Here’s a **table** summarizing the **current capabilities**, **planned enhancements**, and **long-shot advancements** for the **Survivor Tool**. This will give you a clear roadmap of how each feature will evolve and improve the tool, along with key goals for each phase:

| **Feature** | **Current Capabilities** | **Enhancements (Next Steps)** | **Long-shot Advancements** | **Key Goals** | **How Enhancements Improve the Tool** |
| --- | --- | --- | --- | --- | --- |
| **Spot Value Calculation** | Calculates spot values based on **win probability**, **DVOA**, and **homefield advantage**. | Fine-tune DVOA scaling, adjust weighting for **negative DVOA values**, and handle **team fatigue**. | Introduce **team momentum** into spot value scoring based on recent performance or streaks. | **Optimizing pick strategy** using **real-time data**, and adjusting **spot values** based on team trends. | Enhances the accuracy of spot value predictions by accounting for **team performance trends**, **injuries**, and **weather**, improving the ability to make informed picks and avoid underperforming teams. |
| **DVOA Integration** | **DVOA** influences spot value by adjusting for offensive/defensive performance. | Refine the weighting for **TOT DVOA** and **defensive DVOA**. Apply more nuanced adjustments. | Full **API integration** to pull real-time **DVOA** updates each week. | **Refine performance evaluation** to give more weight to **defensive DVOA** and adjust based on **opponent strength**. | This **refines spot value scoring** to focus on both offense and defense, improving accuracy in predicting matchups and value spots based on team performance. |
| **Weekly Data Input** | Data for **win probabilities**, **DVOA**, **spot value**, and **team usage** are updated manually. | Implement **automated API pulls** for **win probabilities** and **injury data**. | Fully **automate weekly data inputs** for **injury**, **weather**, and **team usage**. | **Reduce manual input** and ensure **real-time updates** to allow better pick decisions. | Automates the data collection process, making the tool easier to maintain and more reliable, and improves decision-making through **real-time data** updates. |
| **Pick Evaluation** | Picks are evaluated based on **spot value** and **team performance**. | Introduce **confidence scoring** based on **historical success** and **spot value trends**. | Implement **machine learning** models to predict pick success, adjusting based on team trends and opponent behavior. | **Optimize weekly picks** using **confidence scoring** and **historical success** data, resulting in **better pick decisions** over time. | The **confidence scoring system** allows you to fine-tune **pick predictions** and focus on picks with high probabilities of success. |
| **Simulation & Back-Testing** | Basic **simulations** for future picks based on **spot value**. | Implement full **season-long simulations** to evaluate pick strategies over multiple weeks. | Use **historical data** and **team performance** to simulate future seasons with advanced **scoring models**. | **Refine the pick strategy** using **simulation data** for better **long-term decision-making** and **optimal pick flows**. | **Simulations** enable you to predict **optimal strategies** and test the system’s ability to make accurate picks in different scenarios. |
| **Real-Time Data Integration** | **Injury** and **weather data** are planned for future integration. | Integrate **Injury API**, **weather API**, and **market odds API** to adjust **spot values** in real-time. | Fully **automate injury and weather data processing** to influence the spot value **dynamically** during the season. | **Optimize picks** using **real-time data** to account for **injuries** and **weather conditions**, improving **weekly pick accuracy**. | **Real-time data** ensures that **spot values** adjust instantly based on **changing team conditions** like injuries or weather, improving **pick accuracy**. |
| **UI/Reporting Setup** | **CSV reports** for tracking picks and performance. | Build a **dashboard** to display **weekly picks**, **spot values**, and **pick success**. | Full **web-based dashboard** with interactive charts and data export features for **real-time analysis**. | Create **dynamic reports** and **dashboards** to help **visualize** data and track **team performance** over the season. | The **web dashboard** enhances usability, allowing you to **easily track pick performance**, spot value trends, and **adjust strategies** based on data-driven insights. |
| **Automation of Data Pulls** | **DVOA** and **win probabilities** are manually updated. | Automate **DVOA data pull** and **win probability data fetch** through APIs. | Fully automate data integration from **multiple sources** (e.g., **DVOA**, **weather**, **market odds**, **injuries**). | Ensure **automated updates** for **data accuracy** and reduce manual efforts. | **Automation** saves time, **improves data accuracy**, and makes the Survivor Tool more **efficient** and **reliable**, ensuring you're always working with the latest data. |

**Key Goals for the Enhancements:**

1. **Spot Value Optimization**: Enhancing spot value accuracy and making it responsive to real-time data (injuries, weather, etc.) will significantly improve pick decisions.
2. **Automation of Updates**: Reducing manual input for data and fully automating updates will make the tool more efficient and reduce errors.
3. **Real-Time Data Integration**: By integrating **real-time injury and weather data**, the tool will be able to make more dynamic and accurate pick evaluations.
4. **User Interface Improvements**: Moving toward a **web-based dashboard** will improve usability and provide clear insights into **pick performance**, **spot value trends**, and other key metrics.
5. **Long-Term Simulation Testing**: Advanced simulations based on historical data and season-long models will help improve decision-making strategies and allow for testing various pick flows and strategies.

**Implementation of Long-Shot Features to Improve the Product:**

* **Machine Learning for Pick Prediction**: Integrating machine learning can automate **pick strategy optimization**, adjusting to ongoing data and trends.
* **Real-Time Data Influence**: Using real-time injury, weather, and market odds to dynamically influence the **spot\_value\_score** will make the tool more responsive to **in-season developments**, improving pick accuracy.
* **Web Dashboard for Visualization**: A web dashboard will allow **real-time data tracking** and **visualization of pick performance**, making it easier to analyze the tool’s success and adjust strategies over time.

**Learning Python: Key Instructions for Your Development Journey**

Since you mentioned that you're **green at Python**, here’s what you can focus on as you continue building and improving your tool:

1. **Understand Basic Python Syntax**:
   * Ensure you have a strong grasp of Python fundamentals (loops, conditionals, functions, etc.) before diving deeper into advanced topics.
2. **Work with APIs**:
   * Understand how to **fetch data from APIs** (e.g., Odds API, Injury API) using Python libraries like **requests**. Learn how to parse JSON responses to extract useful data.
   * **Helpful Tutorial**: [Python API Handling with Requests](https://realpython.com/python-requests/)
3. **Data Handling with Pandas**:
   * Use **Pandas** to handle and manipulate CSV files. Learn how to **read**, **write**, and **update** data in CSVs, as this is core to your tool's functionality.
   * **Helpful Tutorial**: [Pandas for Beginners](https://realpython.com/pandas-python-explore-dataset/)
4. **Automating Tasks**:
   * Learn how to **automate scripts** using **cron jobs** (for scheduling scripts) or Python's schedule library for automatic updates and data pulling.
   * **Helpful Tutorial**: [Scheduling Python Jobs with schedule](https://realpython.com/python-schedule/)
5. **Simulation and Testing**:
   * Learn how to write test functions to simulate pick scenarios. Start with simple tests and then build toward more complex simulations.
   * **Helpful Tutorial**: [Simulating Random Events in Python](https://realpython.com/python-random/)
6. **Visualization for Reporting**:
   * As you work on creating reports and dashboards, learn how to use **Matplotlib** or **Seaborn** for generating graphs and charts.
   * **Helpful Tutorial**: [Matplotlib Basics](https://matplotlib.org/stable/tutorials/introductory/pyplot.html)

Let me know if you’d like more detailed instructions on any of these topics or if you need guidance on specific aspects of your tool!