Anxiety Prediction using Machine Learning

# 1. Dataset Loading

We began by loading a large dataset (anxiety.csv) containing over 10,000 responses to psychological questionnaires. The dataset includes responses from various age groups and features such as anxiety levels, satisfaction with life, social phobia, demographics, and personality indicators like narcissism.

df = pd.read\_csv('anxiety.csv', encoding='ISO-8859-1')

# 2. Data Cleaning

 **Initial Inspection:** We checked for missing values using df.isnull().sum() and found many columns with high percentages of missing entries.

 **Cleaning Column Names:** Non-ASCII characters and trailing spaces were stripped to ensure compatibility.

 **Dropping Unnecessary Columns:** Columns such as S. No., Timestamp, and highestleague were removed because they were irrelevant or too sparse.

 **Handling Missing Values:**

* **Numerical Columns:** Filled with the mean.
* **Categorical Columns:** Filled with the most frequent value (mode).

 **Further Cleaning:** We dropped columns with more than 50% missing data and removed rows with any remaining missing values to ensure clean input for modeling.

# 3. Data Visualization & Exploration

**Distribution Plot – Anxiety Scores (GAD\_T)**

sns.histplot(df['GAD\_T'], bins=20, kde=True)

Revealed a slightly right-skewed distribution, suggesting most individuals reported lower anxiety scores.

**Multi-Variable Histogram**

df[['GAD\_T', 'SWL\_T', 'SPIN\_T']].hist(...)

We visualized the distributions of anxiety (GAD\_T), life satisfaction (SWL\_T), and social phobia (SPIN\_T).

**Correlation Heatmap**

sns.heatmap(...)

Revealed moderate positive correlation between GAD\_T and SPIN\_T and a negative correlation with SWL\_T (life satisfaction).

**Boxplot: Age vs GAD\_T**

sns.boxplot(x='Age', y='GAD\_T', ...)

Showed that younger individuals tended to report higher anxiety scores on average.

# 4. Feature Encoding

# All categorical features were label-encoded to convert them into numeric format required by machine learning models. A dictionary of encoders was stored for reproducibility.5. Train-Test Split

* The target (GAD\_T) was binarized: scores ≥10 labeled as 1 (high anxiety), else 0.
* The data was split: 80% for training and 20% for testing.

X = df.drop(columns=['GAD\_T'])  
y = (df['GAD\_T'] >= 10).astype(int)  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2)

# 6. Data Normalization

Standardization was used to bring all features to a common scale (mean=0, std=1), which is especially important for models like Logistic Regression and SVM.

# 7. Linear Regression – Assumption Checks

Although the final task was classification, linear regression was used to evaluate assumptions:

* **Linearity:** Scatter plot of predicted vs actual showed a weak linear trend.
* **Homoscedasticity:** Residuals appeared randomly scattered around zero.
* **Normality:** Histogram of residuals was slightly skewed, suggesting mild violation of normality.

This confirmed that while linear regression could be informative, more robust classification models were needed.

# 8. Model Training and Evaluation

We trained Logistic Regression, Random Forest, XGBoost, and Voting Classifier models.

This ensemble boosted performance by averaging predictions of top 2 models based on AUC.

# 9. SMOTE for Imbalanced Data

# SMOTE was used to oversample the minority class (high anxiety), resulting in better model generalization and improved performance metrics.

# 10. ROC Curve Comparison

# A ROC curve was plotted for each model before and after applying SMOTE. Dashed lines represent models trained with SMOTE, showing notable performance improvement (e.g., AUC increased to 0.96 for XGBoost and Voting).

# 11. Final Model Summary

XGBoost and Voting Classifier achieved the highest AUC after applying SMOTE.

# 12. Streamlit Web App Deployment

The final models were deployed in a Streamlit web application that allows users to:

1. Upload their own data (CSV).
2. Choose a prediction model (XGBoost or Random Forest).
3. Receive anxiety predictions along with probability.

This enables real-world usability, making mental health predictions accessible even to non-technical users

# 13. Conclusion

This project demonstrates that machine learning can play a key role in predicting anxiety based on questionnaire data. By handling real-world data challenges (missing values, class imbalance), and deploying the results in a web app, we created a reliable, user-friendly mental health support tool.