FIRST RECIPE
1st Prompt
Accuracy 9/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Pie chart used to show class imbalance in the `target` variable
Sampling type	Stratified	`train_test_split(..., stratify=y)` ensures balanced train/test split
Outliers removal	No	No filtering or conditions applied to remove outliers
Check for duplicates	No	No use of `duplicated()` or similar functions in code
Imputation of missing values	No	Missing values converted to `'NaN'` string for encoding; no imputation
Drop columns	Yes	`enrollee_id` dropped before modeling
Encoding	Label Encoding	Applied via custom `MultiColumnLabelEncoder` class
Create new columns	No	No new column creation observed
Feature selection	Yes	Implicit: SHAP and feature importance analysis used for interpretability
Data scaling / standardisation	No	Not applied—LightGBM does not require feature scaling
Hyperparameter tuning	Yes	Custom `params` dictionary used with LightGBM

2nd Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	`.value_counts()` on `target` to examine class distribution
Sampling type	Stratified	`train_test_split` with `stratify=y` used

Outliers removal	No	No filtering or condition-based exclusion of outliers
Check for duplicates	No	No check or removal of duplicates
Imputation of missing values	Replace with text	`NaN` strings injected into missing values for label encoding
Drop columns	Yes	`target` and `enrollee_id` columns removed before training
Encoding	Label Encoding	`MultiColumnLabelEncoder` with `LabelEncoder` per column
Create new columns	No	No new features created from external or independent logic
Feature selection	Yes	Used SHAP and LightGBM's built-in feature importance plots
Data scaling or standardisation	No	No use of `StandardScaler`, `MinMaxScaler`, etc.
Hyperparameter tuning	No	Manual hyperparameters provided without systematic tuning

3rd Prompt
Accuracy 10/11

Step	Technique Used	Code Snippet
Check for balanced data	Yes	`fig = px.pie(aug_train['target'].value_counts(), ...)`
Sampling type	Stratified Sampling	`train_test_split(..., stratify=y, ...)`
Outlier removal	No	*Not found*
Check for duplicates	No	*Not found*
Imputation of missing values	Handled implicitly via label encoding	`output[col] = output[col].fillna('NaN')`

Drop columns	Yes	`X_aug_train_transform.drop('enrollee_id',axis = 'columns')`
Encoding	Label Encoding	`LabelEncoder().fit_transform(output[col])` via `MultiColumnLabelEncoder`
Create new columns	No	*Not found*
Feature selection	Yes	`lgb.plot_importance(lgbm)` and `shap.summary_plot(...)` show selected features' importance
Standardization	No	*Not found (no scaling)*
Hyperparameter tuning	Yes (manual tuning)	`'params = {...}'` block before `lgb.train(...)`

4th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	`px.pie` used on `target` column to inspect class distribution
Sampling type	Stratified	`train_test_split(..., stratify=y)` used
Outliers removal	No	No filtering/removal logic found
Check for duplicates	No	No call to `duplicated()` or `drop_duplicates()`
Imputation of missing values	replace with text	Missing values filled with `'NaN'` string before Label Encoding
Drop columns	Yes	`'target'` and `'enrollee_id'` dropped and not reused in model features
Encoding	Label Encoder	Applied via custom `MultiColumnLabelEncoder` on object columns

Create new columns	No	All added columns are derived (e.g., from existing or label encoding)
Feature selection	Yes	Feature importance plotted, and only selected features passed to LightGBM
Data scaling/standardisation	No	No scaler (StandardScaler/MinMax/Robust/etc.) used
Hyperparameter tuning	No	Params manually set; no GridSearch/RandomSearch/Optuna used

5th Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Target distribution explicitly visualized using pie chart (`value_counts`)
Sampling type	Stratified	`train_test_split(..., stratify=y)` used
Outliers removal	No	No evidence of removing outliers
Check for duplicates	No	No `.duplicated()` or `drop_duplicates()` used
Imputation of missing values	replace with text	NaN values replaced with string 'NaN' before LabelEncoding
Drop columns	No	Only `target` and `enrollee_id` dropped for modeling, not true dropping
Encoding	Label Encoder	Custom `MultiColumnLabelEncoder` using `LabelEncoder`
Create new columns	No	No new feature creation beyond encoding and splitting
Feature selection	Yes	`enrollee_id` dropped post-EDA; model-based importance via SHAP plotted
Data scaling/standardisation	No	No scaler (e.g., StandardScaler/MinMax) used
Hyperparameter tuning	No	Manual parameters passed to LightGBM; no GridSearchCV or similar used

Ground Truth

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Target distribution plotted via `px.pie(aug_train['target'].value_counts())`; notebook notes class imbalance.
Sampling type	Stratified	`train_test_split(..., test_size=0.2, shuffle=True, stratify=y, random_state=1301)`.
Outliers removal	No	Only boxplots for variables (e.g., `city_development_index`, `training_hours`); no filtering applied.
Check for duplicates	No	No use of `.duplicated()` / `.drop_duplicates()` in the workflow.
Imputation of missing values	Mixture of imputation techniques	Categorical columns filled with text placeholder `'NaN'` inside `MultiColumnLabelEncoder` then label-encoded; numeric missings left as-is (LightGBM handles missing).
Drop columns	Yes	`enrollee_id` dropped from features before modelling/prediction; not reused elsewhere.
Encoding	Label Encoder	Custom `MultiColumnLabelEncoder` applies `LabelEncoder()` per categorical column (after `fillna('NaN')`).
Create new columns	No	No new features created (no one-hot/dummy expansion; only in-place label encoding).
Feature selection	Yes	Feature importance/SHAP plotted for interpretation; no correlation/model-based dropping performed.
Data scaling/standardisation	No	No scaler (`StandardScaler`, `MinMaxScaler`, etc.) applied.
Hyperparameter tuning	No	LightGBM parameters set manually; no `GridSearchCV`/`RandomizedSearchCV` or similar systematic search (early stopping only).

SECOND RECIPE
1st Prompt
Accuracy 9/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No pie chart or value count shown for target class imbalance
Sampling type	No	No train/test split was done manually—only fitting directly
Outliers removal	No	No conditions or IQR/z-score based removals seen
Check for duplicates	Yes	`train.duplicated()` used
Imputation of missing values	Multivariate	`IterativeImputer` (MICE) used for imputing several features
Drop columns	No	No feature dropped
Encoding	Mixture of encoding	Manual `map()` for ordinal + nominal features; label encoding not used
Create new columns	No	No new columns created
Feature selection	Yes	Feature importances plotted from Random Forest
Data scaling or standardisation	Yes	`StandardScaler()` used via pipeline
Hyperparameter tuning	Yes	Tuned `RandomForestClassifier` and `DecisionTreeClassifier` params manually

2nd Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No explicit value_counts or class distribution checks
Sampling type	No	No `train_test_split()` or sampling method applied

Outliers removal	No	No row filtering based on numeric ranges or z-scores
Check for duplicates	Yes	`train.duplicated().sum()` used to check for duplicates
Imputation of missing values	Multivariate	Used `IterativeImputer` (MICE) on many variables
Drop columns	No	No column dropped without replacement
Encoding	Mixture of encoding	Used `.map()` for ordinal encoding + `Categorical().codes` for nominal
Create new columns	No	No column added independent of others (Prompt-2 and 3 rule applied)
Feature selection	Yes	Feature importance visualised from Random Forest
Scaling or standardisation	Yes	Used `StandardScaler()` in the pipeline
Hyperparameter tuning	No	RF/DT parameters manually selected, no systematic search

3rd Prompt
Accuracy 9/11

Step	Used ?	Technique (if applicable)	Code Snippet (if applicable)
1. Check for balance	No	–	–
2. Sampling	No	–	–
3. Outlier removal	No	–	–
4. Check for duplicates	Yes	`duplicated()`	`train.duplicated().sum()`
5. Imputation of missing values	Yes	Multivariate (MICE)	`IterativeImputer().fit_transform(...)`
6. Drop columns	No	–	–
7. Encoding	Yes	Manual (Ordinal + Nominal)	`map({...})` across columns

8. Create new columns	No	–	–
9. Feature selection	Yes	RandomForest Feature Importance	`feature_importances_`
10. Standardization	Yes	StandardScaler	`Pipeline(..., StandardScaler())`
11. Hyperparameter tuning	Yes	Manual	`RandomForestClassifier(...params...)`

4th Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No explicit check for class distribution using pie or countplot
Sampling type	None	No `train_test_split` or CV strategy shown
Outliers removal	No	No outlier filtering or removal logic observed
Check for duplicates	Yes	`train.duplicated()` used and counted
Imputation of missing values	multivariate	Used `IterativeImputer` (MICE) for multiple columns in train and test sets
Drop columns	No	All features reused or transformed; none dropped irreversibly
Encoding	mixture of encoding	Used manual mapping (ordinal & nominal) + `Categorical(...).codes`
Create new columns	No	Only mappings applied, no new information introduced
Feature selection	Yes	Model-based importance visualised for Random Forest
Data scaling/standardisation	Yes	Used `StandardScaler()` inside pipelines
Hyperparameter tuning	No	Parameters manually defined for RF and DT, no systematic search

5th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No `value_counts()` or plot of target distribution detected
Sampling type	None	No train-test split or resampling used
Outliers removal	No	No code for identifying or removing outliers
Check for duplicates	Yes	`train.duplicated()` checked and summed
Imputation of missing values	multivariate	Missing values imputed using `IterativeImputer` (MICE)
Drop columns	No	No column dropped without replacement or derivation
Encoding	mixture of encoding	Manual `map()` for ordinal features + `Categorical(...).codes` used
Create new columns	No	All encodings are transformations of existing columns
Feature selection	No	No columns dropped.
Data scaling/standardisation	Yes	`StandardScaler()` used in both model pipelines
Hyperparameter tuning	No	Fixed parameters passed manually to both RF and DT models

Ground Truth

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No class distribution check for `target` (no `value_counts()`/plots).
Sampling type	No	No train–test split or resampling; uses provided Kaggle train/test sets.

Outliers removal	No	No IQR/quantile/rule-based filtering applied.
Check for duplicates	Yes	Duplicates inspected via `train.duplicated().sum()`.
Imputation of missing values	Multivariate	MICE via `IterativeImputer` on mapped numeric categories (train & test).
Drop columns	No	No columns fully removed from analytics (e.g., `enrollee_id` retained for submission).
Encoding	Mixture	Ordinal/nominal mapped to integers; `pd.Categorical(...).codes` for `city`, `relevent_experience`.
Create new columns	No	No truly new features; mappings performed in place (no one-hot expansion).
Feature selection	No	Feature importances plotted, but no correlation/model-based drops performed.
Data scaling/standardisation	Yes	`StandardScaler()` inside pipelines for RF and DT.
Hyperparameter tuning	No	RF/DT hyperparameters set manually; no Grid/Random/Optuna search.

THIRD RECIPE
1st Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No value counts or plots of `target` distribution
Sampling type	Random	`train_test_split(..., test_size=0.3)` used without stratification
Outliers removal	No	No outlier detection or filtering applied
Check for duplicates	No	No use of `.duplicated()` or `.drop_duplicates()`
Imputation of missing values	Mixture of imputation	Used `fillna()` with values like `"0ther"` or inferred category

Drop columns	Yes	`city_development_index` and `target` dropped before modeling
Encoding	Label Encoding + Mapping	Used `LabelEncoder` and custom `map()` encodings
Create new columns	No	No derived or engineered features created
Feature selection	Yes	`modelfit()` displays feature importances from Gradient Boosting
Data scaling / standardisation	No	No use of `StandardScaler` or similar scaling methods
Hyperparameter tuning	Yes	Extensive `GridSearchCV` used for multiple hyperparameters in GBM

2nd Prompt
Accuracy 9/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No `.value_counts()` or class distribution visualisation for target
Sampling type	Random	Used `train_test_split()` without stratify
Outliers removal	No	No filtering applied on numeric values
Check for duplicates	Yes	Used `train.duplicated().sum()`
Imputation of missing values	Mixture of techniques	Used fillna with constant values (e.g., 0, "Other") and label-specific strategies
Drop columns	Yes	Dropped `city_development_index` based on correlation with `city`
Encoding	Mixture of encoding	Used `.map()` and `LabelEncoder` depending on column type
Create new columns	No	Only modified or replaced values within existing columns
Feature selection	Yes	Feature importances from `GradientBoostingClassifier`

Scaling or standardisation	No	Not used in the final model (unlike RF/DT pipeline from earlier snippets)
Hyperparameter tuning	Yes	Used `GridSearchCV` across multiple parameter sets with cross-validation

3rd Prompt
Accuracy 10/11

Step	Used ?	Technique (if applicable)	Code Snippet / Description
Check for balance	No	–	–
Sampling	Yes	Random	`train_test_split(..., random_state=42)`
Outlier removal	No	–	–
Check for duplicates	No	–	–
Imputation of missing values	Yes	Summary stats / placeholder	`.fillna('Other')`, `.fillna(0)`, `.fillna(8)`
Drop columns	Yes	Dropped `target`, `city_development_index`	`drop(labels=..., axis=1)`
Encoding	Yes	Label Encoding + map	`LabelEncoder().fit_transform(...)`, `.map({...})`
Create new columns	No	–	–
Feature selection	Yes	Gradient Boost Feature Importance	`alg.feature_importances_`
Standardization	No	–	–
Hyperparameter tuning	Yes	GridSearchCV	`GridSearchCV(...).fit(...)`

4th Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No countplot or piechart of target distribution seen
Sampling type	Random	`train_test_split` without stratification used
Outliers removal	No	No outlier filtering logic present
Check for duplicates	No	No use of `duplicated()` or `drop_duplicates()`
Imputation of missing values	mixture of imputation techniques	Used `fillna` with values (e.g., "Other", 0, mode) + label-specific handling
Drop columns	Yes	Dropped `city_development_index` without reuse
Encoding	mixture of encoding	Used both `LabelEncoder` and `map()` based ordinal encoding
Create new columns	No	Only transformations applied; no new columns added
Feature selection	No	No feature pruning or selection logic applied before model training
Data scaling/standardisation	No	No scaler (StandardScaler, MinMax, etc.) used
Hyperparameter tuning	Yes	Extensive use of `GridSearchCV` with multiple param grids for GBM

5th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No value_counts or chart of target distribution seen
Sampling type	Random	`train_test_split` used without stratify
Outliers removal	No	No method applied for outlier handling
Check for duplicates	No	No `.duplicated()` or equivalent check seen

Imputation of missing values	mixture of imputation techniques	Used both fillna (with specific values) and manual mapping for categorical
Drop columns	No	`city_development_index` dropped post-EDA (→ not counted here)
Encoding	mixture of encoding	Used LabelEncoder and manual mapping
Create new columns	No	No column was added from external data
Feature selection	Yes	Dropped `city_development_index` post-EDA; model importance also shown
Data scaling/standardisation	No	No scaling (e.g., StandardScaler) used anywhere
Hyperparameter tuning	Yes	Extensive `GridSearchCV` used for GradientBoostingClassifier

Ground Truth

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No explicit class-distribution check for `target` (no `value_counts()` / countplot of `target`).
Sampling type	Random	`train_test_split(train, pred, test_size=0.3, random_state=42)` without `stratify`.
Outliers removal	No	No IQR/quantile/rule-based filtering present.
Check for duplicates	No	No use of `.duplicated()` / `.drop_duplicates()`.
Imputation of missing values	Mixture of imputation techniques	Filled categorical missings with labels (`'Other'`, `'no_enrollment'`, `'never'`), and numeric placeholders for encoded categories (`0`, `8`, etc.).
Drop columns	No	No feature dropped prior to EDA; subsequent drop handled under Feature selection (e.g., `city_development_index`).

Encoding	Mixture of Encoding	Label/ordinal mappings and `LabelEncoder` applied in-place to multiple categoricals (e.g., `gender`, `enrolled_university`, `education_level`, `major_discipline`, `relevent_experience`, `company_size`, `company_type`, `last_new_job`).
Create new columns	No	Only in-place recoding/mapping; no genuinely new features created.
Feature selection	Yes	Dropped `city_development_index` after visual inspection; also excluded `enrollee_id` from `predictors` used for modelling. (Drop-after-visualisation counts as feature selection)
Data scaling/standardisation	No	No scaler (`StandardScaler`, `MinMaxScaler`, etc.) applied.
Hyperparameter tuning	Yes	Multiple `GridSearchCV` runs for Gradient Boosting (`n_estimators`, `max_depth`, `min_samples_split`, `min_samples_leaf`, `max_features`, `subsample`).

FOURTH RECIPE
1st Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	`value_counts(normalize=True)` used on `target`; identified imbalance
Sampling type	Stratified	`StratifiedKFold(n_splits=5, shuffle=True, random_state=42)` used
Outliers removal	No	No detection or filtering of outliers observed
Check for duplicates	No	Not performed in code
Imputation of missing values	Category-based (custom)	`fillna("NONE_colname")` used as per Abhishek Thakur's book

Drop columns	Yes	`enrollee_id` removed; unused columns dropped during encoding
Encoding	Mixture of encoding	OHE for low cardinality; frequency encoding for high cardinality
Create new columns	Yes	One-hot encoded columns created from original categorical variables
Feature selection	Yes	Recursive Feature Elimination with CV (RFECV) using Random Forest
Data scaling / standardisation	No	No scaling or standardisation used
Hyperparameter tuning	No	No GridSearchCV or parameter tuning in this notebook

2nd Prompt
Accuracy 9/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Used `train['target'].value_counts(normalize=True)` to check imbalance
Sampling type	Stratified	Used `StratifiedKFold` for cross-validation
Outliers removal	No	No filters, capping, or outlier handling shown
Check for duplicates	No	No `.duplicated()` or `.drop_duplicates()` used
Imputation of missing values	Replace with text	Used `.fillna('NONE_'+colname)` for categorical columns
Drop columns	No	No columns dropped
Encoding	Mixture of encoding	Used `pd.get_dummies()` and `frequency encoding` based on cardinality
Create new columns	Yes	Created dummy columns using one-hot encoding for low-cardinality columns
Feature selection	Yes	Used `RFECV` for recursive feature elimination

Scaling or standardisation	No	No scaling method used
Hyperparameter tuning	No	Used default/random parameters; no grid/random search loop shown

3rd Prompt
Accuracy 9/11

Step	Used ?	Technique (if applicable)	Code Snippet / Description
1. Check for balance	Yes	Value counts	`value_counts(normalize=True)`
2. Sampling	Yes	Stratified K-Fold	`StratifiedKFold(...)`
3. Outlier removal	No	–	–
4. Check for duplicates	No	–	–
5. Imputation of missing values	Yes	Custom fill with `"NONE_"` prefix	`fillna(f'NONE_{n}')`
6. Drop columns	Yes	Dropped OHE source + target	`drop(c, axis=1)`
7. Encoding	Yes	One-Hot + Frequency Encoding	`get_dummies()`, `.map(freq)`
8. Create new columns	Yes	OHE columns	`concat([data, temp], axis=1)`
9. Feature selection	Yes	RFECV	`RFECV(...).fit(...)`
10. Standardization	No	–	–
11. Hyperparameter tuning	No	–	–

4th Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details

Check for balanced data	Yes	`train['target'].value_counts(normalize=True)` shows imbalance (75/25)
Sampling type	Stratified	Used `StratifiedKFold` for cross-validation
Outliers removal	No	No explicit outlier detection or filtering
Check for duplicates	No	No use of `duplicated()` or `drop_duplicates()`
Imputation of missing values	replace with text	Missing categorical values replaced with `NONE_<col>` as separate category
Drop columns	No	Original columns dropped only after encoding (OHE/freq encoding) → replacement
Encoding	mixture of encoding	One-hot encoding for low cardinality; frequency encoding for high cardinality
Create new columns	No	Only OHE/frequency encodings derived from existing columns
Feature selection	Yes	Applied `RFECV` with Random Forest for recursive feature elimination
Data scaling/standardisation	No	No StandardScaler/MinMax/RobustScaler applied
Hyperparameter tuning	No	RandomForest used with set parameters, no grid/random search

5th Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Target distribution checked using `value_counts(normalize=True)`
Sampling type	Stratified	`StratifiedKFold` used for CV and model training
Outliers removal	No	No outlier removal applied
Check for duplicates	No	No check for `.duplicated()` or equivalent

Imputation of missing values	replace with text	Nulls imputed using `"NONE_" + column` string for object columns
Drop columns	No	No dropped columns without replacement
Encoding	mixture of encoding	One-hot encoding for low-cardinality, frequency encoding for high
Create new columns	No	New features were derived only via encoding, no external info used
Feature selection	Yes	`RFECV` used for recursive feature elimination
Data scaling/standardisation	No	No scaler (e.g., StandardScaler) used
Hyperparameter tuning	No	Model used with fixed parameters; no GridSearchCV or similar

Ground Truth

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	`train['target'].value_counts(normalize=True)` used; markdown notes 75% vs 25% imbalance.
Sampling type	Stratified	`StratifiedKFold(n_splits=5, shuffle=True, random_state=42)` used for CV (no separate hold-out split).
Outliers removal	No	No IQR/quantile/rule-based filtering applied.
Check for duplicates	No	No `.duplicated()` / `.drop_duplicates()` present.
Imputation of missing values	Replace with text	For object cols (except `city`, `relevent_experience`): `fillna(f'NONE_{col}')`. Numeric cols have no NA.
Drop columns	Yes	`enrollee_id` excluded from modelling via `required_cols` (not reused).

Encoding	Mixture	One-hot (`pd.get_dummies`) for low-cardinality cats (`relevent_experience`, `enrolled_university`, `gender`); frequency encoding for high-cardinality cats within CV (`freq_encode`).
Create new columns	No	One-hot columns are derived from existing features; not counted as “new”.
Feature selection	Yes	`RFECV` with `RandomForestClassifier` selects features via CV.
Data scaling/standardisation	No	No scaler (`StandardScaler`, `MinMaxScaler`, etc.) used.
Hyperparameter tuning	No	RF parameters fixed (e.g., `n_estimators=1000`); no Grid/Random/Optuna search.

FIFTH RECIPE
1st Prompt
Accuracy 8/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	`df['target'].value_counts()` shows imbalance (75% class 0, 25% class 1)
Sampling type	Random	`train_test_split(..., random_state=42)` used without stratification
Outliers removal	No	No explicit outlier filtering or z-score applied
Check for duplicates	No	Not performed in code
Imputation of missing values	Mixture	Custom classes: `CategoricalImputer`, `NumericImputer`
Drop columns	No	No columns dropped; all passed through pipelines
Encoding	One-hot encoding	`ColumnTransformer` with `OneHotEncoder(drop='first')`
Create new columns	Yes	New feature `experience_unknown` created via custom logic

Feature selection	No	All features used; no RFE or importance ranking implemented
Data scaling / standardisation	Yes	`StandardScaler` used for numeric features
Hyperparameter tuning	Yes	`GridSearchCV` used for tuning `n_estimators` and `max_depth` in XGB

2nd Prompt
Accuracy 9/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	`df['target'].value_counts()` used to assess imbalance
Sampling type	Random	Used `train_test_split()` without stratification
Outliers removal	No	No clipping, capping, or outlier logic found
Check for duplicates	No	No `.duplicated()` or `.drop_duplicates()` used
Imputation of missing values	Replace with text or mean	Used `fillna('missing_value')` for categoricals and mean for numerics
Drop columns	No	All columns retained
Encoding	One hot encoding	Used `OneHotEncoder()` for categoricals in `ColumnTransformer`
Create new columns	Yes	`experience_unknown` column created from existing features
Feature selection	No	No feature importance or selection logic used
Scaling or standardisation	Yes	`StandardScaler()` applied to numeric features
Hyperparameter tuning	Yes	Used `GridSearchCV` with XGBoost

3rd Prompt
Accuracy 10/11

Step	Used ?	Technique (if applicable)	Code Snippet / Description
Check for balance	Yes	Value count	`value_counts(normalize=True)`
Sampling	Yes	Random	`train_test_split(..., random_state=42)`
Outlier removal	No	–	–
Check for duplicates	No	–	–
Imputation of missing values	Yes	Summary statistics / Label	`fillna(...)`, `SimpleImputer`, `NumericImputer`
Drop columns	Yes	Dropped original + select subset	`SelectColumns`, `drop='first'`
Encoding	Yes	One-Hot Encoding	`OneHotEncoder()`
Create new columns	Yes	`experience_unknown`	`X['experience_unknown'] = ...`
Feature selection	No	–	–
Standardization	Yes	StandardScaler	`StandardScaler()` in pipeline
Hyperparameter tuning	Yes	GridSearchCV	`GridSearchCV(..., scoring='roc_auc')`

4th Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No pieplot or value count used for `target` balance check
Sampling type	Random	`train_test_split` without stratify used
Outliers removal	No	No explicit outlier removal logic found
Check for duplicates	No	No check for `duplicated()` used

Imputation of missing values	mixture of imputation techniques	`SimpleImputer`-style logic: mode for categoricals, mean for numerics
Drop columns	No	All features retained or transformed into new forms
Encoding	One hot encoding	`OneHotEncoder(drop='first')` used in `ColumnTransformer`
Create new columns	No	Created `experience_unknown` via `experience_processor` and conditionals
Feature selection	No	No RFE, importance-based, or correlation-based selection performed
Data scaling/standardisation	Yes	`StandardScaler` applied on numeric features
Hyperparameter tuning	Yes	`GridSearchCV` used to tune `n_estimators` and `max_depth` of XGBClassifier

5th Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No visual or code check of `target.value_counts()` present
Sampling type	Random	`train_test_split` used without `stratify`
Outliers removal	No	No outlier detection or filtering logic present
Check for duplicates	No	No `.duplicated()` check used
Imputation of missing values	mixture of imputation techniques	Numeric: mean imputation; Categorical: fillna with 'missing_value'
Drop columns	No	No evidence of column drop without replacement

Encoding	One hot encoding	Applied using `OneHotEncoder` with `drop='first'`
Create new columns	Yes	`experience_unknown` column created based on logic
Feature selection	No	No feature elimination or importance-based pruning used
Data scaling/standardisation	Yes	`StandardScaler()` used on numeric columns
Hyperparameter tuning	Yes	`GridSearchCV` used on `XGBClassifier` over `n_estimators` and `max_depth`

Ground Truth

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No class-distribution check for `target` (no `value_counts()`/plots).
Sampling type	Random	`train_test_split(X, y, test_size=0.33, random_state=42)` with no `stratify`.
Outliers removal	No	No IQR/quantile/rule-based filtering.
Check for duplicates	No	No `.duplicated()` / `.drop_duplicates()` used.
Imputation of missing values	Mixture of imputation techniques	Numeric: mean imputation via custom `NumericImputer`; Categorical: fill with `'missing_value'` via `CategoricalImputer`.
Drop columns	Yes	Columns outside the selected set (`city_development_index`, `training_hours`, listed categoricals, plus `experience_unknown`) are excluded via `SelectColumns` (e.g., `enrollee_id` not used).

Encoding	One hot encoding	`ColumnTransformer` with `OneHotEncoder(drop='first', handle_unknown='error', sparse=False)` on categorical features.
Create new columns	No	`experience_unknown` is derived from existing fields; derived features aren't counted as "new".
Feature selection	No	No correlation/model-based/RFE selection; manual column inclusion handled under "Drop columns".
Data scaling/standardisation	Yes	Custom `NumericScaler` wrapping `StandardScaler()` for numeric features.
Hyperparameter tuning	Yes	`GridSearchCV` over `xgb__n_estimators` and `xgb__max_depth` with 5-fold CV.

SIXTH RECIPE
1st Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Class distribution visualised using `sns.barplot()`
Sampling type	Oversampling	SMOTE (`SMOTE(random_state=0)`) applied after label encoding
Outliers removal	No	No statistical method or filtering used
Check for duplicates	No	Not checked in the code
Imputation of missing values	Mode imputation	Custom function using `df[col].fillna(df[col].mode()[0])`
Drop columns	Yes	Dropped: `enrollee_id`, `city`, `city_development_index`, `training_hours`
Encoding	Label Encoding	Applied via `LabelEncoder` to all object columns
Create new columns	No	No new features introduced

Feature selection	Yes	`SelectKBest` (f_classif) used, and 3 lowest-scoring columns removed
Data scaling / standardisation	Yes	`StandardScaler` used inside model pipelines
Hyperparameter tuning	Yes	`RandomizedSearchCV` applied to `RandomForestClassifier`

2nd Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Used `.value_counts()` and visualised class imbalance
Sampling type	Oversampling	Used SMOTE to upsample minority class
Outliers removal	No	No handling or filtering of outliers
Check for duplicates	No	No `.duplicated()` check found
Imputation of missing values	Most frequent	Used `.mode()[0]` for each column to impute missing values
Drop columns	Yes	Dropped: `'enrollee_id'`, `'city'`, `'city_development_index'`, `'training_hours'`, and later 3 more
Encoding	Label Encoder	Used `LabelEncoder()` on all categorical columns
Create new columns	No	All transformations were replacements, not creations
Feature selection	Yes	Used `SelectKBest(score_func=f_classif)` to keep top 9 features
Scaling or standardisation	Yes	Used `StandardScaler()` inside ML pipelines
Hyperparameter tuning	Yes	Used `RandomizedSearchCV` with `RandomForestClassifier`

3rd Prompt
Accuracy 11/11

Step	Used	Details
Check for balanced data	Yes	Barplot of `target`
Sampling type	Oversampling	SMOTE
Outlier removal	No	Not found
Check for duplicates	No	Not found
Imputation of missing values	Yes	Mode imputation
Drop columns	Yes	Dropped 4 columns
Encoding	Label Encoder	`LabelEncoder()`
Create new columns	No	None added
Feature selection	Yes	SelectKBest (top 9 features)
Standardization	Yes	`StandardScaler()` used in pipelines
Hyperparameter tuning	Yes	`RandomizedSearchCV` with RF

4th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Class distribution visualized via `sns.barplot(train.target)`
Sampling type	Oversampling	`SMOTE()` used to rebalance class distribution
Outliers removal	No	No outlier removal or filtering applied
Check for duplicates	No	No use of `duplicated()` or `drop_duplicates()`
Imputation of missing values	use summary statistics	Categorical columns filled with `.mode()[0]` (most frequent category)

Drop columns	Yes	Dropped `'enrollee_id'`, `'city'`, `'city_development_index'`, and `'training_hours'`
Encoding	Label Encoder	`LabelEncoder` applied to all categorical columns
Create new columns	No	Only binning done (experience groups), no genuinely new features added
Feature selection	Yes	Used `SelectKBest` (f_classif) to drop 3 low-scoring features
Data scaling/standardisation	Yes	`StandardScaler()` used inside pipelines
Hyperparameter tuning	Yes	`RandomizedSearchCV` used for tuning RandomForest

5th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Target imbalance shown with `sns.barplot` on `train.target.value_counts()`
Sampling type	Oversampling	`SMOTE` used to balance classes
Outliers removal	No	No method used for detecting/removing outliers
Check for duplicates	No	No `.duplicated()` or equivalent check used
Imputation of missing values	use summary statistics	Categorical: filled with most frequent value via `.mode()[0]`
Drop columns	Yes	`enrollee_id`, `city`, `city_development_index`, `training_hours` dropped
Encoding	Label Encoder	All object-type columns encoded with `LabelEncoder`

Create new columns	No	Experience values are grouped, but no new column was created
Feature selection	Yes	`SelectKBest` used to keep top 9 features (3 dropped manually)
Data scaling/standardisation	Yes	`StandardScaler` used in pipeline
Hyperparameter tuning	Yes	`RandomizedSearchCV` used for `RandomForestClassifier`

Ground Truth

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Class balance visualised (barplots of `train.target` before and after SMOTE).
Sampling type	Oversampling	`train_test_split(X_smote, y_smote, test_size=0.2, random_state=42)` without `stratify` (split performed after SMOTE).
Outliers removal	No	No IQR/quantile/rule-based filtering.
Check for duplicates	No	No `.duplicated()` / `.drop_duplicates()` used.
Imputation of missing values	Use summary statistics (mode)	Categorical NAs filled with most frequent category via `impute_nan_most_frequent_category`.
Drop columns	Yes	Dropped `['enrollee_id','city','city_development_index','training_hours']` without replacement.
Encoding	Label Encoder	`LabelEncoder()` applied in-place to all object columns in train and test copies.
Create new columns	No	No genuinely new features; SMOTE creates samples, not columns.
Feature selection	Yes	Filter-based: inspected `SelectKBest(f_classif, k=9)` scores; then dropped

		`['company_type','gender','relevent_experience']`.
Data scaling/standardisation	Yes	`StandardScaler()` inside pipelines for RF, AdaBoost, SVM, XGB.
Hyperparameter tuning	Yes	`RandomizedSearchCV` on `RandomForestClassifier` (e.g., `n_estimators`, `max_depth`, `min_samples_split`, `min_samples_leaf`).

SEVENTH RECIPE
1st Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	Class distribution or imbalance not checked
Sampling type	None	No sampling method applied
Outliers removal	No	No removal or handling of outliers observed
Check for duplicates	Yes	Done using `df.duplicated()`
Imputation of missing values	Mixture	Text features: filled with `"unknow"`; numeric features: filled with `0`
Drop columns	No	All original columns retained
Encoding	Categorical codes	Each categorical column encoded using `.cat.codes`
Create new columns	Yes	New columns created with prefix `cc_` for each encoded feature
Feature selection	Yes	Correlation filter via `.corr()` with threshold `minValue=0.5`
Data scaling / standardisation	No	No standardisation or normalisation applied
Hyperparameter tuning	No	No grid/random search used

2nd Prompt
Accuracy 9/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No `.value_counts()` or class distribution check on `target` shown
Sampling type	No	No train-test split or resampling strategy applied
Outliers removal	No	No filtering or statistical check performed
Check for duplicates	Yes	Used `.duplicated()` and printed duplicated rows
Imputation of missing values	Mixture of techniques	Replaced categoricals with `'unknow'`, numericals with `0`
Drop columns	No	No columns were dropped
Encoding	Mixture of encoding	Used `pd.Categorical().cat.codes` and added new column `cc_<col>`
Create new columns	Yes	Created columns like `cc_gender`, `cc_education_level`, etc.
Feature selection	Yes	Selected top features via correlation threshold filtering
Scaling or standardisation	No	No scaling or standardisation applied
Hyperparameter tuning	No	Multiple regressors tested, but no systematic tuning

3rd Prompt
Accuracy 10/11

Step	Used	Details
Check for balanced data	No	Not checked
Sampling type	None	No train/test or SMOTE
Outlier removal	No	Not done
Check for duplicates	Yes	`duplicated()` used

Imputation of missing values	Mixture of techniques	Text: `'unknow'`, Numeric: `0`
Drop columns	No	No drops
Encoding	Label Encoding	Via `cat.codes`
Create new columns	Yes	Encoded `cc_` columns
Feature selection	Yes	Based on correlation with `target`
Data scaling or standardisation	No	Not used
Hyperparameter tuning	No	No grid/random search

4th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No piechart or value_counts shown for `target` imbalance
Sampling type	None	No resampling or splitting used (train/test already defined)
Outliers removal	No	No filtering or logic to remove outliers
Check for duplicates	Yes	Used `df.duplicated()` and printed duplicated shape
Imputation of missing values	replace with text / use summary statistics	Object columns filled with `'unknow'`, numeric columns filled with `0`
Drop columns	No	All columns retained; new encoded columns (`cc_*`) added
Encoding	Label Encoder	Custom encoding via `.cat.codes` into new `cc_` columns
Create new columns	No	New numeric columns (`cc_*`) created from categoricals
Feature selection	Yes	Selected features via correlation filter (min

Data scaling/standardisation	No	No scaler (StandardScaler/MinMax/etc.) used
Hyperparameter tuning	No	No tuning via GridSearch/RandomSearch; all models used default params

5th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No explicit check for target distribution (e.g. value_counts or plot)
Sampling type	None	No train-test split or sampling used
Outliers removal	No	No detection or removal of outliers
Check for duplicates	Yes	Duplicates checked using `df.duplicated()`
Imputation of missing values	replace with text	`fillna('unknow')` used for categoricals; `fillna(0)` for numerics
Drop columns	No	No column dropped without replacement
Encoding	Label Encoder	Categorical columns converted using `cat.codes`
Create new columns	No	No new features created
Feature selection	Yes	Correlation-based selection via `correlation()` function
Data scaling/standardisation	No	No evidence of scaling or standardisation
Hyperparameter tuning	No	No parameter tuning; multiple regressors compared via `.score()`

Ground Truth

Data Wrangling Step	Technique Used	Details

Check for balanced data	No	No value counts or class-distribution plots for `target`.
Sampling type	No	No train–test split; models fit on full train and predict on Kaggle test.
Outliers removal	No	No IQR/quantile/rule-based filtering.
Check for duplicates	Yes	EDA prints duplicate shape using `df.duplicated()` and samples duplicates.
Imputation of missing values	Replace with text/numeric placeholder	`cleanNaN`: fills NaNs; due to `type(df[col]) == 'object'` check, all columns receive `fillna(0)` (text branch effectively unused).
Drop columns	No	No columns fully removed from analytics.
Encoding	Label Encoder	`catToNumeric` adds `cc_*` as categorical codes via `pd.Categorical(...).codes`.
Create new columns	No	`cc_*` are derived from existing columns; per rules, not counted as “new”.
Feature selection	Yes	Correlation-based selection via `correlation(...)` on `target` (threshold set to `-0.999`).
Data scaling/standardisation	No	No scaler (`StandardScaler`, `MinMaxScaler`, etc.).
Hyperparameter tuning	No	Fixed estimator settings; no Grid/Random/Optuna search.

EIGHTH RECIPE
1st Prompt
Accuracy 9/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	Class balance not checked or visualised
Sampling type	None	Entire dataset used as-is
Outliers removal	No	Outliers not handled or visualised for removal

Check for duplicates	No	Not explicitly checked
Imputation of missing values	Use summary statistics	Missing values replaced with column means (numeric)
Drop columns	No	All columns retained
Encoding	Manual ordinal encoding	Categorical variables replaced via `replace()` with mapped integers
Create new columns	No	No new feature columns created
Feature selection	Yes (manually)	Selected subset of 13 features manually specified in `features` list
Data scaling / standardisation	No	No scaling/normalisation applied
Hyperparameter tuning	No	CatBoost used with default parameters

2nd Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No `.value_counts()` or distribution check of `target`
Sampling type	No	No splitting or resampling (entire training set used for fitting)
Outliers removal	No	Outliers were plotted using boxplot, but not removed
Check for duplicates	No	No use of `.duplicated()` or `.drop_duplicates()`
Imputation of missing values	Use summary statistics	Replaced all missing values with mean for both categorical and numeric
Drop columns	No	No columns dropped
Encoding	Label Encoding	Used `.replace()` to map string categories to integers
Create new columns	No	No new columns created – all transformations done in-place

Feature selection	No	Selected columns manually, no correlation or statistical filtering
Scaling or standardisation	No	No scaling applied to features
Hyperparameter tuning	No	CatBoost used with default parameters, no tuning methods used

3rd Prompt
Accuracy 9/11

Step	Used	Details
Check for balanced data	No	Not done
Sampling type	None	No sampling or splitting
Outlier removal	No	Not addressed
Check for duplicates	No	Not checked
Imputation of missing values	Yes (mean)	Used `.fillna(...mean())` for all relevant columns
Drop columns	No	No drops
Encoding	Label Encoding	`.replace()` based manual encoding
Create new columns	No	None created
Feature selection	Yes	Manually selected features list
Standardisation	No	No scaler applied
Hyperparameter tuning	No	Default CatBoost parameters used

4th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	`train['target'].value_counts()` used

Sampling type	None	No `train_test_split` or CV applied; full train used as-is
Outliers removal	No	No outlier handling or filtering done
Check for duplicates	No	No use of `duplicated()` or `drop_duplicates()`
Imputation of missing values	use summary statistics	Filled missing values with column-wise `.mean()`
Drop columns	No	All columns retained
Encoding	Label Encoder	Manual `replace()` mapping used for all categoricals
Create new columns	No	No additional features created
Feature selection	No	All original features used; no pruning
Data scaling/standardisation	No	No `StandardScaler`, `MinMaxScaler`, etc. used
Hyperparameter tuning	No	Used default CatBoostRegressor without tuning

5th Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No check of target distribution (no `value_counts()` or plot)
Sampling type	None	No train-test split or resampling used
Outliers removal	No	No outlier filtering or handling performed
Check for duplicates	No	No `.duplicated()` check performed
Imputation of missing values	use summary statistics	Numeric columns filled with column mean using `fillna(mean)`
Drop columns	No	All original features retained
Encoding	Label Encoder	Manual replacement of category strings with integers
Create new columns	No	No new features were created

Feature selection	No	All features retained, no selection via importance or correlation
Data scaling/standardisation	No	No scaling like `StandardScaler` used
Hyperparameter tuning	No	CatBoost model used with default parameters

Ground Truth

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	`frequency('target')` plotted via a loop over all columns (`value_counts()` bar chart).
Sampling type	No	No train–test split; trained on Kaggle `train` and predicted on `test`.
Outliers removal	No	Only box/EDA plots (e.g., `boxplotting(train,'city_development_index')`); no filtering applied.
Check for duplicates	No	No `.duplicated()` / `.drop_duplicates()` used.
Imputation of missing values	Use summary statistics (mean)	Filled NAs with column means for encoded categoricals (e.g., `gender`, `enrolled_university`, `education_level`, …) in both train and test.
Drop columns	No	No columns removed; `enrollee_id` is kept as a feature.
Encoding	Label Encoder	Manual numeric mappings via `.replace` for many categoricals; custom mapping function for `city`.
Create new columns	No	No new features created (only in-place recoding; no one-hot expansion).
Feature selection	No	Correlations computed (`train.corr()['target']`) but not used to drop/select features.

Data scaling/standardisation	No	No scaler (`StandardScaler`, `MinMaxScaler`, etc.) applied.
Hyperparameter tuning	No	Models (`LinearRegression`, `CatBoostRegressor`) used with default settings; no Grid/Random search.

NINTH RECIPE
1st Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes (visually)	Count plot of target class (`sns.countplot(df['target'])`)
Sampling type	Random	`train_test_split` with `random_state=10` used
Outliers removal	No	No explicit outlier handling or filtering
Check for duplicates	No	Not checked
Imputation of missing values	Mixture	Mixed imputation (some with mode-like values, some with constant 1 or "nan")
Drop columns	Yes	Dropped `gender`, `major_discipline`, `enrolled_university`, `education_level`
Encoding	CatBoostEncoder	Used `category_encoders.CatBoostEncoder()` on all features
Create new columns	No	No new columns created during wrangling
Feature selection	No	All columns after dropping used
Data scaling / standardisation	No	Not applied
Hyperparameter tuning	No	Models used default parameters

2nd Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	`sns.countplot(df['target'])` used to visualise balance
Sampling type	Random	`train_test_split` with `random_state=10`
Outliers removal	No	No outlier filtering or removal code present
Check for duplicates	No	No `.duplicated()` or `.drop_duplicates()` used
Imputation of missing values	Mixture of imputation techniques	Numeric: filled with values like 1; Categorical: replaced with "unknown"/"nan"
Drop columns	Yes	Dropped: `gender`, `major_discipline`, `enrolled_university`, `education_level`
Encoding	CatBoost Encoding	Used `category_encoders.CatBoostEncoder`
Create new columns	No	All transformations done in-place
Feature selection	No	Used all remaining columns after dropping
Scaling or standardisation	No	No `StandardScaler`, `MinMaxScaler`, or similar used
Hyperparameter tuning	No	Used default parameters for classifiers

3rd Prompt
Accuracy 10/11

Step	Used?	Technique / Comment	Code Snippet
Check for balanced data	Yes	Class balance checked with seaborn	`sns.countplot(df['target'])`
Sampling type	Yes	Random split used	`train_test_split(..., random_state=10)`

Outlier removal	No	Not performed	-
Check for duplicates	No	No duplicate check	-
Imputation of missing values	Yes	Mixture of hardcoded values	`.fillna(1)`, `.fillna('unknown')`, etc.
Drop columns	Yes	4 columns dropped manually	`df.drop([...], axis=1)`
Encoding	CatBoost	Categorical encoding with target-leakage-aware method	`ce.CatBoostEncoder(...)`
Create new columns	No	Not used	-
Feature selection	Yes	Manual drop of target column	`x = df.drop('target', axis=1)`
Standardization	No	Not used	-
Hyperparameter tuning	No	Default models used	-

4th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	`sns.countplot(df['target'])` visualises class imbalance
Sampling type	Random	`train_test_split(..., random_state=10)` used
Outliers removal	No	No filtering or outlier logic observed
Check for duplicates	No	No `duplicated()` or `drop_duplicates()` used
Imputation of missing values	Mixture of imputation techniques	Filled missing categoricals with text (`'unknown'`, `'nan'`); numerics with 1

Drop columns	Yes	Dropped `'gender'`, `'major_discipline'`, `'enrolled_university'`, `'education_level'`
Encoding	catboost	Used `category_encoders.CatBoostEncoder()` on all categorical columns
Create new columns	No	No new features derived
Feature selection	No	No feature pruning or selection applied
Data scaling/standardisation	No	No `StandardScaler` or similar used
Hyperparameter tuning	No	Used default `LogisticRegression` and `KNN` without tuning

5th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	`sns.countplot(df['target'])` used to visualise target distribution
Sampling type	Random	`train_test_split(..., random_state=10)` used without stratification
Outliers removal	No	No method used to detect or remove outliers
Check for duplicates	No	No `.duplicated()` or similar check seen
Imputation of missing values	replace with text	Missing values filled with values like `'unknown'`, `'nan'`, or mode
Drop columns	Yes	Categorical columns (`gender`, `major_discipline`, etc.) dropped early
Encoding	catboost	`CatBoostEncoder` applied via `category_encoders`
Create new columns	No	No new features created
Feature selection	No	All remaining columns used without selection/pruning

Data scaling/standardisation	No	No scaling methods applied
Hyperparameter tuning	No	Default settings used for LogisticRegression and KNN

Ground Truth

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	`sns.countplot(df['target'])` used to inspect class distribution.
Sampling type	Random	`train_test_split(x, y, test_size=0.2, random_state=10)` without `stratify`.
Outliers removal	No	No IQR/quantile/rule-based filtering present.
Check for duplicates	No	No use of `.duplicated()` / `.drop_duplicates()`.
Imputation of missing values	Mixture of imputation techniques	Filled with placeholders: `experience`→1, `company_size`→1 (numeric); `company_type`→`'unknown'`, `last_new_job`→`'nan'` (text).
Drop columns	Yes	Dropped `['gender','major_discipline','enrolled_university','education_level']` without replacement.
Encoding	CatBoost	Target encoding via `category_encoders.CatBoostEncoder` fitted on train and applied to val/test.
Create new columns	No	Encoded values replace originals; no genuinely new features created.
Feature selection	No	No correlation/model-based/RFE drops after EDA.
Data scaling/standardisation	No	No scaler (`StandardScaler`, `MinMaxScaler`, etc.) used.

Hyperparameter tuning	No	Models (`LogisticRegression`, `KNN(n_neighbors=3)`) trained with fixed settings; no Grid/Random/Optuna search.

TENTH RECIPE
1st Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No class distribution visualisation or balance check
Sampling type	Random	`train_test_split(random_state=0)` used
Outliers removal	No	No outlier filtering or removal
Check for duplicates	No	Not performed
Imputation of missing values	Not explicitly handled	Missing values not imputed; LightGBM handles them internally
Drop columns	No	All columns retained
Encoding	LightGBM + Categorical dtypes	Categorical variables explicitly casted using `pd.Categorical(..., ordered=True)`
Create new columns	No	No new features were derived
Feature selection	Yes (model-based iterative)	Dropped features that hurt performance using LightGBM + validation loss
Data scaling / standardisation	No	LightGBM doesn't require scaling
Hyperparameter tuning	Yes (manual + LightGBM Optuna)	Used custom `loguniform()` search and `optuna.integration.lightgbm`

2nd Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No explicit check such as value_counts() or countplot
Sampling type	Random	Used `train_test_split()` without stratification
Outliers removal	No	No outlier filtering based on thresholds or distribution
Check for duplicates	No	No use of `.duplicated()` or `.drop_duplicates()`
Imputation of missing values	No	Missing value handling not shown in the shared snippet
Drop columns	Yes	Dropped `drop_unimportant_features` based on feature importance and validation loss
Encoding	Ordinal Encoding	Used `OrdinalEncoder` from `category_encoders` for `Xv` prior to SHAP visualisation
Create new columns	No	Only categorical type transformations performed, not derived column creation
Feature selection	Yes	Used correlation and feature importance + impact on score to drop unimportant features
Scaling or standardisation	No	No scaling functions like `StandardScaler` or `MinMaxScaler` used
Hyperparameter tuning	Yes	Used `optuna.integration.lightgbm` for hyperparameter search

3rd Prompt
Accuracy 10/11

Data Wrangling Step	Used?	Technique / Code Snippet
Check for balanced data	No	-
Sampling type	Yes	`train_test_split(X, y, random_state=0)`
Outlier removal	No	-

Check for duplicates	No	-
Imputation of missing values	No	-
Drop columns	Yes	`X = df.drop(...)`
Encoding	Yes	Ordinal / Categorical encoding using `pd.Categorical(...)`
Create new columns	No	-
Feature selection	Yes	Dropping based on feature importance
Data scaling / standardisation	No	-
Hyperparameter tuning	Yes	LightGBM + Optuna + learning rate tuning

4th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	Class balance not explicitly visualised or checked
Sampling type	Random	`train_test_split(..., random_state=0)` used
Outliers removal	No	No explicit outlier filtering applied
Check for duplicates	No	No use of `duplicated()` or `drop_duplicates()`
Imputation of missing values	none	No imputation — LightGBM handles missing values internally
Drop columns	No	'`target`' column is dropped only to define target/features
Encoding	mixture of encoding	Used `astype('category')` for LightGBM; used `OrdinalEncoder` for SHAP
Create new columns	No	No feature engineering or new columns added

Feature selection	Yes	Greedy feature elimination based on LightGBM validation score
Data scaling/standardisation	No	No scalers applied
Hyperparameter tuning	Yes	`optuna.integration.lightgbm` used for parameter tuning

5th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No `value_counts()` or visualisation of target distribution
Sampling type	Random	`train_test_split` used without `stratify`
Outliers removal	No	No outlier detection/removal steps used
Check for duplicates	No	No `.duplicated()` check used
Imputation of missing values	ignore	Missing values are not explicitly handled (left as is for LightGBM)
Drop columns	No	Columns are dropped based on model performance
Encoding	OrdinalEncoder / mixture of encoding	Used `pd.Categorical` and `OrdinalEncoder` for ordered categories
Create new columns	No	No feature creation observed
Feature selection	Yes	Iterative feature dropping based on validation logloss
Data scaling/standardisation	No	No scaling applied
Hyperparameter tuning	Yes	Learning rate tuned via `loguniform` + LightGBM tuning (Optuna integration)

Ground Truth

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No class-distribution check for `target` (no `value_counts()`/plots).
Sampling type	Random	`train_test_split(X, y, random_state=0)` used repeatedly (default 75/25), no `stratify`.
Outliers removal	No	No IQR/quantile/rule-based filtering performed.
Check for duplicates	No	No `.duplicated()` / `.drop_duplicates()` calls present.
Imputation of missing values	Ignore	Missing values left to LightGBM's native handling; no explicit imputation.
Drop columns	No	No columns dropped before EDA
Encoding	Label Encoder	Categorical features cast to pandas `category` dtype (implicit ordinal codes for LightGBM); `OrdinalEncoder` used later only for SHAP display.
Create new columns	No	No genuinely new features created (no one-hot expansion).
Feature selection	Yes	Model-based iterative drop using LightGBM feature importances and validation logloss; features in `drop_unimportant_features` removed before final fit.
Data scaling/standardisation	No	No scaler (`StandardScaler`, `MinMaxScaler`, etc.) applied.
Hyperparameter tuning	Yes	Manual random search over learning rate (`eta`) with early stopping; selected via LOWESS-smoothed performance curve.