Capstone Project Proposal

**Title: Predicting Wine Quality Using Machine Learning**

**Problem Identification**

**Problem Statement** Wine quality is traditionally assessed by human tasters, which makes the process subjective, inconsistent, and time-consuming. The goal of this project is to build a machine learning model that can objectively predict the quality of wine based on its physicochemical properties. This will aid in faster, more consistent quality control processes in the wine industry.

**Context**

In the competitive wine industry, ensuring product quality is crucial for brand reputation and customer satisfaction. Vinho Verde, a well-known Portuguese wine, is evaluated by experts for quality, but such assessments can vary. By leveraging historical data and machine learning, this project aims to replicate or enhance expert evaluations in a consistent and scalable manner.

**Criteria for Success**

* **Achieve 90% alignment with expert tasters:** The predictive model should match or exceed the consistency of human tasters by correctly classifying wine quality within ±1 point of expert scores at least 90% of the time.
* **Top 3 model performance:** Identify and fine-tune the top three models with a focus on performance and interpretability, targeting classification accuracy above 75% or regression RMSE below 0.6.
* **Feature discovery:** Surface at least 3–5 key physicochemical properties that most influence perceived wine quality, providing actionable insights to winemakers.
* **Scalable prototype:** Deliver a reproducible, well-documented pipeline that can be easily adapted to similar quality control tasks in food or beverage industries.
* **Stakeholder communication:** Present findings in a visually intuitive format to support decisions by non-technical stakeholders in production and distribution.

**Scope of Solution Space**

* **Goal-driven modeling:** Explore both classification (e.g., low/medium/high quality) and regression (exact quality score) to determine which better aligns with stakeholder needs and interpretability.
* **Model candidates:** Begin with baseline models like Logistic/Linear Regression, progressing to more robust methods like Random Forest, XGBoost, and LightGBM. Neural networks will be explored if interpretability allows.
* **Smart handling of imbalanced data:** Apply techniques such as class reweighting, SMOTE, and stratified sampling to manage the skewed distribution of wine scores.
* **Feature engineering and enhancement:** Evaluate log transformations, polynomial features, and principal component analysis (PCA) to improve model signal.
* **Explainability-first approach:** Use SHAP values or tree-based feature importances to ensure the model provides clear rationale for predictions, critical for adoption by winemakers.

**Constraints**

* The dataset lacks domain-specific details like grape type, brand, or price due to privacy.
* The quality scores are imbalanced; most wines are rated average (score 5-6).
* Wine quality is inherently subjective, so some noise in the data may be irreducible.

**Stakeholders**

* **Wine Producers & Quality Control Teams:** Can use the model to flag low or high-quality batches early.
* **Distributors/Retailers:** May benefit from consistent, data-backed quality metrics for marketing or pricing decisions.
* **Consumers:** Indirectly benefit through more consistent product quality.
* **Data Scientists:** Gain a reusable framework for product quality prediction tasks in other industries.

**Data Sources**

* The dataset is publicly available from the UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/wine+quality
* It contains two datasets: red and white wine samples.
* Each sample includes 11 physicochemical properties (e.g., fixed acidity, pH, alcohol) and a quality score (0–10 scale).

**Approach Overview**

1. **Exploratory Data Analysis (EDA)**
   * Understand feature distributions, correlations, and outliers.
   * Assess class imbalance and decide on resampling techniques if needed.
2. **Preprocessing**
   * Scale features, encode targets (if classification), handle missing values (if any).
   * Outlier detection (e.g., IQR, Isolation Forest).
3. **Modeling**
   * Test both regression and classification frameworks.
   * Start with baseline models; then try tree-based methods and ensembles.
4. **Evaluation & Interpretation**
   * Use cross-validation and metrics appropriate for the task.
   * SHAP or feature importance analysis for interpretability.
5. **Deployment (Optional)**
   * A simple Streamlit or Flask-based UI where users can input wine attributes and get a predicted quality score.

**Deliverables**

* Jupyter Notebooks with complete EDA, modeling, and evaluation.
* A final slide deck summarizing findings, approach, and key results.
* A well-documented GitHub repository with code, README, and instructions.
* Optional: Web-based demo (Streamlit app).