Data Preparation & Ethical Data Handling

AI Masters Capstone Project - Presentation 2

Jonathan Agustin

November 2024

What We'll Cover Today

- Embedding ethics and fairness at the pipeline level
- Advanced automated preprocessing: beyond missing values
- Rigorous data validation: schema enforcement + anomaly detection
- Future-proofed privacy & compliance (e.g., GDPR and beyond)
- Expert validation set design (Thomas, 2017) for real-world resilience
- Next-level bias detection and rebalancing strategies

We're moving from "good enough" to elite data practices that anticipate tomorrow's challenges.

Ethical Data Handling Matters

- Data isn't just numbers—it's people's lives and societal narratives.
- Anticipate downstream impacts: prevent models that discriminate or misinform.
- Build trust: ethical data stewardship differentiates you in a crowded market.
- **Pro tip:** Involve domain and ethics experts from project inception, not as an afterthought.

Ethics done expertly: a strategic investment, not a cost.

Automated Preprocessing “Power Moves”

- Dynamic and modular pipelines: easy to update when data schema evolves
- Parametrized transformations: track imputation strategies, encoders, scalers in version control
- CI/CD integration: automated checks prevent subtle data drifts from reaching production
- Advanced transformations: leverage domain knowledge to engineer more predictive features from raw inputs

Set up your pipeline so that improvements flow seamlessly, without reinventing the wheel.

Practical Preprocessing Techniques (Expert-Level)

df = pd.read$_c sv(''raw_data.csv'')$

Log data schema version

mlflow.set$_e xperiment(''data_preprocessing'')with mlflow.start_run():$
$mlflow.log_param(''raw_data_shape'', df.shape)$

numeric$_c ols = df.select_dtypes(include = ['float','int']).columns imputer =$
$SimpleImputer(strategy =' median')df[numeric_cols] =$
$imputer.fit_transform(df[numeric_cols])$

categorical$_c ols = df.select_dtypes(include = ['object']).columns Advanced encoding :$
$TargetEncoder for high - cardinality features encoder = TargetEncoder(cols =$
$categorical_cols)df[categorical_cols] =$
$encoder.fit_transform(df[categorical_cols], df['target_variable'])$

scaler = StandardScaler() df[numeric$_c ols] = scaler.fit_transform(df[numeric_cols])$

Data Quality Assurance: Advanced Strategies

class InputSchema(pa.SchemaModel): age: Series[int] = pa.Field(ge=0, le=120) income: Series[float] = pa.Field(ge=0) Add dynamic checks that reference domain insights target$_{v}ariable : Series[int] = pa.Field(in_{v}alues = [0, 1])$

class Config: strict = True

Add anomaly checks for advanced validation schema = InputSchema.to$_{s}chema().add_{c}hecks(Check(lambdadf : df['income'].mean() < 1e6, error = "Averageincomesuspiciouslyhigh"))$

try: validated$_{d}f = schema.validate(df)exceptpa.errors.SchemaErrorase :$ $print("Datavalidationfailed :$ $", e)Integratewithalertingsystems(PagerDuty, Slack)$

Expert-level validation: layered checks, domain logic, automated alerts.

Crafting Expert-Level Validation Sets (Thomas, 2017)

- Multiple “scenario-based” validation sets, not just one
- Time-based splits that reflect production roll-out schedules
- Varying user cohorts to mimic new demographics or products
- Regularly refresh validation sets as data distribution drifts

Top teams treat validation sets as living assets—continuously improved for maximum realism.

Beyond Random Splits: Pro-Level Validation (Thomas, 2017)

- **Stress Testing**: Remove entire feature segments to simulate sensor failures or data pipeline downtime.
- **Cohort-Based Validation**: Validate on subsets representing future strategic markets or demographics.
- **Multi-Stage Validation**: Incrementally reveal validation data, mirroring real-time production data arrival.

Such techniques yield resilience and maintain trust as conditions evolve.

Practical Expert-Level Examples (Thomas, 2017)

- **Time Series**: Rolling window validations (e.g., train on Jan–May, validate on June; then train on Feb–June, validate on July, etc.)
- **New Entities**: Introduce synthetic "unseen" products or users to test model adaptability.
- **Domain Shifts**: Create scenario-based validations if you anticipate policy changes, new competitor products, or economic downturns.

Master-level validation ensures future-readiness and stable performance in a dynamic world.

Kaggle Production: Insider Secrets (Thomas, 2017)

- Maintain independent validation sets that align with long-term product goals.
- Don't let public leaderboard scores dictate final decisions—correlate with private validation sets.
- Regularly "audit" model performance over multiple, evolving validation sets for stable generalization.

This is how you avoid "trophy overfitting" and achieve genuine real-world impact.

Privacy Protection: Future-Proof Flexible

Drop direct PII and consider synthetic data generation if needed
pii$_{c}ols = ['name',' email',' phone_{n}umber']df = df.drop(columns = [cforcinpii_{c}olsifcindf.columns])$

Expert tip: integrate a differential privacy library to add statistical noise to sensitive aggregates. e.g., "privacy$_{e}ngine" from Opacus (for PyTorch)$

By designing flexible compliance tools, you stay ahead of evolving regulatory landscapes.

Bias Detection and Mitigation: Advanced Methods

data = BinaryLabelDataset(favorable$_{l}abel = 1, unfavorable_{l}abel = 0, df = df, label_{n}ames = ['approved_{l}oan'], protected_{a}ttribute_{n}ames = ['gender'])$

Check intersectional groups if available Use adversarial debiasing to reduce discrimination debiasing$_{m}odel = AdversarialDebiasing(unprivileged_{g}roups = ['gender' : 0], privileged_{g}roups = ['gender' : 1], scope_{n}ame =' debiasing_{m}odel', sess = None)Fitthismodelandevaluatemetricslikeequalizedoddsovertime$

Top practitioners treat bias mitigation as an ongoing process, not a one-time fix.

A Holistic, Expert Data Strategy

- Pipeline as a living system: continuously evolving and improving
- Ethical, privacy-first approach as a strategic differentiator
- Validation sets as flexible testbeds for future conditions
- Bias mitigation as a dynamic, iterative practice

These tactics don't just solve problems—they preempt them, ensuring your ML solutions stand the test of time.

Next Steps

- Next: Automating Model Training & Ethical Model Evaluation
- Connect these expert data strategies to model tuning and monitoring

You're now ready to build ML pipelines that aren't just good—they're world-class.

References

- Thomas, R. (2017). *How (and why) to create a good validation set*. Retrieved from https://rachel.fast.ai/posts/2017-11-13-validation-sets/
- *AIF360* toolkit: https://github.com/Trusted-AI/AIF360
- *pandera* library: https://pandera.readthedocs.io/
- GDPR guidelines: https://gdpr.eu/
- Opacus (differential privacy for PyTorch): https://github.com/pytorch/opacus
- Great Expectations (data quality): https://greatexpectations.io/