Below is a proposed software specification outlining a set of Python API functions. Each function is designed to support one or more of the instructions in semiconductor chip test data analytics. You may adjust parameters as needed. For clarity, each function is given a title (with a sequential number), a function name, a unique ID (based on Workflow ID and Instruction ID), a purpose description, a Python function header with a descriptive docstring, and a “Used For” list that cites the instruction(s) it serves.

**Function 1: Identify Significant Metrics**

**Function Name**: catalog\_significant\_etest\_metrics  
**ID**: 12\_7  
**Purpose**: Catalog site‐level E-test metrics that exhibit statistically significant separation between high‐yield and low‐yield lots.  
**Signature**:

def catalog\_significant\_etest\_metrics(data: pd.DataFrame, p\_value\_threshold: float) -> List[str]:

"""

Catalog site-level E-test metrics showing statistically significant separation

between high-yield and low-yield groups.

Parameters:

data (pd.DataFrame): Test data containing E-test metrics.

p\_value\_threshold (float): Significance level (e.g., p-value threshold).

Returns:

List[str]: A list of metric names that pass the significance threshold.

"""

**Used For**:

* Identify Significant Metrics – Catalog site-level E-test metrics that show statistically significant separation between the two groups.

**Function 2: Identify Specific Measurement Family Impact**

**Function Name**: compare\_measurement\_family\_failure\_counts  
**ID**: 18\_7  
**Purpose**: Compare failure counts for a specified measurement family between test program revisions to assess its impact on yield change.  
**Signature**:

def compare\_measurement\_family\_failure\_counts(data\_rev1: pd.DataFrame, data\_rev2: pd.DataFrame, measurement\_family: str) -> Dict[str, Any]:

"""

Compare failure counts for a measurement family between two revisions and

evaluate its impact on overall yield.

Parameters:

data\_rev1 (pd.DataFrame): Test data for revision 1.

data\_rev2 (pd.DataFrame): Test data for revision 2.

measurement\_family (str): The measurement family to compare.

Returns:

Dict[str, Any]: Results with comparison metrics and yield impact details.

"""

**Used For**:

* Identify Specific Measurement Family Impact – Compare failure counts between revisions and assess impact on yield.

**Function 3: Identify Outlier Wafer Candidates**

**Function Name**: identify\_outlier\_wafer\_candidates  
**ID**: 21\_6  
**Purpose**: Identify wafers exhibiting both significant E-test parameter drift and high voltage failure rates as potential candidates for a process shift.  
**Signature**:

def identify\_outlier\_wafer\_candidates(wafer\_data: pd.DataFrame, etest\_drift\_threshold: float, voltage\_failure\_threshold: float) -> List[int]:

"""

Identify wafers that show significant E-test drift along with high voltage failure rates.

Parameters:

wafer\_data (pd.DataFrame): Data containing wafer E-test and voltage metrics.

etest\_drift\_threshold (float): Threshold to define significant drift.

voltage\_failure\_threshold (float): Threshold for high voltage failure rates.

Returns:

List[int]: List of wafer IDs considered candidates for a process shift.

"""

**Used For**:

* Identify Outlier Wafer Candidates – Identify wafers with significant E-test drift and high voltage failure rates.

**Function 4: Identify Clusters with Low Yield**

**Function Name**: identify\_low\_yield\_clusters  
**ID**: 22\_4  
**Purpose**: Evaluate yield metrics for clusters and identify those consistently falling below a target yield threshold.  
**Signature**:

def identify\_low\_yield\_clusters(cluster\_data: pd.DataFrame, target\_yield: float) -> List[str]:

"""

Evaluate cluster yield metrics and identify clusters with yield below the target.

Parameters:

cluster\_data (pd.DataFrame): Data with cluster yield information.

target\_yield (float): The yield threshold.

Returns:

List[str]: A list of cluster identifiers with low yield.

"""

**Used For**:

* Identify Clusters with Low Yield – Identify clusters that consistently underperform relative to the target yield.

**Function 5: Identify High Variance Measurements**

**Function Name**: identify\_high\_variance\_measurements  
**ID**: 29\_4  
**Purpose**: Compare computed variances and flag measurements with significantly high variance.  
**Signature**:

def identify\_high\_variance\_measurements(measurement\_data: pd.DataFrame, variance\_threshold: float) -> List[str]:

"""

Compare measurement variances and flag those with high variance.

Parameters:

measurement\_data (pd.DataFrame): Data containing test measurement values.

variance\_threshold (float): Variance threshold to define “high variance.”

Returns:

List[str]: A list of measurement names that exceed the variance threshold.

"""

**Used For**:

* Identify High Variance Measurements – Flag measurements with high variance.

**Function 6: Identify Test Sites with Increased Failure Rates**

**Function Name**: identify\_failure\_rate\_test\_sites  
**ID**: 31\_7  
**Purpose**: Compare failure statistics across test sites to determine which sites have notably higher failure rates.  
**Signature**:

def identify\_failure\_rate\_test\_sites(site\_data: pd.DataFrame, failure\_rate\_threshold: float) -> List[str]:

"""

Identify test sites that exhibit failure rates exceeding a specified threshold.

Parameters:

site\_data (pd.DataFrame): Data with test site failure statistics.

failure\_rate\_threshold (float): The threshold for flagging high failure rates.

Returns:

List[str]: A list of test site identifiers with increased failure rates.

"""

**Used For**:

* Identify Test Sites with Increased Failure Rates – Compare failure statistics to flag problematic test sites.

**Function 7: Identify Yield Drop Lots**

**Function Name**: identify\_yield\_drop\_lots  
**ID**: 34\_6  
**Purpose**: Define a yield threshold and isolate lots that drop below that threshold.  
**Signature**:

def identify\_yield\_drop\_lots(lot\_data: pd.DataFrame, yield\_threshold: float) -> List[str]:

"""

Isolate lots with yield below a defined threshold.

Parameters:

lot\_data (pd.DataFrame): Data containing lot yield information.

yield\_threshold (float): The yield threshold for flagging lots.

Returns:

List[str]: A list of lot identifiers with yield drops.

"""

**Used For**:

* Identify Yield Drop Lots – Isolate lots with yield below the threshold.

**Function 8: Identify Key Test Bins for Test House A**

**Function Name**: identify\_key\_test\_bins\_test\_house\_a  
**ID**: 41\_6  
**Purpose**: Pinpoint test bins that consistently contribute most to failing dies in yield‐failing lots from Test House A.  
**Signature**:

def identify\_key\_test\_bins\_test\_house\_a(test\_data: pd.DataFrame) -> List[str]:

"""

Pinpoint key test bins that contribute to high failure counts in Test House A.

Parameters:

test\_data (pd.DataFrame): Data containing test bin failure counts and yield information.

Returns:

List[str]: A list of test bin identifiers.

"""

**Used For**:

* Identify Key Test Bins for Test House A – Highlight test bins driving failing dies in yield-failing lots.

**Function 9: Identify Sharp Yield Drops**

**Function Name**: identify\_sharp\_yield\_drops  
**ID**: 42\_5  
**Purpose**: Compare weekly yield averages to detect weeks with sharp yield drops using a defined threshold or anomaly detection.  
**Signature**:

def identify\_sharp\_yield\_drops(weekly\_yield: pd.DataFrame, drop\_threshold: float) -> List[str]:

"""

Identify weeks with significant yield drops based on a defined threshold.

Parameters:

weekly\_yield (pd.DataFrame): Weekly yield data.

drop\_threshold (float): The percentage or statistical threshold for a yield drop.

Returns:

List[str]: A list of week identifiers with sharp yield drops.

"""

**Used For**:

* Identify Sharp Yield Drops – Detect weeks with significant yield decreases.

**Function 10: Identify Failing Test Measurements**

**Function Name**: identify\_failing\_test\_measurements  
**ID**: 44\_6  
**Purpose**: Extract and analyze failing test measurement data from wafers with concentrated failures to determine common measurement families or tests driving the failures.  
**Signature**:

def identify\_failing\_test\_measurements(wafer\_data: pd.DataFrame) -> Dict[str, Any]:

"""

Analyze failing test measurement data to identify common failure drivers.

Parameters:

wafer\_data (pd.DataFrame): Data with detailed test measurement and failure information.

Returns:

Dict[str, Any]: Analysis results including common measurement families or tests.

"""

**Used For**:

* Identify Failing Test Measurements – Extract and analyze failing measurements from wafers with concentrated failures.

**Function 11: Identify Key Test Bins**

**Function Name**: identify\_key\_test\_bins  
**ID**: 54\_5  
**Purpose**: Filter and rank test bins by failure counts to highlight those with the highest yield impact.  
**Signature**:

def identify\_key\_test\_bins(bin\_data: pd.DataFrame, top\_n: int = 3) -> List[str]:

"""

Identify key test bins based on failure counts.

Parameters:

bin\_data (pd.DataFrame): Data containing test bin failure counts.

top\_n (int): Number of top bins to return (default is 3).

Returns:

List[str]: A list of top test bin identifiers.

"""

**Used For**:

* Identify Key Test Bins – Rank test bins by failure counts to determine their yield impact.

**Function 12: Identify Yield Drop Events**

**Function Name**: identify\_yield\_drop\_events  
**ID**: 57\_6  
**Purpose**: Flag instances where yield falls below a defined threshold.  
**Signature**:

def identify\_yield\_drop\_events(yield\_data: pd.DataFrame, threshold: float) -> List[str]:

"""

Flag instances where the yield drops below a defined threshold.

Parameters:

yield\_data (pd.DataFrame): Data with yield information.

threshold (float): Yield threshold for flagging.

Returns:

List[str]: A list of event or lot identifiers where yield is below the threshold.

"""

**Used For**:

* Identify Yield Drop Events – Flag instances of yield falling below the threshold.

**Function 13: Identify Failure-Causing Measurements**

**Function Name**: identify\_failure\_causing\_measurements  
**ID**: 76\_3  
**Purpose**: For each failed die, list the test measurements that reported a failure outcome (logical or parametric).  
**Signature**:

def identify\_failure\_causing\_measurements(die\_data: pd.DataFrame) -> Dict[int, List[str]]:

"""

List test measurements causing failures for each failed die.

Parameters:

die\_data (pd.DataFrame): Data with die test measurements and outcomes.

Returns:

Dict[int, List[str]]: A mapping from die ID to a list of failing test measurement names.

"""

**Used For**:

* Identify Failure-Causing Measurements – List measurements responsible for die failures.

**Function 14: Identify Failing Dies**

**Function Name**: identify\_failing\_dies  
**ID**: 79\_3  
**Purpose**: Determine which dies on a wafer have any failing test measurement.  
**Signature**:

def identify\_failing\_dies(wafer\_data: pd.DataFrame) -> List[int]:

"""

Identify dies with any failing test measurement.

Parameters:

wafer\_data (pd.DataFrame): Data containing test results for each die.

Returns:

List[int]: A list of die IDs that have failures.

"""

**Used For**:

* Identify Failing Dies – Determine failing dies on the wafer.

**Function 15: Map Failing Dies to Test Bins**

**Function Name**: map\_failing\_dies\_to\_test\_bins  
**ID**: 71\_2  
**Purpose**: Map each failing die to its corresponding test bin(s) based on the test measurement family responsible for the failure.  
**Signature**:

def map\_failing\_dies\_to\_test\_bins(die\_failure\_data: pd.DataFrame) -> Dict[int, List[str]]:

"""

Map failing dies to the test bins that correspond to the failure-inducing measurement families.

Parameters:

die\_failure\_data (pd.DataFrame): Data linking dies with test measurement families.

Returns:

Dict[int, List[str]]: A dictionary mapping die IDs to one or more test bin identifiers.

"""

**Used For**:

* Map Failing Dies to Test Bins – Associate each failing die with its test bin(s).

**Function 16: Select Top Three Test Programs**

**Function Name**: select\_top\_three\_test\_programs  
**ID**: 10\_6  
**Purpose**: Identify the three test programs with the highest failure contributions.  
**Signature**:

def select\_top\_three\_test\_programs(program\_data: pd.DataFrame) -> List[str]:

"""

Select the top three test programs with the highest failure contributions.

Parameters:

program\_data (pd.DataFrame): Data containing failure counts per test program.

Returns:

List[str]: A list of the top three test program identifiers.

"""

**Used For**:

* Select Top Three – Identify the top three test programs contributing to failures.

**Function 17: Select Top Three Test Bins**

**Function Name**: select\_top\_three\_test\_bins  
**ID**: 52\_4  
**Purpose**: Identify the top three test bins with the highest failure counts.  
**Signature**:

def select\_top\_three\_test\_bins(bin\_data: pd.DataFrame) -> List[str]:

"""

Identify the top three test bins based on failure counts.

Parameters:

bin\_data (pd.DataFrame): Data containing failure counts for test bins.

Returns:

List[str]: A list of the top three test bin identifiers.

"""

**Used For**:

* Select Top Three – Identify the top three test bins with highest failure counts.

**Function 18: Filter Low-Yield Wafers**

**Function Name**: filter\_low\_yield\_wafers  
**ID**: 35\_3  
**Purpose**: Identify and isolate wafers with a computed yield below the target of 95%.  
**Signature**:

def filter\_low\_yield\_wafers(wafer\_data: pd.DataFrame, target\_yield: float = 95.0) -> List[int]:

"""

Filter wafers with yield below the specified target.

Parameters:

wafer\_data (pd.DataFrame): Data containing wafer yield metrics.

target\_yield (float): Yield target (default is 95.0).

Returns:

List[int]: A list of wafer IDs with yield below the target.

"""

**Used For**:

* Filter Low-Yield Wafers – Identify wafers with yield below 95%.

**Function 19: Identify Lots for Additional Inspection**

**Function Name**: identify\_lots\_for\_additional\_inspection  
**ID**: 3\_8  
**Purpose**: Highlight lots that show significant E-test fluctuation trends or statistically significant parametric variation to recommend for additional inspection.  
**Signature**:

def identify\_lots\_for\_additional\_inspection(lot\_data: pd.DataFrame, etest\_fluctuation\_threshold: float, parametric\_variation\_threshold: float) -> List[str]:

"""

Identify lots that may require additional inspection based on E-test fluctuations

or significant parametric variation.

Parameters:

lot\_data (pd.DataFrame): Data containing lot yield and parametric details.

etest\_fluctuation\_threshold (float): Threshold for E-test fluctuation.

parametric\_variation\_threshold (float): Threshold for significant parametric variation.

Returns:

List[str]: A list of lot identifiers recommended for further inspection.

"""

**Used For**:

* Identify Lots for Additional Inspection – Highlight lots with significant E-test or parametric anomalies.

**Function 20: Identify Key Test Measurement Families**

**Function Name**: identify\_key\_test\_measurement\_families  
**ID**: 5\_6  
**Purpose**: Rank test measurement families based on aggregated failure counts to determine which are most frequently contributing to failures.  
**Signature**:

def identify\_key\_test\_measurement\_families(failure\_data: pd.DataFrame) -> List[str]:

"""

Rank test measurement families by their aggregated failure counts.

Parameters:

failure\_data (pd.DataFrame): Data with failure counts per measurement family.

Returns:

List[str]: A list of key test measurement families.

"""

**Used For**:

* Identify Key Families – Rank measurement families based on failure counts.

**Function 21: Identify Failing Outcomes**

**Function Name**: identify\_failing\_outcomes  
**ID**: 9\_3  
**Purpose**: Filter test records to identify failing outcomes for both logical tests (explicit 'fail') and parametric tests (values not meeting test limits).  
**Signature**:

def identify\_failing\_outcomes(test\_records: pd.DataFrame) -> pd.DataFrame:

"""

Identify records with failing outcomes for both logical and parametric tests.

Parameters:

test\_records (pd.DataFrame): Data containing test records.

Returns:

pd.DataFrame: A filtered DataFrame with failing test outcomes.

"""

**Used For**:

* Identify Failing Outcomes – Filter records for failing logical outcomes and parametric test limit violations.

**Function 22: Identify Significant PCM Parameters**

**Function Name**: identify\_significant\_pcm\_parameters  
**ID**: 13\_5  
**Purpose**: Isolate PCM parameters that show strong negative correlations with FT1 yield, indicating their predictive power for low yield.  
**Signature**:

def identify\_significant\_pcm\_parameters(pcm\_data: pd.DataFrame, ft1\_yield: pd.Series, correlation\_threshold: float) -> List[str]:

"""

Identify PCM parameters with strong negative correlations to FT1 yield.

Parameters:

pcm\_data (pd.DataFrame): Data containing PCM parameters.

ft1\_yield (pd.Series): Series of FT1 yield values.

correlation\_threshold (float): Minimum correlation magnitude for significance.

Returns:

List[str]: A list of significant PCM parameters.

"""

**Used For**:

* Identify Significant PCM Parameters – Isolate PCM parameters predictive of low FT1 yield.

**Function 23: Identify Key Test Measurements**

**Function Name**: identify\_key\_test\_measurements  
**ID**: 14\_5  
**Purpose**: Extract detailed test measurement data and calculate failure rates or deviations to determine which measurements are strongly correlated with yield differences.  
**Signature**:

def identify\_key\_test\_measurements(measurement\_data: pd.DataFrame) -> Dict[str, Any]:

"""

Extract test measurement data and compute metrics to identify key measurements

that correlate with yield differences.

Parameters:

measurement\_data (pd.DataFrame): Detailed test measurement records.

Returns:

Dict[str, Any]: Analysis results including failure rates and deviations.

"""

**Used For**:

* Identify Key Test Measurements – Determine measurements correlated with yield differences.

**Function 24: Identify Sudden Drops**

**Function Name**: identify\_sudden\_drops  
**ID**: 16\_4  
**Purpose**: Use a defined threshold or statistical method to flag weeks with sudden or significant drops in yield.  
**Signature**:

def identify\_sudden\_drops(weekly\_yield: pd.DataFrame, threshold: float, method: str = "statistical") -> List[str]:

"""

Identify weeks with sudden or significant drops in yield.

Parameters:

weekly\_yield (pd.DataFrame): Weekly yield metrics.

threshold (float): Yield drop threshold.

method (str): Detection method (default "statistical").

Returns:

List[str]: Week identifiers where sudden drops are observed.

"""

**Used For**:

* Identify Sudden Drops – Flag weeks with significant yield drops.

**Function 25: Identify Abnormal Patterns**

**Function Name**: identify\_abnormal\_patterns  
**ID**: 17\_4  
**Purpose**: Filter and label wafer maps that exhibit abnormal grid or checkerboard patterns when compared to a normal baseline.  
**Signature**:

def identify\_abnormal\_patterns(wafer\_map\_data: np.ndarray, pattern\_criteria: Dict[str, Any]) -> List[str]:

"""

Identify wafer maps with abnormal grid or checkerboard patterns.

Parameters:

wafer\_map\_data (np.ndarray): Image data representing wafer maps.

pattern\_criteria (Dict[str, Any]): Criteria to define abnormal patterns.

Returns:

List[str]: A list of wafer map identifiers flagged as abnormal.

"""

**Used For**:

* Identify Abnormal Patterns – Flag wafer maps with abnormal grid/checkerboard patterns.

**Function 26: Identify the Top Test Program**

**Function Name**: identify\_top\_test\_program  
**ID**: 24\_5  
**Purpose**: Determine the test program with the highest failure count from the most recent lot.  
**Signature**:

def identify\_top\_test\_program(program\_data: pd.DataFrame) -> str:

"""

Identify the test program that contributes the most to die failures.

Parameters:

program\_data (pd.DataFrame): Data containing test program failure counts.

Returns:

str: The identifier of the top test program.

"""

**Used For**:

* Identify the Top Test Program – Select the program with the highest failure count.

**Function 27: Identify Center-Fail Patterns**

**Function Name**: identify\_center\_fail\_patterns  
**ID**: 26\_2  
**Purpose**: Detect and flag wafers with failing dies predominantly in the center region using pattern recognition.  
**Signature**:

def identify\_center\_fail\_patterns(wafer\_map: np.ndarray, region: str = "center") -> List[str]:

"""

Detect wafers with failing dies predominantly in the center.

Parameters:

wafer\_map (np.ndarray): Image data for a wafer map.

region (str): Region to analyze (default "center").

Returns:

List[str]: A list of wafer identifiers with center-fail patterns.

"""

**Used For**:

* Identify Center-Fail Patterns – Detect wafers with central failure clusters.

**Function 28: Identify Consistent Load Board Outliers**

**Function Name**: identify\_consistent\_load\_board\_outliers  
**ID**: 33\_9  
**Purpose**: Highlight load boards that consistently underperform (or outperform) based on yield distribution analysis.  
**Signature**:

def identify\_consistent\_load\_board\_outliers(lb\_data: pd.DataFrame, yield\_data: pd.DataFrame) -> List[str]:

"""

Identify load boards that consistently show yield performance outliers.

Parameters:

lb\_data (pd.DataFrame): Data on load board performance.

yield\_data (pd.DataFrame): Yield distribution data.

Returns:

List[str]: A list of load board identifiers flagged as outliers.

"""

**Used For**:

* Identify Consistent Outliers – Highlight load boards with consistent performance anomalies.

**Function 29: Identify Low-Performing Lots**

**Function Name**: identify\_low\_performing\_lots  
**ID**: 39\_5  
**Purpose**: Define a yield threshold and flag lots that underperform (yield below the target level).  
**Signature**:

def identify\_low\_performing\_lots(lot\_data: pd.DataFrame, yield\_threshold: float) -> List[str]:

"""

Identify lots with yield below a defined threshold.

Parameters:

lot\_data (pd.DataFrame): Data containing lot yield information.

yield\_threshold (float): The yield threshold for flagging.

Returns:

List[str]: A list of underperforming lot identifiers.

"""

**Used For**:

* Identify Low-Performing Lots – Flag lots with yield below the target.

**Function 30: Identify Persistent Low-Performing Test Bins**

**Function Name**: identify\_persistent\_low\_performing\_test\_bins  
**ID**: 39\_7  
**Purpose**: Across all lots, identify test bins that consistently contribute a high percentage of failures, marking them as persistent low-performing.  
**Signature**:

def identify\_persistent\_low\_performing\_test\_bins(bin\_data: pd.DataFrame, failure\_pct\_threshold: float) -> List[str]:

"""

Identify persistent low-performing test bins based on failure percentages.

Parameters:

bin\_data (pd.DataFrame): Data containing test bin failure counts.

failure\_pct\_threshold (float): Failure percentage threshold.

Returns:

List[str]: A list of test bin identifiers flagged as persistently low-performing.

"""

**Used For**:

* Persistent Low-Performing Test Bins Analysis – Identify bins with consistently high failure percentages.

**Function 31: Detect Recurring Spatial Patterns**

**Function Name**: detect\_recurring\_spatial\_patterns  
**ID**: 27\_5  
**Purpose**: Apply image processing and pattern recognition to identify recurring spatial patterns (such as clustering or grid patterns) in wafer maps.  
**Signature**:

def detect\_recurring\_spatial\_patterns(wafer\_images: List[np.ndarray], pattern\_params: Dict[str, Any]) -> List[str]:

"""

Detect recurring spatial patterns in wafer maps.

Parameters:

wafer\_images (List[np.ndarray]): A list of wafer map images.

pattern\_params (Dict[str, Any]): Parameters and criteria for pattern recognition.

Returns:

List[str]: A list of identifiers for wafer maps with recurring patterns.

"""

**Used For**:

* Detect Recurring Spatial Patterns – Identify recurring spatial patterns in wafer maps.

**Function 32: Recognize Wafer Map Patterns**

**Function Name**: recognize\_wafer\_map\_patterns  
**ID**: 1\_5  
**Purpose**: Apply pattern recognition techniques to identify known wafer map patterns (e.g., center clusters, edge-ring patterns).  
**Signature**:

def recognize\_wafer\_map\_patterns(wafer\_map: np.ndarray, known\_patterns: List[str]) -> List[str]:

"""

Recognize known patterns in a wafer map using pattern recognition techniques.

Parameters:

wafer\_map (np.ndarray): Image data of the wafer map.

known\_patterns (List[str]): A list of pattern descriptors to match.

Returns:

List[str]: Identified pattern names present in the wafer map.

"""

**Used For**:

* Pattern Recognition – Identify known wafer map patterns.

**Function 33: Detect Grid Pattern Failures**

**Function Name**: detect\_grid\_pattern\_failures  
**ID**: 10\_3  
**Purpose**: Apply image processing and pattern recognition to detect grid-like failure patterns in wafer maps.  
**Signature**:

def detect\_grid\_pattern\_failures(wafer\_map: np.ndarray) -> List[str]:

"""

Detect grid-like failure patterns in a wafer map.

Parameters:

wafer\_map (np.ndarray): Image data representing a wafer map.

Returns:

List[str]: A list of identifiers for detected grid-like failure patterns.

"""

**Used For**:

* Grid Pattern Detection – Identify grid-like failure patterns in wafer maps.

**Function 34: Isolate High-Correlation Wafers**

**Function Name**: isolate\_high\_correlation\_wafers  
**ID**: 13\_6  
**Purpose**: For significant PCM parameters, identify and rank wafers where these parameters deviate most, marking them as high-correlation wafers.  
**Signature**:

def isolate\_high\_correlation\_wafers(wafer\_data: pd.DataFrame, pcm\_parameters: List[str], deviation\_threshold: float) -> List[int]:

"""

Identify and rank wafers with high deviations in significant PCM parameters.

Parameters:

wafer\_data (pd.DataFrame): Data containing wafer measurements.

pcm\_parameters (List[str]): List of significant PCM parameter names.

deviation\_threshold (float): Deviation threshold for high correlation.

Returns:

List[int]: A list of wafer IDs with high-correlation deviations.

"""

**Used For**:

* Isolate High-Correlation Wafers – Rank wafers based on deviations in PCM parameters.

**Function 35: Filter Low-Yield Wafers (90% Threshold)**

**Function Name**: filter\_low\_yield\_wafers\_90  
**ID**: 23\_4  
**Purpose**: Identify wafers with yield below the threshold of 90%.  
**Signature**:

def filter\_low\_yield\_wafers\_90(wafer\_data: pd.DataFrame, yield\_threshold: float = 90.0) -> List[int]:

"""

Filter wafers with yield below the specified threshold (default is 90%).

Parameters:

wafer\_data (pd.DataFrame): Data containing wafer yield metrics.

yield\_threshold (float): Yield threshold (default 90.0).

Returns:

List[int]: A list of wafer IDs with yield below the threshold.

"""

**Used For**:

* Filter Low-Yield Wafers – Identify wafers where the yield is below 90%.

This specification provides a high‐level design for each Python API function needed to implement the required semiconductor chip test data analytics instructions. Adjust parameter types, thresholds, and return types as appropriate for your implementation and data structures.