Calls get_random_number_range int DataFrame = pd.DataFrame NpArray = np.ndarray Randomizer <<Abstract>> AutoMachineLearning get_random_number_range_int(start: int, end: int, step: int) -> int # random state: int Uses Static Method # _n_folds_validation: int # shuffle data: bool + <<pre>+ <<pre>property>> estimator: Any + __init__(self, n_folds_validation: int, shuffle_data: bool, max rand: int) -> None + fit model(self, x_train: DataFrame, y_train: NpArray) AutoExecutioner Uses + predict model(self, x test: DataFrame) -> tuple # auto ml: AutoMachineLearning + __init__(self, auto_ml: AutoMachineLearning) -> None + train model(self, df: DataFrame, size : float = 0.0) -> None + score_model(self, test: DataFrame, expected: tuple) -> float Implements + get model(self) -> str JarAutoML Uses static method + __init__(self, n_folds_validation: int, shuffle_data: bool, max_rand: int) -> None + fit_model(self, x_train: DataFrame, y_train: NpArray) -> None SplitterReturner + predict_model(self, x_test: DataFrame) -> tuple + train_test_split (x: DataFrame, y: NpArray, size: float) -> tuple + split_x_y_from_df(df: DataFrame) -> tuple