## EDA in CS:GO

**1. Understanding Player Performance:**

* **Step:** Analyze data like Kills, Deaths, and Assists (KDA) for different ranks or regions.
* **Insight:** This reveals skill distribution across the player base. Are lower ranks struggling with a specific weapon type? Does KDA differ significantly between regions?
* **Action:** Game balance adjustments or tutorials targeted at specific weaknesses.

**2. Analyzing Weapon Balance:**

* **Step:** Explore metrics like weapon pick rates, kill counts per weapon, and first-shot kill percentages for different weapons.
* **Insight:** Are certain weapons dominating the meta? Do specific weapon types underperform at certain ranks? (e.g., shotguns in high ranks)
* **Action:** Weapon balance adjustments to promote a diverse and competitive environment.

**3. Identifying Map Trends:**

* **Step:** Analyze win rates, bomb plant/defuse success rates, and common player movement patterns on various maps.
* **Insight:** Do certain maps heavily favor one team (Terrorists or Counter-Terrorists)? Are specific bomb sites rarely used?
* **Action:** Map balancing tweaks or highlighting under-utilized strategies through in-game tutorials.

## Machine Learning

**1. Data Loading:**

* **Available here** [**https://github.com/pnxenopoulos/esta**](https://github.com/pnxenopoulos/esta)
* **Load either LAN or Online for pre-processing**
  + **The following Colab should be helpful:** https://colab.research.google.com/drive/1Oqgr4LT3d9pCW4vj4isyR1AfNGtY50sF?usp=sharing

**2. Pipeline for Train/Test Split (prefabs)**

* **Ensure Training data is small enough for model training within given timeframe of the lab**

**3. Model Training**

* **Only utilize sklearn-based models for training a classifier!**
* **Hyper-parameter tuning will be expansive. So either do it beforehand and set those parameters as default or enforce a pattern that does take that into account**
* **Objective: Students understand the process of training a model**

**4. Model Inference**

* **In case of evaluation, only evaluate overall accuracy**
* **Objective: Students should be able to report accuracy of the model.**
* **NOTE: Avoid showing metrics such as Precision and Recall as that would be hard for the school kids to understand**

**5. Single Test set observation prediction**

* **Randomly select a data-point in test set**
* **Pass the data-point in model for prediction**
* **Assess what variables were important in giving the prediction that was given**
* **Check Interpret package:** [**https://interpret.ml/docs/python/examples/explain-blackbox-classifiers.html**](https://interpret.ml/docs/python/examples/explain-blackbox-classifiers.html)
* **Objective: The students should be able to answer what features were important in giving prediction for that single point**

**6. Model Inperpretation**

* **Features model gives importance to:** [**https://interpret.ml/docs/python/examples/interpretable-classification.html**](https://interpret.ml/docs/python/examples/interpretable-classification.html)
* **Objective: The students should be able to answer what features does the model consider to be important.**