Responsible AI – Predicting Customer Interest in Insurance

Data Science Assignment for Lloyds Bank

Author: Divya Thekke Kanapram
Role Target: Responsible AI Specialist

Executive Summary / Goal

The Challenge:

- Marketing department wants to increase home insurance product adoption
- Need to improve customer selection for future campaigns
- Current approach: Low conversion rate, inefficient resource allocation

The Solution:

- Build ML model to predict which customers will create insurance accounts
- Use historical campaign data to identify high-probability customers
- Enable targeted marketing with higher conversion rates

Expected Outcome:

- Higher proportion of customers purchasing product
- More efficient marketing spend
- Data-driven campaign decisions

Project Overview

What We Built:

- Production-ready ML pipeline
- Two classification models (Logistic Regression + Random Forest)
- Comprehensive feature engineering
- Model explainability tools
- Full test coverage

Key Metrics:

- **ROC AUC**: 0.97 (Logistic Regression), 0.99 (Random Forest)
- **Training Data:** 1,604 labeled samples from 16,591 merged records
- **Precision@50**: 0.48-0.54 (top 50 customers)

PART 1: DATA DISCOVERY AND SOURCING

What data is available?

Campaign Dataset: Customer demographics & campaign interactions.

Mortgage Dataset: Financial & employment attributes.

Dataset	Campaign	Mortgage
Number of rows	32060	32561
Number of columns	16	18
Numerical	1	0
Categorical	16	21
Boolean	0	0
Datetime	0	1

```
Campaign shape: (32060, 16)
Mortgage shape: (32561, 18)
```

Campaign columns:

```
 ['participant_id', 'name_title', 'first_name',
'last_name', 'age', 'postcode', 'marital_status',
'education', 'job_title', 'occupation_level',
'education_num', 'familiarity_FB', 'view_FB',
'interested_insurance', 'company_email',
'created_account']
```

Mortgage columns:

```
 ['full_name', 'dob', 'town', 'paye', 'salary_band',
'years_with_employer', 'months_with_employer',
'hours_per_week', 'capital_gain', 'capital_loss',
'new_mortgage', 'sex', 'religion', 'relationship',
'race', 'native_country', 'workclass',
'demographic_characteristic']
```

Implementation: `src/load_data.py` - Centralized data loading with configurable paths

What Information Do Sources Contain?

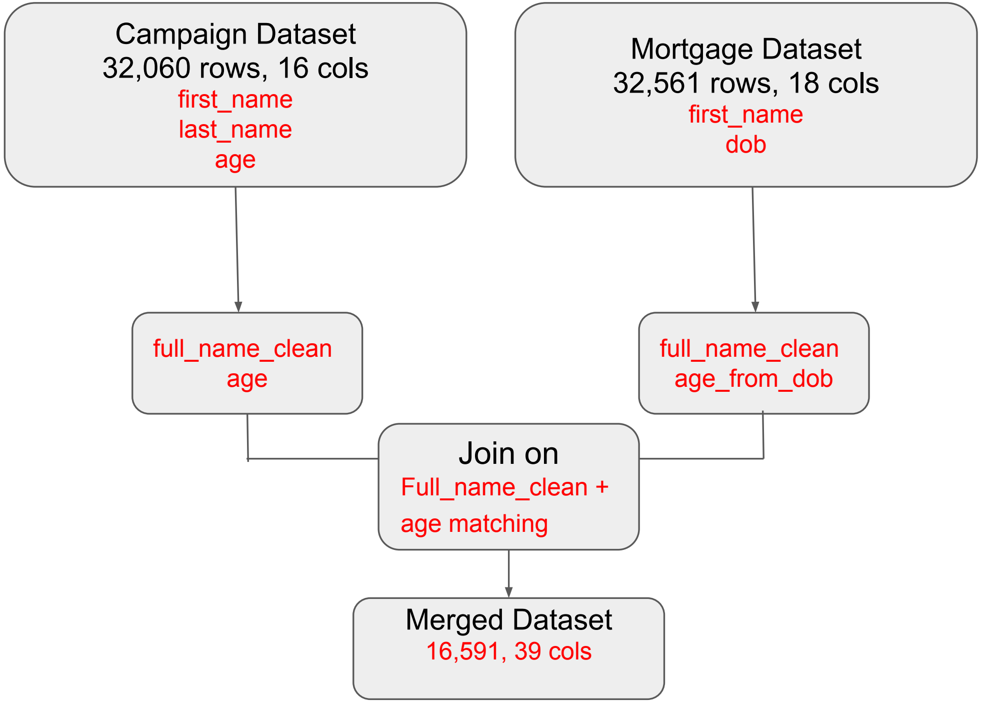
Campaign Dataset Features:
- Demographics: Age, education, marital status, occupation
- Target: `created_account` (Yes/No/NaN) - **90.6% missing** (expected for prediction)

Mortgage Dataset Features:
- Financial: Salary band, capital gains/losses, employment duration
- Demographics: Sex, religion, relationship, race, native country, workclass
- Geographic: Town, demographic characteristics

Usefulness Assessment:
- Rich demographic and financial features
- Geographic data for segmentation
- High missing rate in target (expected for prediction task)

Implementation: `src/eda.py` - Comprehensive EDA module with data discovery functions

Are We Able to Use the Data?



Data usability

Data Usability:

- Mergeable: 23,621 name matches found, 16,591 successfully merged (51.7% match rate)
- Clean: No duplicate rows detected
- Sufficient Volume: 1,604 labeled samples for training

Data pipeline:

Load Data → Clean & Standardize → Merge → Feature Engineering → Model Training

PART 2: DATA QUALITY

Data Quality Assessment

Aspect	Details
Overall Quality	Good with some issues
Campaign Dataset	32,060 rows × 16 columns
Mortgage Dataset	32,561 rows × 18 columns
Merged Dataset	16,591 rows × 39 columns (before cleaning)
Duplicates	0 in both datasets
Missing Values	created_account: 90.3% missing; name_title: 38.1% missing
Merge Success Rate	51.7% of campaign records matched
Consistency	Data types consistent across datasets
Volume	Sufficient for modeling (>30k records per source)

Issue	Solution Applied
Salary units	Standardized to annual GBP (salary_value_gbp)
Employment duration	Combined years + months → employment_duration_years
Capital gain/loss	Combined into net_profit
Age formats	Calculated age_from_dob from DOB (reference year: 2018)
Redundant columns	Dropped name, age, and temporary merge columns

How Can We Join Datasets?

Join Strategy	Composite Key: full_name_clean + age
Name standardization	Lowercase, strip whitespace, remove titles
Age validation	Exact match using DOB-derived age
Merge logic	Inner join on [full_name_clean, age]
Match results	23,621 name matches → 16,591 final matches (51.7%)

PART 3: EXPLORATORY ANALYSIS

Are There Any Obvious Patterns?

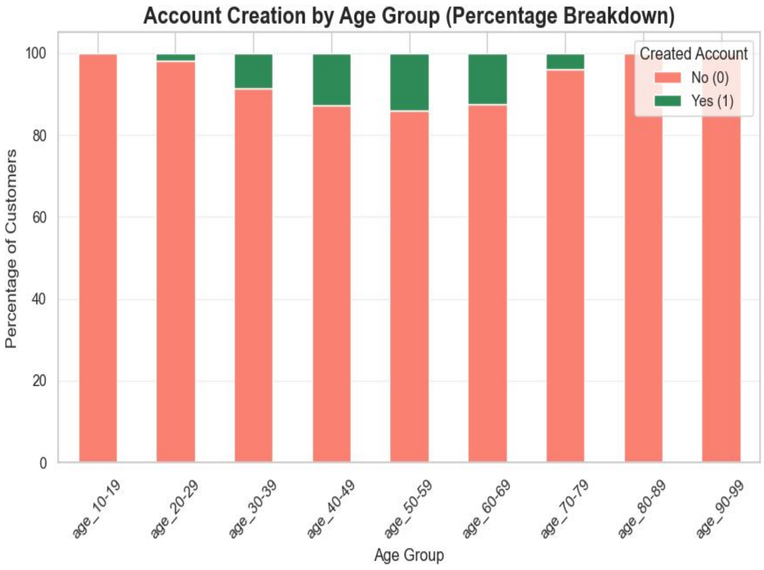
Target Variable Distribution:

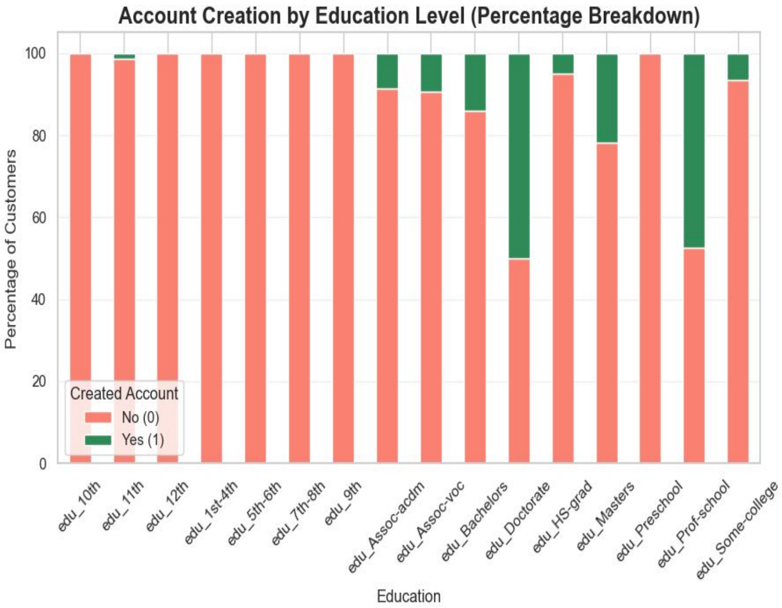
- Class 0 (No Account): 1,468 samples (91.4%)
- Class 1 (Account Created): 136 samples (8.6%)
- Severe class imbalance - handled with `class_weight='balanced'`

Demographic Patterns:

- Age Groups: 30-59 show higher conversion (8-14% vs <2% for other groups)
- Education: Higher education correlates with conversion
    - Masters: 21.84% conversion
    - Doctorate: 50% conversion
    - Prof-school: 47.37% conversion
- Marital Status: Married-civ-spouse: 17.23% conversion (highest)
- Geographic: Edinburgh dominates (59.6% of records)

Feature-Level Account Creation Insights



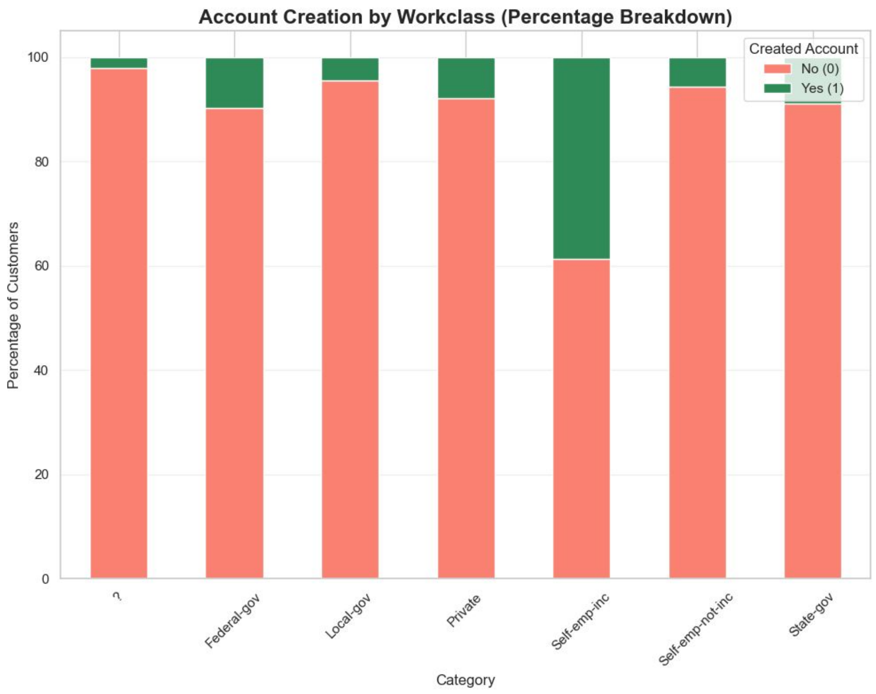


created_account	**0.0**	**1.0**	**Yes Count**
age_group			
age_10-19	100.00	0.00	0
age_20-29	98.12	1.88	7
age_30-39	**91.33**	**8.67**	**36**
age_40-49	**87.22**	**12.78**	**51**
age_50-59	85.98	14.02	30
age_60-69	87.50	12.50	11
age_70-79	96.00	4.00	1

----

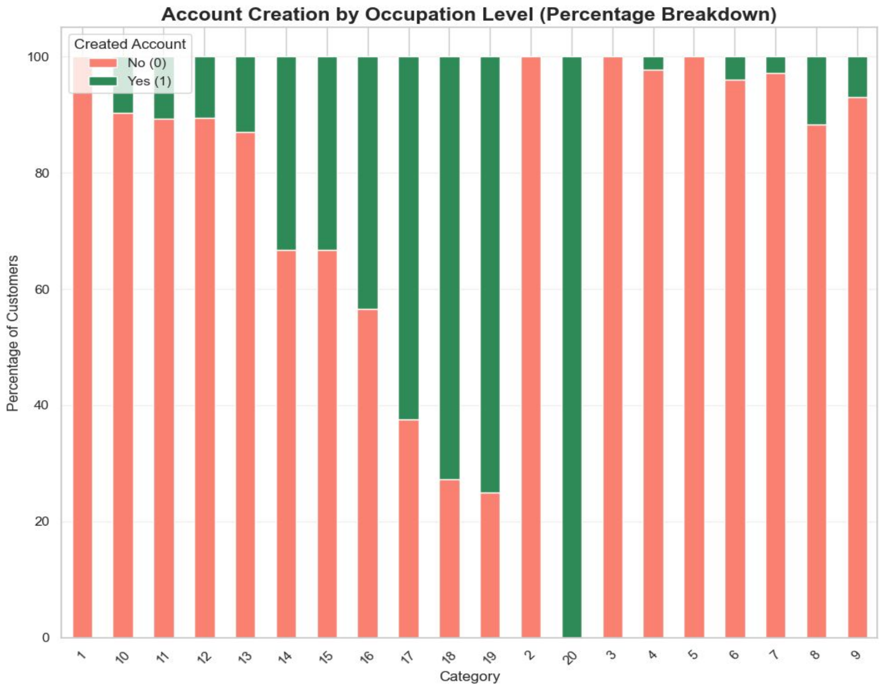
created_account	**0.0**	**1.0**	**Yes Count**
education			
edu_Assoc-acdm	91.49	8.51	4
edu_Assoc-voc	90.67	9.33	7
edu_Bachelors	**86.09**	**13.91**	**37**
edu_Doctorate	50.00	50.00	10
edu_HS-grad	**95.10**	**4.90**	**26**
edu_Masters	78.16	21.84	19

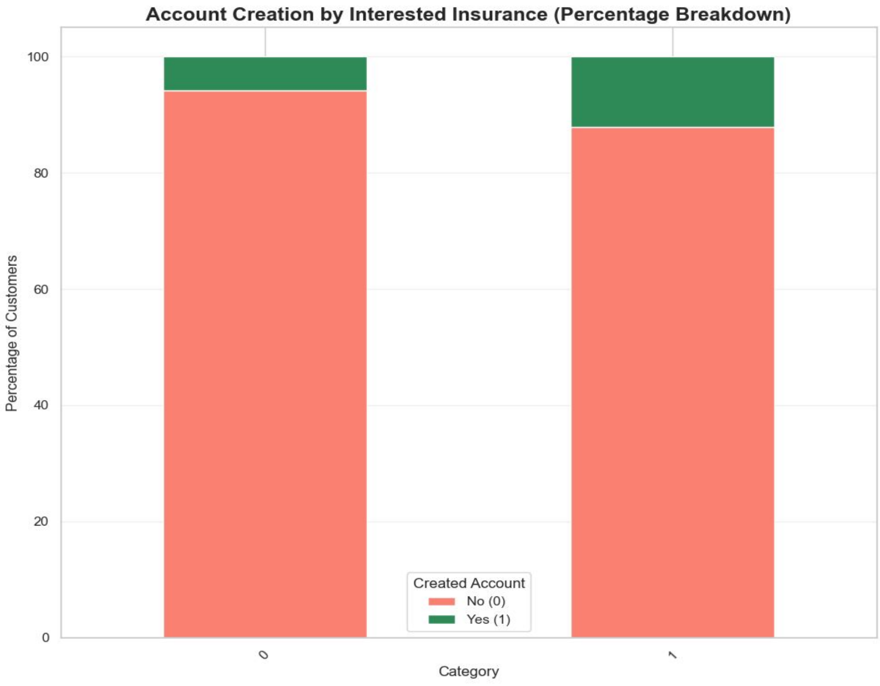
Feature-Level Account Creation Insights



created_account	0.0	1.0	Yes Count
category			
Local-gov	95.56	4.44	4
Private	**92.09**	**7.91**	**89**
Self-emp-inc	**61.29**	**38.71**	**24**
Self-emp-not-inc	94.31	5.69	7
State-gov	91.18	8.82	6

Feature-Level Account Creation Insights





Exploratory Insight: Age & Education

Age Group Pattern

- **Peak conversion** occurs in the **40–59 age range**, with:
    - `age_40-49`: **12.78%** conversion (51 accounts created)
    - `age_50-59`: **14.02%** conversion (30 accounts created)
- **Young adults (20–39)** show moderate interest:
    - `age_30-39`: **8.67%** conversion (36 accounts)
    - `age_20-29`: only **1.88%** (7 accounts), despite likely high volume
- **Older groups (70+) and teens (10–19)** show **negligible or zero conversion**, possibly due to eligibility, digital literacy, or product relevance.

Implication: Targeting middle-aged segments (30–59) may yield better ROI. Consider tailored messaging for younger adults to boost engagement.

Education Level Patterns

- **Highest conversion rates**:
    - `edu_Prof-school`: **47.37%**
    - `edu_Doctorate`: **50.00%**
    - `edu_Masters`: **21.84%**
- **Strong volume contributors**:
    - `edu_Bachelors`: **13.91%** (37 accounts)
    - `edu_HS-grad`: **4.90%** (26 accounts)
    - `edu_Some-college`: **6.46%** (23 accounts)
- **Low or zero conversion**:
- `edu_10th`, `edu_Preschool`: **0%**
- `edu_11th`: **1.33%**

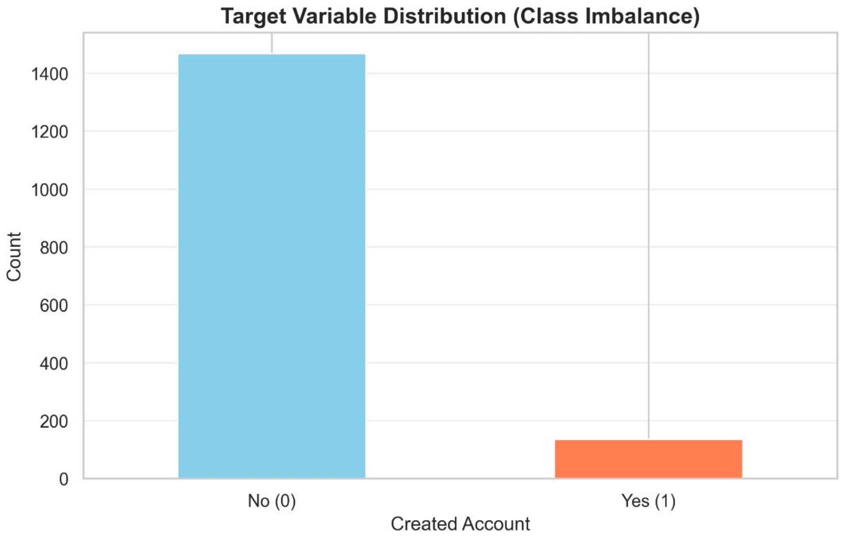
Implication: Higher education correlates with stronger conversion. Consider segmenting campaigns by education level and emphasizing product value for educated users.

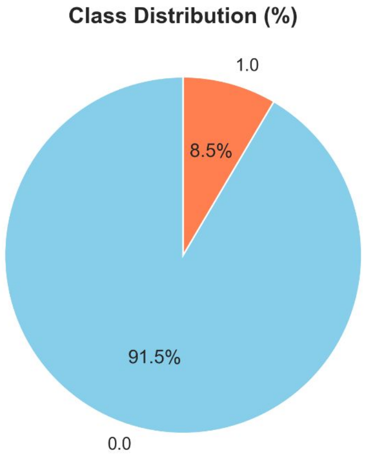
Combined Insight

- *Users aged **40–59** with **Bachelor's or higher degrees** are the most responsive.*
- ***Doctorate and professional school graduates** show exceptional conversion rates despite lower volume — ideal for premium offerings.*

Target Distribution & Challenge

Class	Count	Percentage
0 (No)	1468	91.5%
1 (Yes)	136	8.5%





PART 4: MODEL AND FEATURE SELECTION

What Types of Model Could We Try?

Models Selected:

1. Logistic Regression
- Why: Interpretable, fast, good baseline
- Hyperparameters: C=0.5, L2 penalty, liblinear solver, class_weight='balanced'
- Use Case: When interpretability is critical

2. Random Forest
- Why: Handles non-linear relationships, feature interactions
- Hyperparameters: 1000 trees, class_weight='balanced'
- Use Case: Maximum predictive performance

Both models:
- Use sklearn Pipeline for production-ready implementation
- Same preprocessing pipeline (numeric + categorical)
- Reproducible (random_state=1234)

What Features Do We Want to Use?

1. Demographic Features:
- Age (binned into 10-year groups)
- Sex (encoded: Male=1, Female=0)
- Marital status, education (one-hot encoded)
- Education number (one-hot encoded)

2. Geographic: Town (frequency encoded)

3. Employment:
- Job title (label encoded)
- Occupation level
- Employment duration (years + months combined, capped, square root transformed)

4. Financial:
- Salary (parsed, converted to annual GBP)
- Net profit (capital_gain - capital_loss, capped, square root transformed)

6. Categorical (One-Hot):
- Religion, relationship, workclass, race, native country

7. Derived:
- Demographic characteristic (quantile binned into 8 groups)

Excluded: Identifiers, name fields, dates, high-cardinality categoricals (>50 unique values)

PART 5: MODEL EVALUATION

What Measures Should We Look At?

1. Classification Metrics:

- Precision, Recall, F1-Score
- Confusion Matrix

2. Ranking Metrics:

- **ROC AUC**: Overall model performance
  - Logistic Regression: 0.9699
  - Random Forest: 0.9936
- **Precision@K**: Top-K precision for ranking
  - Precision@50: 0.48 (LogReg), 0.54 (RF)
  - Precision@100: 0.27 (both)
  - Precision@200: 0.135 (both)

Model Performance Comparison

Metric	Logistic Regression	Random Forest
ROC AUC	0.9699	**0.9936**
Precision (Class 1)	0.50	0.94
Recall (Class 1)	0.85	0.63
F1-Score (Class 1)	0.63	0.76
Best For	High-recall use cases (identify all potential customers)	High-precision use cases (minimize false positives)

Recommendation:

- Use **Random Forest for maximum performance**
- Use Logistic Regression when interpretability is critical

What Thresholds Are We Targeting?

Default Threshold: **0.5** (standard for binary classification)

Optimal Threshold Selection:

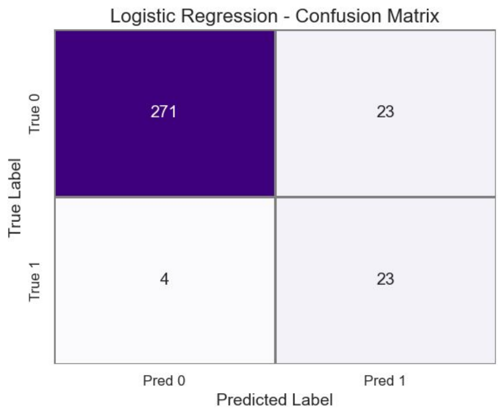
- Function: `find_optimal_threshold()` optimizes for F1, precision, or recall
- Business-dependent: Choose based on priorities
    - High Precision: Minimize false positives (save marketing costs)
    - High Recall: Maximize true positives (don't miss opportunities)

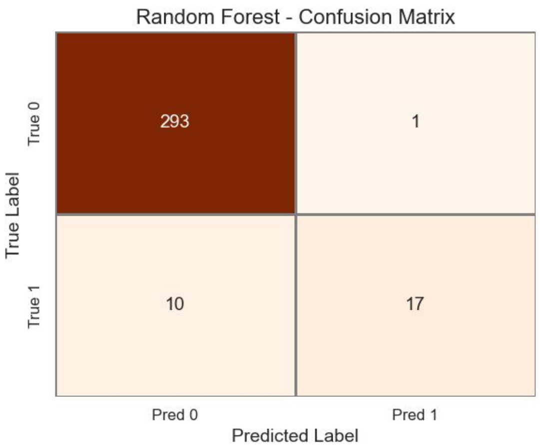
Example Results:

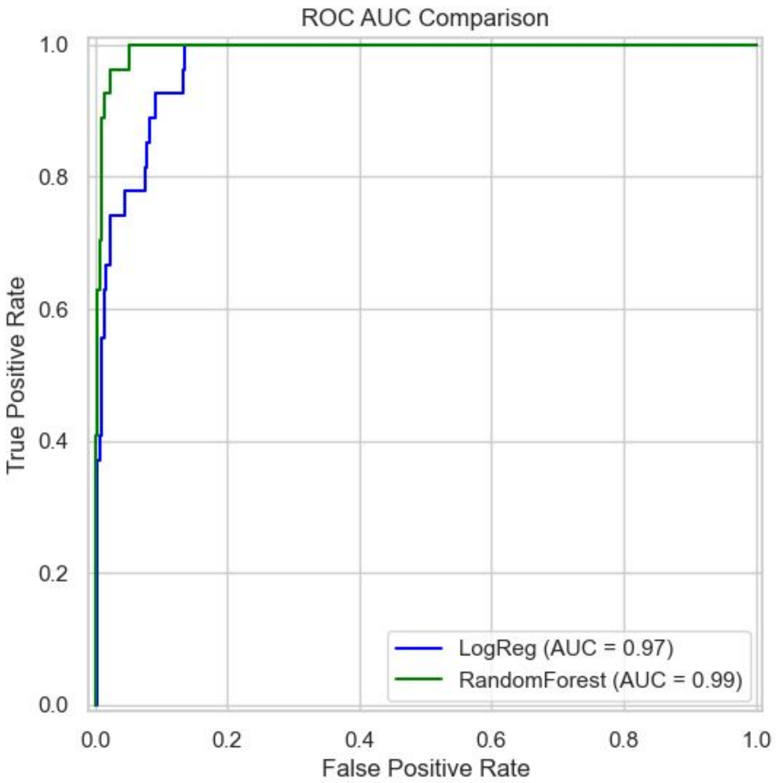
- Optimal F1 threshold: ~0.4-0.5 (varies by model)
- Can be tuned based on campaign budget and goals

Key Insights:

- Random Forest has higher ROC AUC and better top-k precision.
- Logistic Regression offers slightly higher recall for positives.
- Both models capture key financial, demographic, and engagement signals.







Predicted New Accounts -Threshold Scenarios

📌 Steps Used

1. Prepared the target customer dataset
   - Removed the created_account column
   - Aligned feature columns with the training dataset
2. Generated prediction scores
   - Random Forest score: score_rf
   - Logistic Regression score: score_logreg
3. Ranked top customers- Identified Top 100 high-probability customers

- Total base evaluated: 14,987 unlabeled customers
- Next step: pick threshold + model depending on budget/precision goals
- Top 100 can also be ranked directly via score_rf (Precision@100 ≈ 27%)

Random Forest

- Threshold 0.5 → 771 / 14,987 predicted accounts (5.1% conversion)
- Threshold **0.4** → 923 / 14,987 predicted accounts (6.16% conversion)
- Use 0.5 when we want higher precision (smaller, confident list); use 0.4 when we want a larger campaign pool

Logistic Regression

- Threshold 0.5 → 1,932 / 14,987 predicted accounts (12.9% conversion)
- Threshold **0.4** → 2,224 / 14,987 predicted accounts (14.8% conversion)
- LogReg is more generous; use when we prefer recall/coverage

Predicted creators Random Forest, Threshold =0.4

Predicted creators by age group:

age_group	
40-49	**340**
30-39	**250**
50-59	202
60-69	76
20-29	37
70-79	15
90-99	2
10-19	1

Predicted creators by education:

education	
Bachelors	**268**
HS-grad	**156**
Some-college	142
Masters	131
Prof-school	71
Doctorate	50
Assoc-voc	42
Assoc-acdm	26
11th	12
7th-8th	7
12th	5
9th	4
10th	3
5th-6th	3
1st-4th	2
Preschool	1

Predicted creators by marital status (RF):

marital_status	
Married-civ-spouse	**771**
Never-married	66
Divorced	61
Separated	11

Predicted creators Logistic regression, Threshold =0.4

📘 **Predicted creators by education (LogReg):**

education	
Bachelors	**570**
Some-college	393
HS-grad	334
Masters	332
Prof-school	218
Doctorate	136
Assoc-voc	111
Assoc-acdm	83
11th	13
12th	11
9th	7
Preschool	4
1st-4th	4
10th	3
5th-6th	3
7th-8th	2

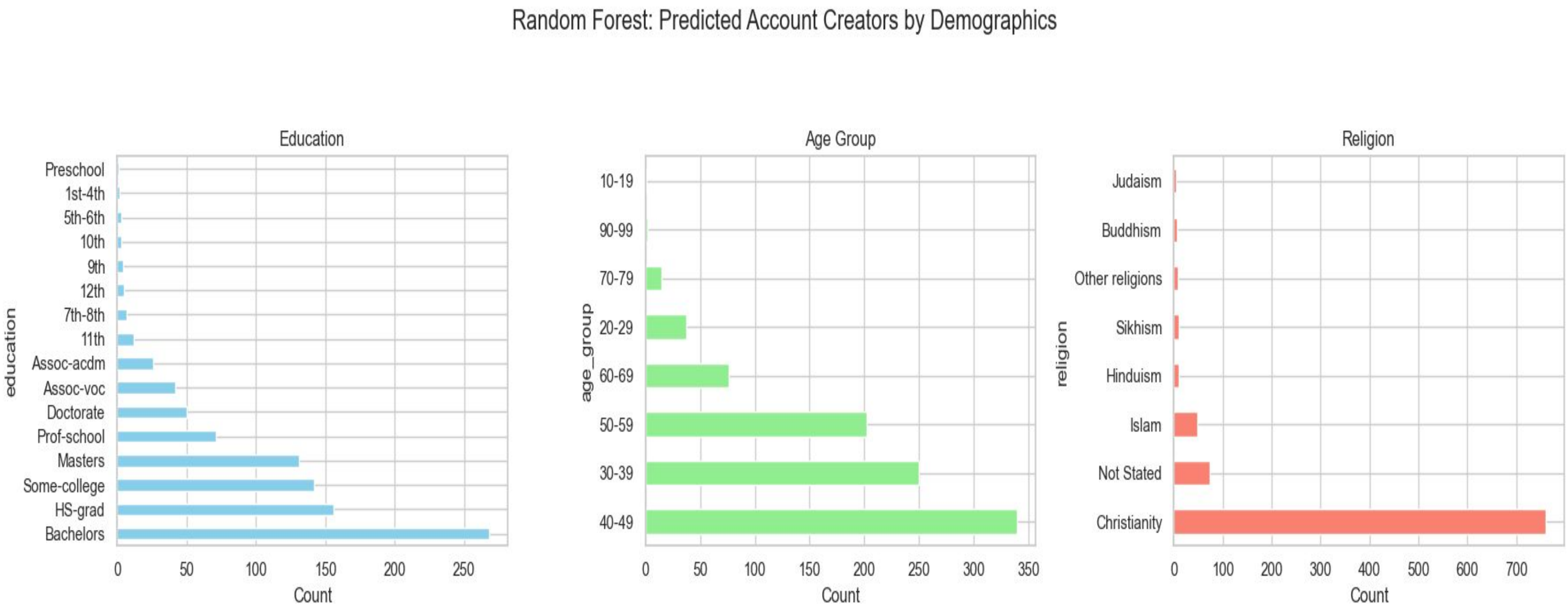
Predicted creators by age group (LogReg):

age_group	
40-49	**747**
30-39	**584**
50-59	497
20-29	193
60-69	170
70-79	23
10-19	5
80-89	3
90-99	2

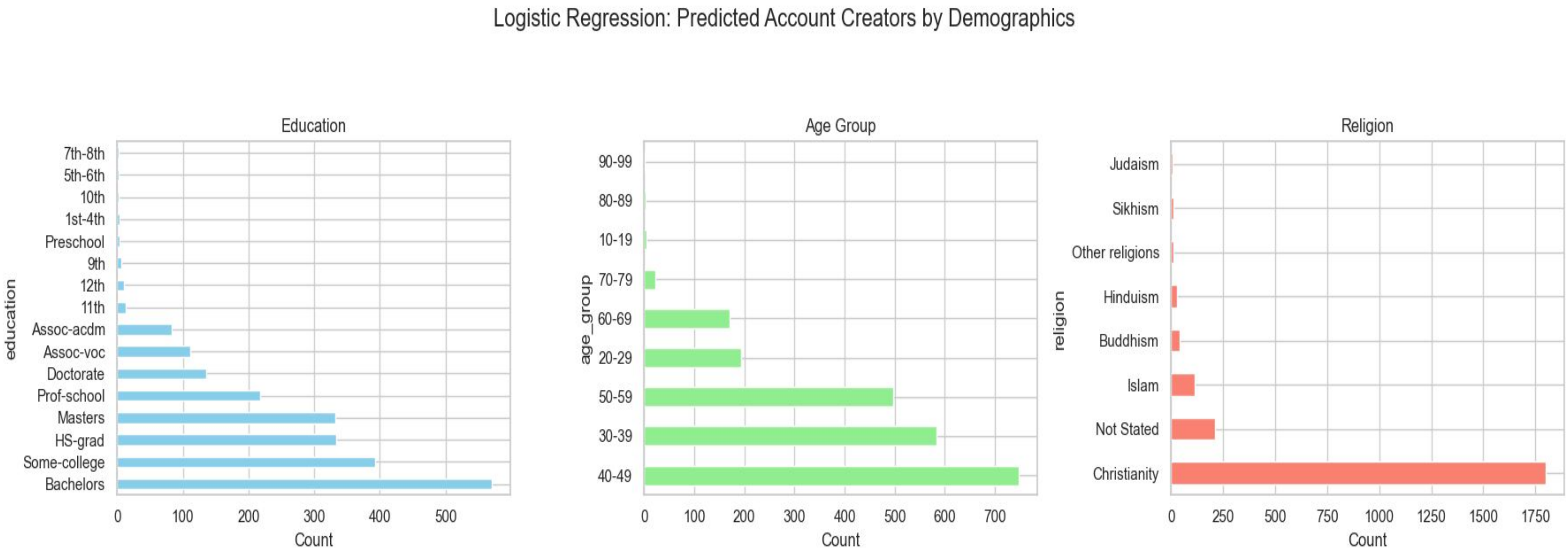
Predicted creators by marital status (LogReg):

marital_status	
Married-civ-spouse	1897
Never-married	139
Divorced	112
Widowed	46

Predicted creators by Demographics random Forest



Predicted creators by Demographics Logistic Regression



PART 6: PRODUCTIONISATION

How Can We Productionise the Model?

- **Modular Structure:**
  - `src/load_data.py` - Data loading
  - `src/data_cleaning.py` - Data cleaning
  - `src/feature_engineering.py` - Feature engineering
  - `src/model_train.py` - Model training
  - `src/evaluation.py` - Evaluation
  - `run_pipeline.py` - End-to-end pipeline

- **Production-Ready Features:**
  - Functions instead of notebook cells
  - Type hints throughout
  - Error handling
  - Configuration management (`src/config.py`)

Functions, Unit Tests, and Best Practices

- **Unit Tests:**
  - Comprehensive test suite (`tests/` directory)
  - Integration tests for full pipeline
  - Test coverage reporting

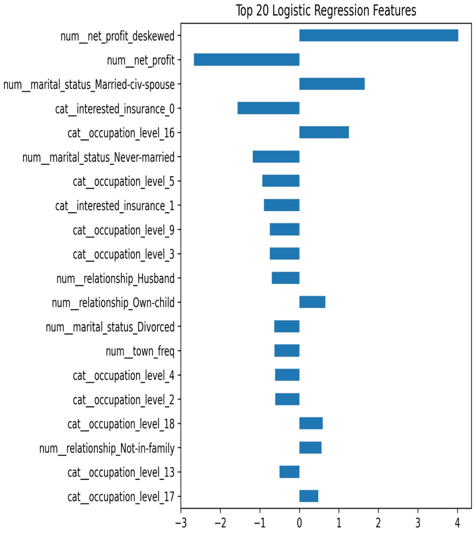
- **Best Practices:**
  - Type hints for all functions
  - Comprehensive documentation
  - Reproducible (fixed random seeds)
  - Version control ready

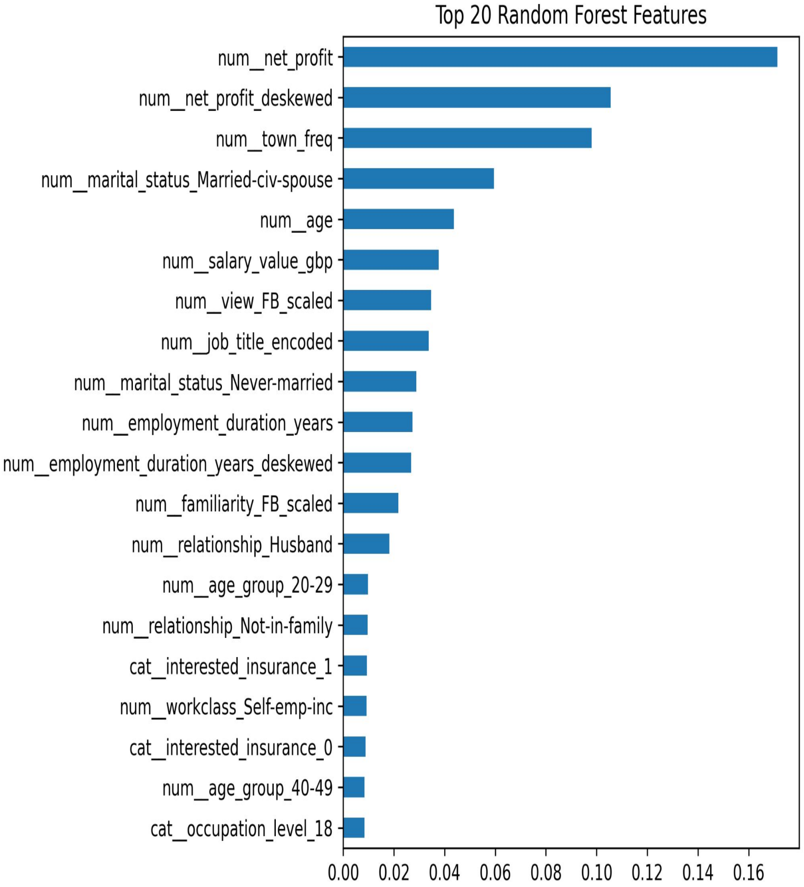
Implementation:
- Test files: `tests/test_*.py`
- Run tests: `pytest --cov=src`
- Documentation: `README.md`, `docs/` directory

PART 7: MODEL EXPLAINABILITY

How Can We Explain the Model?

1. **Feature Importance:**
   - Logistic Regression: Coefficient magnitudes
   - Random Forest: Feature importances
   - Visualization: Top 20 features plotted
   - Saved to: `output/explainability_plots/{model}_feature_importance.png`





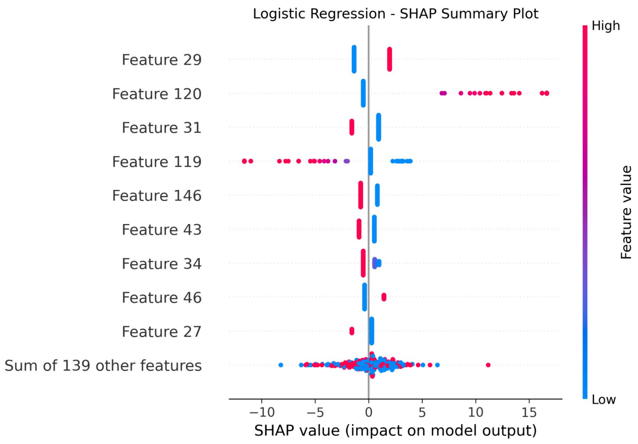
How Can We Explain the Model? cont..

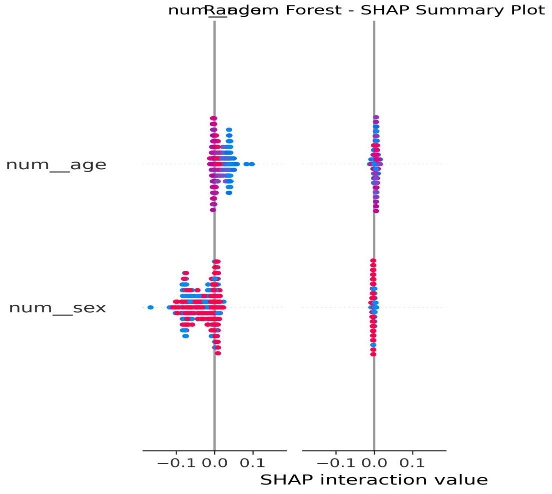
2. SHAP Values:

- Random Forest: Tree Explainer (fast, exact)
- Logistic Regression: Explainer (linear model)
- Visualization: Summary plots showing feature contributions
- Saved to: `output/explainability_plots/{model}_shap_summary.png`

3. Model Coefficients:

- Logistic Regression coefficients show feature impact
- Positive coefficients increase probability
- Negative coefficients decrease probability





Rationale for Marketing Department

Key Insights from Explainability:

- Age groups (30-59 highest)
- Education level (higher education = higher conversion)
- Marital status (married = higher conversion)
- Employment duration
- Financial indicators (salary, net profit)

Business Rationale:

- Target customers aged 30-59 with higher education and stable employment
- Married customers show 17% conversion vs 0.6% for never-married
- Higher salary and employment duration correlate with conversion

Regulatory Compliance:

- SHAP values provide transparent feature contributions
- Can explain individual predictions
- Feature importance shows model is not using protected attributes inappropriately

Explainability for Customers and Regulators

Customer-Facing Explanations:

- Your profile matches customers with X% likelihood of interest
- Can show which factors contribute positively/negatively
- Transparent decision-making process

Regulatory Compliance:

- Model documentation in `docs/model_card.md`
- Feature importance analysis
- Bias assessment across demographic groups

Implementation: `docs/model_card.md` - Comprehensive model documentation

Key Achievements

Data Pipeline:
- Successfully merged 16,591 records from two datasets
- Comprehensive feature engineering (100+ features)
- Production-ready data processing

Model Performance:
- Random Forest: 0.99 ROC AUC
- Logistic Regression: 0.97 ROC AUC
- Precision@50: 0.48-0.54

Production Readiness:
- Modular, tested codebase
- Comprehensive documentation
- Model explainability tools

Business Value:
- Data-driven customer selection
- Higher conversion rates expected
- Transparent, explainable decisions

Thank You and Questions