
12.Kill per death



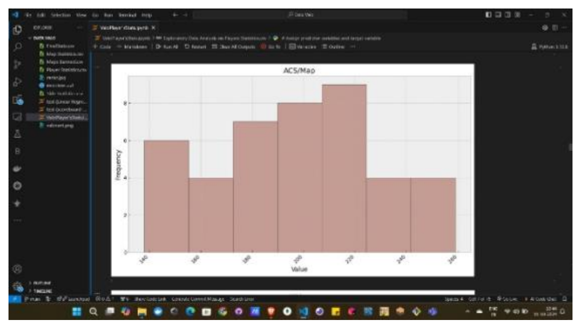
13.KDA



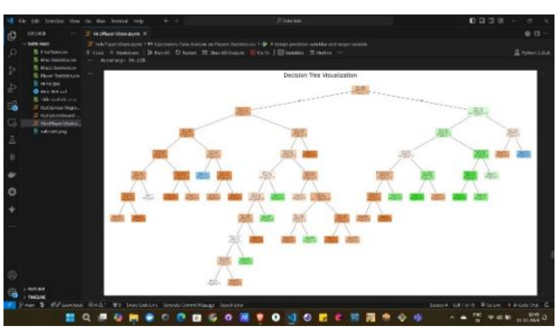
14.D/map



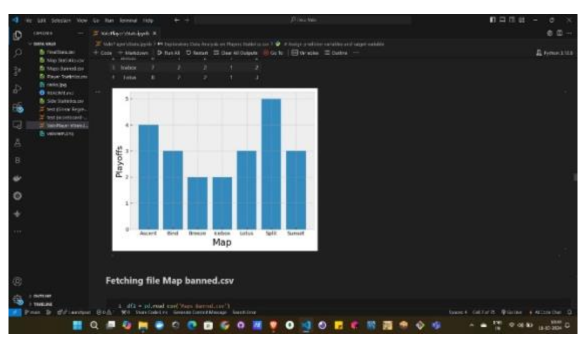
15.Team



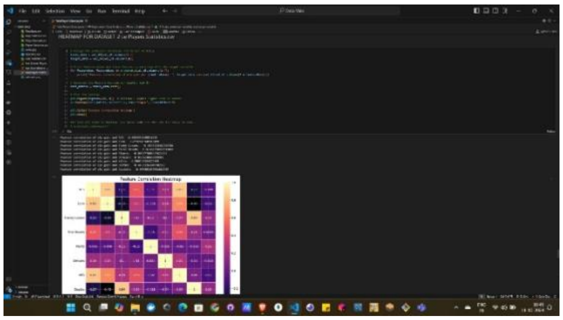
16. ACS/map



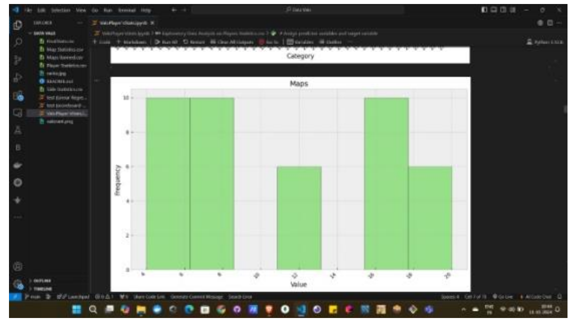
17.Decision tree visualization



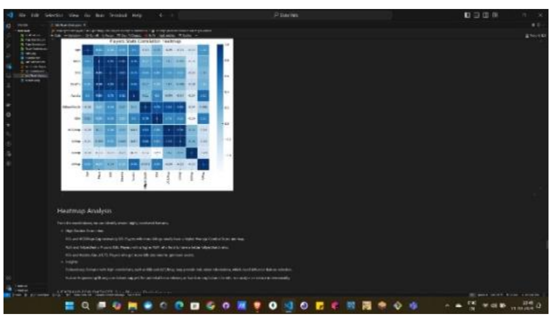
18. Playoffs



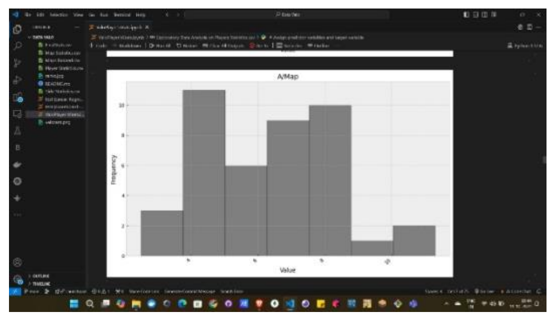
19.Hitmap



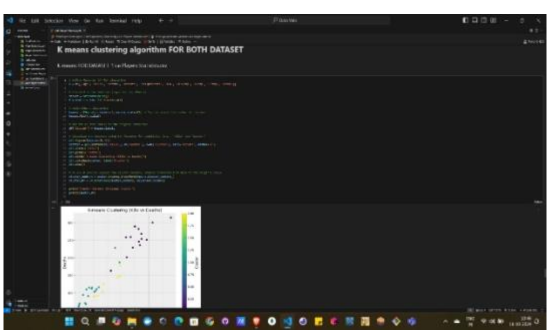
20.Frequency map



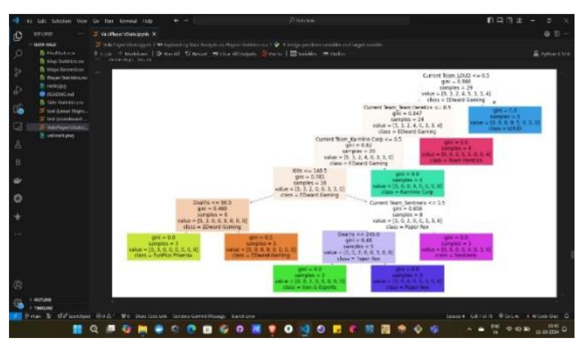
21. Heatmap for dataset



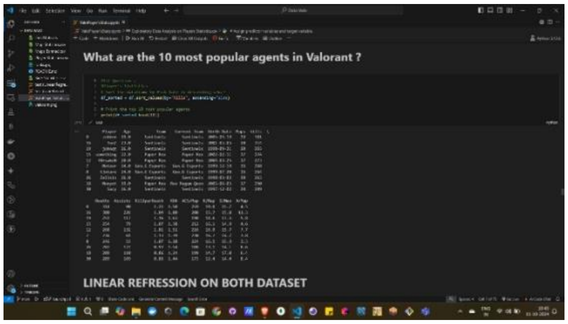
22. A/map



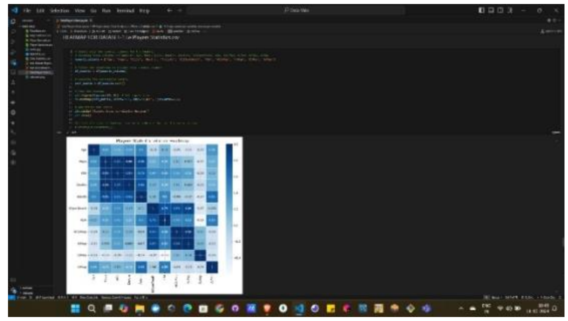
23. K means



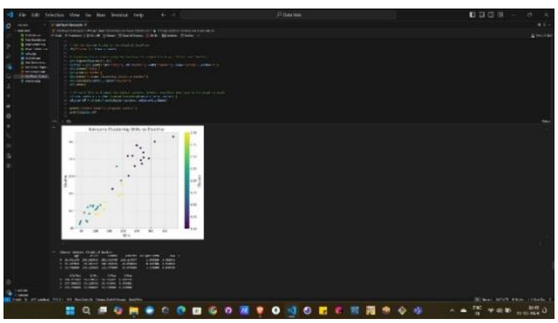
24. Graphic representation



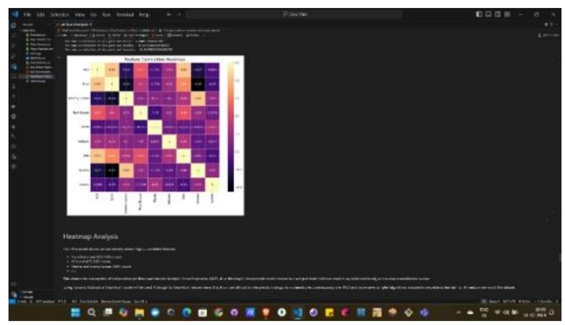
25. Linear regression



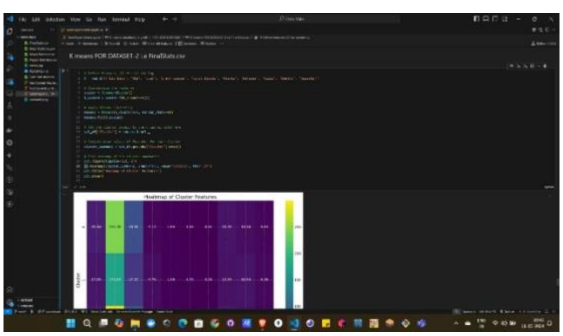
26. Heatmap for dataset



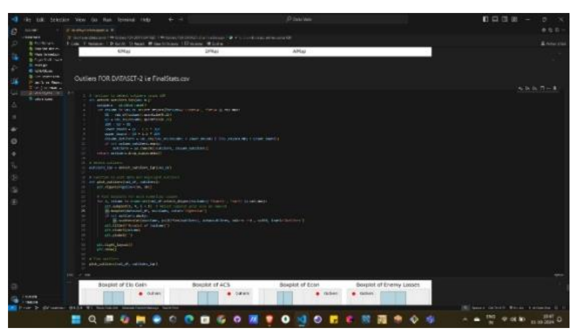
27. Kmeans for both dataset



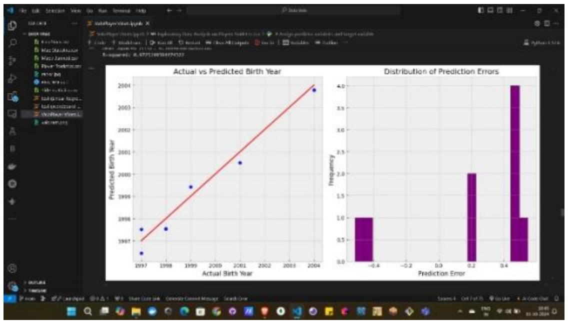
28. Heatmap for dataset

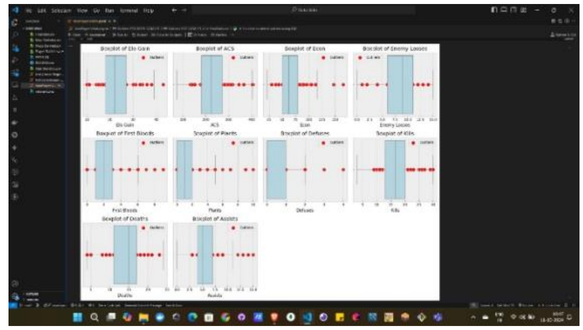


29. k means for dataset-2



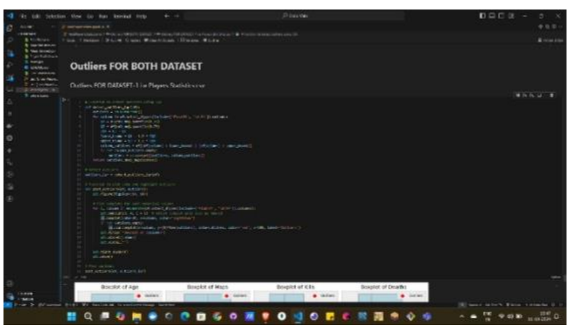
30. Outliers

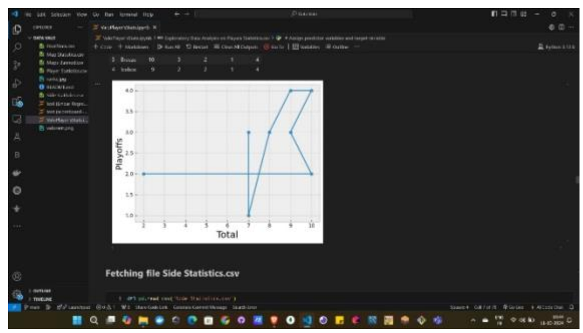




31. Prediction error

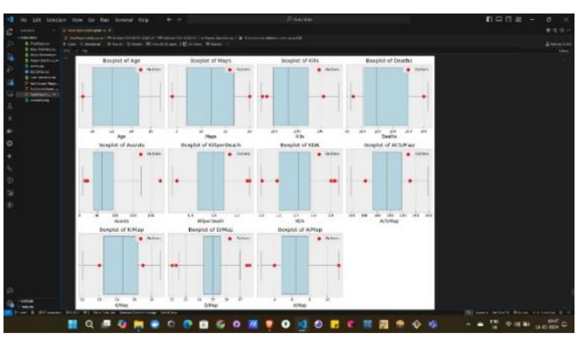
32. Boxplot

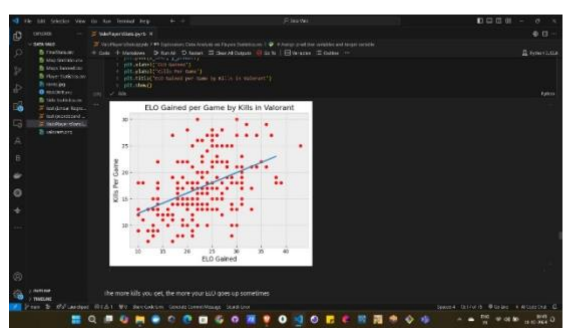




33. Outliers for both dataset

34. Playoffs





35. Boxplots

36. ELO gained per game

CONCLUSION :

The **Valorant Stats Analysis** project provides valuable insights into the game by analyzing a Kaggle dataset using statistical and machine learning techniques. Key findings include:

1. **Top Agents**: Identified the 10 most popular agents, reflecting current game meta.
2. **Predictive Modeling**: Linear regression uncovered relationships between performance metrics and factors like agent selection, aiding strategic planning.
3. **Decision Tree Insights**: Decision trees helped predict match outcomes by identifying key decision points.
4. **Clustering Analysis**: K-Means clustering revealed common playstyles and strategies.
5. **Anomaly Detection**: Outlier detection highlighted unusual performances, signaling unique strategies or anomalies.

The project sets a foundation for future work, such as applying deep learning, enabling real-time analysis, and developing a user-friendly interface for broader accessibility.