Absolutely! **Feature-target association tests** are used in **diagnostic analytics** to determine how strongly each feature (independent variable) is associated with a target variable (dependent variable), often before building predictive models. These tests depend on the data types involved (categorical, numeric, ordinal) and the nature of the relationship you're exploring.

Below is a curated list of **commonly used feature-target association tests**, categorized by data type combinations, with the **purpose**, **test name**, and **Python library** for each.

---

1. Categorical Feature vs Categorical Target

Test	Purpose	Python Implementation
Chi-Squared Test	Tests independence between two categorical variables	`scipy.stats.chi2_contingency`
Fisher's Exact Test	Alternative to chi-squared for small sample sizes	`scipy.stats.fisher_exact`
Cramér's V	Measures strength of association (0–1 scale)	`pingouin.cramers_v` or custom
Theil's U (Uncertainty Coefficient)	Asymmetric measure of information gain	`dython.nominal.theils_u`
Mutual Information (Categorical)	Measures information shared between two variables	`sklearn.feature_selection.mutual_info_class`

---

2. Numeric Feature vs Categorical Target

(Assumes binary or multi-class target)

Test	Purpose	Python Implementation
ANOVA F-test	Compares means across groups (multi-class target)	`sklearn.feature_selection.f_classif`
Point-Biserial Correlation	Special case of Pearson's r for binary target	`scipy.stats.pointbiserialr`
T-test (independent samples)	Compare means between two classes	`scipy.stats.ttest_ind`
Kruskal-Wallis Test	Non-parametric version of ANOVA	`scipy.stats.kruskal`
Mutual Information (Numeric)	Measures non-linear dependence	`sklearn.feature_selection.mutual_info_classif`

---

3. Numeric Feature vs Numeric Target

(Continuous regression targets)

Test	Purpose	Python Implementation
Pearson Correlation	Measures linear association	`scipy.stats.pearsonr`
Spearman Rank Correlation	Measures monotonic relationships	`scipy.stats.spearmanr`

Test	Purpose	Python Implementation
Kendall's Tau	Measures ordinal association	`scipy.stats.kendalltau`
Mutual Information (Regressor)	Captures nonlinear relationships	`sklearn.feature_selection.mutual_info_regression`
Maximal Information Coefficient (MIC)	General association strength	`minepy` or `sklearn-contrib`

---

4. Categorical Feature vs Numeric Target

(Reverse of category vs numeric)

Test	Purpose	Python Implementation
ANOVA F-test	Tests if means differ by group	`statsmodels.formula.api.ols` + `.fit()` + `.anova_lm()`
T-test	For two-category variables	`scipy.stats.ttest_ind`
Eta Squared	Effect size for ANOVA	`pingouin.anova` or `custom`
Boxplots & Visual EDA	Helps understand distributional separation	`seaborn.boxplot`, `sns.violinplot`

Bonus: Model-Based Feature Importance

While not tests in a strict statistical sense, these are **very commonly used** in diagnostic analytics:

Test	Purpose	Python Implementation
Logistic Regression Coefficients	Linear relationship to odds	`sklearn.linear_model.LogisticRegression`
Random Forest Feature Importance	Nonlinear association importance	`sklearn.ensemble.RandomForestClassifier`
SHAP Values	Local/global interpretability	`shap`
Permutation Importance	Measures change in performance	`sklearn.inspection.permutation_importance`

Tips on Choosing a Test

- Use **Chi-Squared**, **Cramér's V**, or **Mutual Info** for categorical-to-categorical.
- Use **ANOVA**, **t-test**, or **Point-Biserial** for numeric-to-categorical.
- Use **Pearson/Spearman** for numeric-to-numeric.
- Use **Mutual Information** for detecting nonlinear relationships.
- Use **SHAP** or **Permutation Importance** for model-based insights.

Would you like a flowchart or cheat sheet to help choose the right test based on data types?