🎯 Comprehensive Feature Engineering & FP Reduction Strategy Plan

Executive Summary

Based on your distribution analysis, here's a **strategic roadmap** for reducing FPs while maintaining/improving recall through intelligent temporal feature engineering across your LR → RF → RNN cascade.

🔬 Key Statistical Insights from Your Distributions

Critical Findings:

1. **TenureBucket 0 (0-11 months): 48.3% churn rate** ← MASSIVE signal
2. **Contract Type: Month-to-month 42.7% churn** vs One year 11.3% vs Two year 2.8%
3. **OnlineSecurity: "No" = 46% churn** vs "Yes" = 15% churn (protective!)
4. **UsageSlope: Distinct bimodal pattern** - churners cluster in specific ranges
5. **Tenure: Strong negative correlation** (Cohen's d = -0.852, large effect)
6. **MonthlyCharges: Churners have higher charges** (Cohen's d = -0.487, medium effect)
7. **Service bundles show protective effects** - but only when sustained

Gender: DROP IT ✂️

- Near-identical churn rates (26.9% vs 26.2%)
- Chi-square will show no significance
- Adds noise, reduces generalizability

🏗️ Strategic Feature Engineering Framework

Philosophy: Temporal Stability vs. Temporal Volatility

Your FPs are likely **stable customers showing temporary anomalies**. We need features that distinguish:

- ✅ **True churners**: Sustained instability + risk accumulation

- ❌ **False positives**: Temporary spikes in otherwise stable customers

---

📊 Stage 1: Logistic Regression (LR) Features

Goal: Capture linear relationships & establish baseline interpretability

Feature Set (8-10 features):

1. Temporal Stability Indicators

````
Tenure_Maturity_Score
= tenure / (tenure + 12)  # Asymptotic, values 0→1
# Penalizes early-stage customers, protects veterans

Early_Critical_Period
= 1 if tenure < 3 else 0  # First 3 months = 48.3% churn zone

Tenure_Contract_Lock
= Contract * log1p(tenure)  # Interaction: long-term contracts gain strength with time
```

#### **2. Financial Stability Signals**
```
Charges_Consistency
= 1 - abs(MonthlyCharges - (TotalCharges / max(tenure, 1))) / MonthlyCharges
# Measures if charges are consistent over time
# High = stable billing, Low = volatility

Premium_Early_Exit_Risk
= (MonthlyCharges > median) & (tenure < 12)
# High-paying early customers = churners or high-value?

Financial_Commitment
= TotalCharges / (MonthlyCharges + 1e-5)
# How much have they actually paid? (proxy for investment)
```

#### **3. Service Protection Score**
```
Core_Security_Bundle
= OnlineSecurity + TechSupport + OnlineBackup
# Your analysis shows these are HIGHLY protective
````

Service_Dependency
= (StreamingTV + StreamingMovies) / 2
Entertainment stickiness (but weaker than security)

No_Internet_Safety_Flag
= 1 if InternetService == 0 else 0
Your data shows "No internet" = 7.4% churn (super safe!)
```

#### **4. Contract Strength**
```
Contract_Inverted
= 2 - Contract # Flip so month-to-month = high risk
Direct encoding of 42.7% → 11.3% → 2.8% pattern

Payment_Stability
= 1 if PaymentMethod in [3, 4] else 0 # Auto payments = stable
E-check (45.3%) vs Credit card (15.7%) churn
```

**LR Feature List (10 total):**
1. Tenure_Maturity_Score
2. Early_Critical_Period
3. Tenure_Contract_Lock
4. Charges_Consistency
5. Financial_Commitment
6. Core_Security_Bundle
7. Contract_Inverted
8. Payment_Stability
9. No_Internet_Safety_Flag
10. MonthlyCharges (raw, as baseline)

---

## 🌲 Stage 2: Random Forest (RF) Features
**Goal:** Capture non-linear interactions & identify complex patterns

### **Feature Set (12-15 features):**

#### **5. Temporal Transition Features** (alluvial-inspired!)
```
Tenure_Velocity
= tenure / TenureBucket # How fast did they reach this bucket?
Fast = committed, Slow = lingering risk

Tenure_Momentum
= 1 if TenureBucket >= 2 else 0 # Past the "danger zone"
Your data: Bucket 0 = 48.3%, Bucket 2 = 22.0%, Bucket 4 = 15.0%

Stability_Trajectory
= (tenure > 12) * Core_Security_Bundle * (Contract > 0)
Multi-factor lock-in: passed critical period + security + contract
```

#### **6. Usage Pattern Sophistication**
```
UsageSlope_Percentile
= percentile_rank(UsageSlope) # Normalize distribution
Your UsageSlope is bimodal - capture position in distribution

UsageSlope_Tenure_Alignment
= (UsageSlope - median(UsageSlope)) * (tenure - median(tenure))
Are usage and tenure aligned? (both high = stable, mismatch = risk)

High_Usage_Early_Churn_Trap
= (UsageSlope > 75th_percentile) & (tenure < 6)
Your scatter plot shows red dots in high-usage/low-tenure zone
```

#### **7. Service Evolution Features**
```
Service_Complexity
= sum([MultipleLines, OnlineSecurity, OnlineBackup,
      DeviceProtection, TechSupport, StreamingTV, StreamingMovies])
Total service engagement (0-7 scale)

Security_vs_Entertainment_Ratio
= Core_Security_Bundle / (Service_Dependency + 1)
Prioritize security (protective) vs entertainment (neutral)

Service_Underutilization
= (InternetService == 2) & (Core_Security_Bundle == 0)
Fiber optic (41.9% churn) WITHOUT security = high risk
```

#### **8. Financial Pressure Indicators**
```
High_Charge_No_Commitment

= (MonthlyCharges > 75th_percentile) & (Contract == 0)
Paying premium prices without contract lock = flight risk

Spending_Acceleration
= MonthlyCharges / (TotalCharges / max(tenure, 1))
Are charges increasing? (>1 = acceleration = dissatisfaction?)

Value_Perception_Gap
= MonthlyCharges / (Service_Complexity + 1)
Paying a lot for few services = poor perceived value
```

#### **9. Risk Accumulation Score**
```
Multi_Risk_Convergence
= sum([
    Contract == 0, # Month-to-month
    tenure < 12, # Early stage
    OnlineSecurity == 0, # No protection
    PaymentMethod == 1, # E-check
    MonthlyCharges > median
]) # Count of risk factors (0-5)

Protective_Factor_Count
= sum([
    Contract > 0,
    tenure >= 12,
    Core_Security_Bundle > 0,
    Partner == 1,
    Dependents == 1
]) # Loyalty shields (0-5)

Net_Risk_Balance
= Multi_Risk_Convergence - Protective_Factor_Count
Negative = protected, Positive = at risk
```

**RF Feature List (15 total):**
1. All 10 from LR stage
2. Tenure_Velocity
3. Stability_Trajectory
4. UsageSlope_Percentile
5. UsageSlope_Tenure_Alignment
6. High_Usage_Early_Churn_Trap
```

7. Service_Complexity
8. Security_vs_Entertainment_Ratio
9. Service_Underutilization
10. High_Charge_No_Commitment
11. Spending_Acceleration
12. Value_Perception_Gap
13. Multi_Risk_Convergence
14. Protective_Factor_Count
15. Net_Risk_Balance

🧠 Stage 3: RNN Features (Temporal Sequences)
Goal: Model customer journey trajectories & state transitions

Feature Set (18-20 features + sequence encoding):

10. Alluvial Flow Features (THIS IS KEY FOR FP REDUCTION!)
```
State_Stability_Score
= (tenure > 12) * (Contract > 0) * (Core_Security_Bundle > 0) *
  (PaymentMethod in [3,4])
# Binary: are they in a "stable state"?

State_Transition_Count
= number of times features changed significantly
# (Requires temporal data or approximation via volatility proxies)

Loyalty_Streak_Length
= tenure * State_Stability_Score
# How long have they been in stable state?

Momentum_Direction
= sign(UsageSlope - rolling_mean(UsageSlope))
# Are they trending up or down?
```

11. False Positive Dampeners (CRITICAL!)
```
Veteran_Stability_Override
= (tenure > 24) * (Protective_Factor_Count >= 3)
# Long-term customers with multiple shields = UNLIKELY to churn
# Increase threshold for these customers
```

Sustained_Engagement_Flag
= (Core_Security_Bundle > 0) & (Contract > 0) & (tenure > 12)
Triple lock = very low churn probability

Temporary_Anomaly_Filter
= (MonthlyCharges > 80th_percentile) but (TotalCharges consistent)
Single-period spike in charges ≠ churn intent
```

#### **12. Trajectory Clustering Features**
```
Churn_Journey_Archetype
= k-means cluster on (tenure, UsageSlope, Contract, Core_Security_Bundle)
Identify customer journey patterns:
- Fast Exit (high churn)
- Slow Build (low churn)
- Service Expander (low churn)
- Price Shopper (high churn)

Distance_From_Stable_Archetype
= euclidean_distance(current_features, "stable_cluster_centroid")
How far are they from the stable customer profile?
```

#### **13. Temporal Consistency Metrics**
```
Behavioral_Consistency_Score
= 1 / (std_dev([MonthlyCharges, UsageSlope, Service_Complexity]) + 1)
Low variance = consistent behavior = stable customer

Lifecycle_Stage_Alignment
= match(TenureBucket, expected_services_per_bucket)
Are their services appropriate for lifecycle stage?
E.g., Bucket 0 with full services = committed
Bucket 3 with minimal services = risk
```

#### **14. Sequence-Based Features** (RNN-specific)
```
Rolling_3_Month_Risk_Trend
= slope of Multi_Risk_Convergence over last 3 tenure buckets
Is risk increasing or decreasing over time?

Service_Addition_Momentum

= change in Service_Complexity over time
Positive = expanding, Negative = contracting (red flag!)

Charges_Evolution_Pattern
= pattern match(MonthlyCharges_sequence, known_churn_patterns)
Does their charge history match churners?
```

**RNN Feature List (20 total):**
1. All 15 from RF stage
2. State_Stability_Score
3. Loyalty_Streak_Length
4. Veteran_Stability_Override
5. Sustained_Engagement_Flag
6. Temporary_Anomaly_Filter
7. Churn_Journey_Archetype (one-hot encoded, 4-5 clusters)
8. Distance_From_Stable_Archetype
9. Behavioral_Consistency_Score
10. Lifecycle_Stage_Alignment
11. Rolling_3_Month_Risk_Trend (approximated if no temporal data)
12. Service_Addition_Momentum

---

## 🎚️ Asymmetric Threshold Strategy

### **Stage 1: LR**
```
Base thresholds:
- Churn: 0.25 (sensitive, catch early signals)
- No-Churn: 0.75 (standard)
```

### **Stage 2: RF**
```
Adaptive thresholds based on Protective_Factor_Count:

If Protective_Factor_Count >= 3:
   Churn_threshold = 0.35 # Harder to trigger churn
Else if Protective_Factor_Count == 2:
   Churn_threshold = 0.28
Else:
   Churn_threshold = 0.22 # Extra sensitive for high-risk
```
```

Stage 3: RNN
```
Context-aware thresholds:

If Veteran_Stability_Override == 1:
    Churn_threshold = 0.45  # Very high bar

Else if Sustained_Engagement_Flag == 1:
    Churn_threshold = 0.35

Else if Early_Critical_Period == 1:
    Churn_threshold = 0.18  # Extra sensitive

Else:

    Churn_threshold = 0.25  # Standard
```

📈 Expected FP Reduction Mechanisms

How These Features Reduce FPs:

1. **Veteran_Stability_Override** → Catches long-term customers with temporary anomalies
2. **Temporary_Anomaly_Filter** → Distinguishes one-time spikes from sustained risk
3. **Behavioral_Consistency_Score** → Rewards stable patterns over time
4. **Protective_Factor_Count** → Multi-factor authentication for churn prediction
5. **Loyalty_Streak_Length** → Penalizes predicting churn for sustained stable customers
6. **State_Stability_Score** → Alluvial-inspired: requires sustained state change
7. **Adaptive thresholds** → Dynamically adjust sensitivity based on customer profile

Expected Outcomes:

Metric	Current	Target (Conservative)	Target (Optimistic)
Churn Precision	0.5126	0.62-0.65	0.68-0.72
Churn Recall	0.7973	0.78-0.80	0.82-0.85
FP Count	226.8	170-190	140-160
F1 Score	0.7154	0.74-0.76	0.77-0.80

🛠️ Implementation Roadmap

Phase 1: Data Preparation (Day 1, Morning)

1. ✅ Drop `gender`
2. Create all **Stage 1 features** (10 features)
3. Validate distributions & correlations
4. Check for multicollinearity (VIF < 5)

Phase 2: LR Baseline (Day 1, Afternoon)

1. Train LR with 10 features
2. Tune `C` parameter (0.01, 0.1, 1.0)
3. Test asymmetric threshold (0.20-0.30 range)
4. Establish baseline metrics
5. Feature importance analysis

Phase 3: RF Enhancement (Day 2, Morning)

1. Create **Stage 2 features** (15 total)
2. Train RF with hyperparameter tuning:
   - `n_estimators`: [200, 300, 500]
   - `max_depth`: [10, 15, 20]
   - `min_samples_split`: [10, 20, 30]
3. Implement adaptive thresholds
4. Measure FP reduction vs LR

Phase 4: RNN Simulation (Day 2, Afternoon)

1. Create **Stage 3 features** (20 total)
2. Implement k-means clustering for journey archetypes
3. Create pseudo-temporal features
4. Train enhanced RF as RNN proxy
5. Implement context-aware thresholds

Phase 5: Cascade Integration (Day 3)

1. Build LR → RF → RNN pipeline
2. Implement stage-wise feature passing
3. Test full cascade on validation set
4. Analyze FP/FN breakdown by feature combination

5. Fine-tune thresholds per stage

Phase 6: Analysis & Visualization (Day 3-4)

1. Create alluvial plots showing customer journey transitions
2. FP analysis: which features caused misclassification?
3. Feature stability analysis across stages
4. ROC/Precision-Recall curves per stage
5. Business impact calculation (cost-benefit)

🎯 Key Success Metrics

Primary Goals:

- ✅ Reduce FP by 15-20% (from 226.8 → 170-190)
- ✅ Maintain recall ≥ 78%
- ✅ Increase precision to 0.62-0.65

Secondary Goals:

- ✅ Improve interpretability (feature importance + SHAP)
- ✅ Demonstrate generalizability (features applicable to other domains)
- ✅ Reduce inference time vs pure neural network

🔬 Research Contribution Angles

1. Domain-Specific Cascade Architecture

- Show that **specialized stages** outperform single black-box models
- Prove **interpretability ≠ performance trade-off**

2. Temporal Stability Features

- Introduce **alluvial-inspired transition tracking** for churn
- Demonstrate **false positive reduction** via temporal consistency

3. Asymmetric Threshold Optimization

- Context-aware thresholds based on **customer lifecycle stage**

- Show **adaptive sensitivity** improves precision without killing recall

4. Generalizability Framework

- Features designed for **cross-domain applicability**:
  - Subscription services (SaaS, streaming)
  - Financial services (banking, insurance)
  - Retail (loyalty programs)
- Prove features transfer to **non-telecom datasets**