Data Preparation & Ethical Data Handling

AI Masters Capstone Project - Presentation 2

Jonathan Agustin

November 2024

What We'll Cover Today

- Why ethical data handling matters: trust, fairness, reputation
- Practical, automated preprocessing: from cleaning to transformation
- Ensuring data quality: validation scripts and continuous checks
- Privacy and compliance: embedding GDPR and data protection into your pipeline
- Identifying and mitigating bias: concrete techniques and tools

Our focus: Move from theory to practice, ensuring data integrity, privacy, and fairness.

Why Ethical Data Handling Matters

- Data = real individuals' personal details, behaviors, and rights
- Trust and credibility: A must for sustainable product adoption
- Reduces legal risks and reputational damage from misuse or bias

Ethical data stewardship is the moral and practical cornerstone of responsible AI.

Automated Data Preprocessing

- Systematically detect and address missing or inconsistent values
- Standardize units, formats, and encodings without manual intervention
- Increase reproducibility, reduce human bias and manual errors

Automation lets us spend less time fixing data manually, and more time ensuring quality and fairness.

Practical Preprocessing Techniques

Load your raw data df = pd.read$_{c}sv(''raw_{d}ata.csv'')$

Handle missing numeric values by imputing the mean numeric$_{c}ols = df.select_{d}types(include = ['float','int']).columnsimputer = SimpleImputer(strategy =' mean')df[numeric_{c}ols] = imputer.fit_{t}ransform(df[numeric_{c}ols])$

One-hot encode categorical variables categorical$_{c}ols = df.select_{d}types(include = ['object']).columnsencoder = OneHotEncoder(sparse_{o}utput = False, handle_{u}nknown =' ignore')encoded = encoder.fit_{t}ransform(df[categorical_{c}ols])encoded_{d}f = pd.DataFrame(encoded, columns = encoder.get_{f}eature_{n}ames_{o}ut(categorical_{c}ols))df = pd.concat([df.drop(columns = categorical_{c}ols), encoded_{d}f], axis = 1)$

Scale numeric features for model readiness scaler = StandardScaler()

Data Quality Assurance

Define a schema class InputSchema(pa.SchemaModel): age: Series[int] = pa.Field(ge=0, le=120) income: Series[float] = pa.Field(ge=0) Add more columns and constraints as needed

class Config: strict = True no unexpected columns allowed

Validate the dataframe try:
validated$_{d}f = InputSchema.validate(df)exceptpa.errors.SchemaErrorase: Logerrorandhaltthepipelineprint("Datavalidationfailed: ", e)handletheerror(e.g., sendalert, stopthepipeline)$

In practice, this ensures we catch data issues before model training, improving reliability and trust.

Privacy Protection and Compliance

Example: Pseudonymize user IDs if 'user$_i d' in df.columns : df['user_i d_h ashed'] = df['user_i d'].apply(lambda x : hashlib.sha256(str(x).encode()).hexdigest()) df.drop(columns = ['user_i d'], inplace = True)$

Remove personally identifiable information (PII) pii$_c olumns = ['name', 'email', 'phone_n umber'] df = df.drop(columns = [col for col in pii_c olumns if col in df.columns])$

Further steps could include encryption for storage or implementing user consent checks

Integrating privacy measures at the data layer reduces risks and builds user trust.

Detecting and Mitigating Bias

Convert df to aif360 dataset, specifying label and protected attribute data $=$ BinaryLabelDataset(favorable$_label = 1, unfavorable_label = 0, df = df, label_names = ['approved_loan'], protected_attribute_names = ['gender'])$

metric $=$ BinaryLabelDatasetMetric(data, unprivileged$_groups = ['gender' : 0], privileged_groups = ['gender' : 1])$

Check disparate impact (ratio of favorable outcomes between groups) disparate$_impact = metric.disparate_impact()print("DisparateImpact : ", disparate_impact)$

If disparate impact < 0.8, consider reweighing if disparate$_impact < 0.8 : rw = Reweighing(unprivileged_groups = ['gender' : 0], privileged_groups = ['gender' : 1])data_transformed = rw.fit_transform(data)data_transformedcannowbeusedformodelingwithlessbias$

A Holistic Data Strategy

- Integrate cleaning, validation, privacy, and bias mitigation into one automated flow
- Document steps for transparency and auditability
- Regularly iterate and improve based on feedback and monitoring

A well-structured, ethical data pipeline is the backbone of fair and effective AI systems.

Next Steps

- Next Presentation: Automated Model Training & Ethical Evaluation
- Connect ethical data pipelines to model building and deployment best practices

The path ahead: turning ethical principles into sustainable, real-world AI solutions.