NYOHAKI GROUP

Lead Generation AI

Lead generation in the insurance system is key to drive market share being held and distributed effectively .  
This document is property and all rights are held by NYOHAKI GROUP .

****

**Insurance Lead Scoring Models Documentation**

**Table of Contents**

1. Overview.

2.Base Insurance Lead Scoring Model.

3. Healthcare Insurance Lead Scoring Model

4. Life Insurance Lead Scoring Model.

5. API Documentation.

6. Model Evaluation.

7. Compliance & Ethics.

8. Deployment Guide.

**Overview**

The Insurance Lead Scoring System is a comprehensive machine learning platform designed to evaluate and score insurance leads across multiple product lines. The system provides accurate, fair, and compliant lead scoring with specialized models for different insurance types.

**System Architecture**

* Insurance Lead Scoring System
* Base Insurance Model (Foundation)
* Healthcare Insurance Model (Specialized)
* Life Insurance Model (Specialized)
* API Layer (FastAPI).
* Evaluation Framework
* Compliance Engine

**Key Features**

* Multi-Product Support: Base, Healthcare, and Life Insurance
* Real-time Scoring: Sub-second response times
* Compliance-First: GDPR, CCPA, TCPA compliant
* Bias Detection: Automated fairness monitoring
* Interpretable Results: Feature importance and reasoning
* Scalable Architecture: Batch and real-time processing

**Base Insurance Lead Scoring Model**

**Purpose**

The base model serves as the foundation for all insurance lead scoring, providing core functionality and common features across insurance products.

**Model Details**

* Algorithm: XGBoost Regressor
* Output: Conversion probability score (0-100)
* Training Data: General insurance leads with consent
* Update Frequency: Monthly retraining

**Core Feature  
python**

feature\_columns = [

'age', 'income', 'location', 'employment\_status', 'credit\_score',

'previous\_insurance', 'contact\_method', 'lead\_source', 'time\_of\_contact',

'interaction\_history', 'demographic\_segment', 'risk\_profile'

]

```

**Feature Engineering**

* Risk Assessment: Credit score normalization and risk categorization
* Demographic Scoring: Age-income interaction effects
* Behavioral Patterns: Contact timing and interaction history
* Lead Quality: Source quality and engagement metrics

**Model Performance**

* Target Accuracy: RÂ² > 0.85
* Response Time: < 100ms per lead
* Batch Processing: 10,000+ leads per minute
* Fairness Threshold: Demographic parity < 0.1

**Usage Example**

**Python**

from models.insurance\_lead\_scoring.inference import InsuranceLeadScorer

scorer = InsuranceLeadScorer()

result = scorer.score\_lead({

'lead\_id': 'BASE\_001',

'age': 35,

'income': 75000,

'employment\_status': 'employed',

'credit\_score': 720,

'consent\_given': True

})

**Healthcare Insurance Lead Scoring Model**

**Purpose**

Specialized model for healthcare insurance leads, incorporating health-specific factors and regulatory compliance requirements.

**Model Details**

* Algorithm: XGBoost Regressor (Healthcare-optimized)
* Output: Healthcare conversion score (0-100) + health insights
* Training Data: Healthcare insurance leads with HIPAA compliance
* Update Frequency: Bi-weekly retraining

**Healthcare-Specific Features**

```python

healthcare\_features = [

'age', 'income', 'family\_size', 'employment\_status', 'current\_insurance\_status',

'health\_conditions', 'prescription\_medications', 'healthcare\_utilization',

'preferred\_provider\_network', 'coverage\_preferences', 'deductible\_preference',

'chronic\_conditions', 'preventive\_care\_usage', 'specialist\_needs'

]

**Advanced Feature Engineering**

* Health Risk Scoring: Chronic conditions and medication analysis
* Network Preferences: Provider network alignment scoring
* Cost Sensitivity: Deductible and premium preference modeling
* Utilization Patterns: Healthcare usage prediction
* Family Coverage: Dependent coverage needs assessment

**Healthcare-Specific Outputs**

**```python**

{

'score': 87.5,

'health\_risk\_category': 'MODERATE',

'recommended\_plan\_type': 'PPO',

'estimated\_annual\_cost': 8400,

'network\_preference\_match': 0.92,

'chronic\_condition\_coverage': ['diabetes', 'hypertension'],

'preventive\_care\_score': 8.2,

'urgency\_level': 'HIGH'

}

```

**Compliance Features**

* HIPAA Compliance: Health data anonymization and encryption
* ACA Compliance: Non-discrimination in health status
* State Regulations: State-specific healthcare requirements
* Privacy Protection: PII masking and secure processing

**Performance Metrics**

* Overall Accuracy: RÂ² > 0.88
* High-Risk Accuracy RÂ² > 0.85 for chronic conditions  
  Family Plan Accuracy: RÂ² > 0.90 for family coverage
* Network Match Accuracy: 94% provider preference alignment

**Life Insurance Lead Scoring Model**

**Purpose**

Specialized model for life insurance leads, incorporating mortality risk assessment, coverage adequacy analysis, and life stage considerations.

**Model Details**

* Algorithm: XGBoost Regressor (Life Insurance-optimized)
* Output: Life insurance score (0-100) + actuarial insights
* Training Data: Life insurance leads with actuarial data
* Update Frequency: Monthly retraining with mortality table updates

**Life Insurance Features**

**```python**

life\_insurance\_features = [

'age', 'income', 'marital\_status', 'dependents\_count', 'employment\_status',

'health\_status', 'smoking\_status', 'coverage\_amount\_requested', 'policy\_term',

'existing\_life\_insurance', 'beneficiary\_count', 'debt\_obligations',

'mortgage\_balance', 'education\_level', 'occupation\_risk\_level',

'life\_stage', 'financial\_dependents', 'estate\_planning\_needs'

]

```

**Actuarial Feature Engineering**

* Mortality Risk Scoring: Age, health, smoking, occupation risk
* Coverage Adequacy: Income replacement and debt coverage analysis
* Life Stage Analysis: Young professional to estate planning stages
* Financial Responsibility: Dependents and debt obligations
* Estate Planning: High-net-worth and inheritance considerations

**Life Insurance Outputs**

```python

{

'score': 92.3,

'life\_stage': 'family\_building',

'mortality\_risk\_score': 3.2,

'recommended\_coverage': 750000,

'coverage\_adequacy': 'adequate',

'coverage\_gap': 0,

'recommended\_policy\_type': 'TERM\_LIFE',

'urgency\_level': 'CRITICAL',

'estate\_planning\_urgency': 4.5

}

```

**Policy Recommendations**

* Term Life: Young families, temporary needs
* Whole Life: Estate planning, permanent coverage
* Universal Life: Flexible premiums, investment growth
* Variable Life: Investment-minded, high-income clients

**Actuarial Compliance**

* Mortality Tables: Industry-standard life expectancy data
* Non-Discrimination: Fair underwriting practices
* State Regulations: Insurance commissioner requirements
* Actuarial Fairness: Evidence-based risk assessment

**Performance Metrics**

* Overall Accuracy: RÂ² > 0.87
* Family Building Accuracy: RÂ² > 0.92 for family stage
* High Coverage Accuracy: RÂ² > 0.89 for $500k+ policies
* Policy Recommendation: 91% accuracy in policy type matching
* API Documentation
* Base Insurance API
* Endpoint: `/score-lead`

http

POST /score-lead

Content-Type: application/json

{

"lead\_id": "BASE\_001",

"age": 35,

"income": 75000,

"employment\_status": "employed",

"credit\_score": 720,

"consent\_given": true,

"consent\_timestamp": "2024-01-15T10:30:00Z"

}

```

\*\*Response:\*\*

```json

{

"lead\_id": "BASE\_001",

"score": 78.5,

"confidence": 0.92,

"risk\_category": "MEDIUM",

"timestamp": "2024-01-15T10:30:15Z",

"model\_version": "1.0\_base",

"compliance\_status": "PASSED"

}

```

**Healthcare Insurance API**

Endpoint: `/score-healthcare-lead`

```http

POST /score-healthcare-lead

Content-Type: application/json

{

"lead\_id": "HEALTH\_001",

"age": 42,

"income": 85000,

"family\_size": 4,

"current\_insurance\_status": "uninsured",

"health\_conditions": ["diabetes", "hypertension"],

"preferred\_provider\_network": "PPO",

"consent\_given": true

}

```

\*\*Response:\*\*

```json

{

"lead\_id": "HEALTH\_001",

"score": 87.5,

"health\_risk\_category": "MODERATE",

"recommended\_plan\_type": "PPO",

"estimated\_annual\_cost": 8400,

"network\_preference\_match": 0.92,

"urgency\_level": "HIGH",

"compliance\_status": "PASSED"

}

```

**Life Insurance API**

Endpoint: `/score-life-insurance-lead`

```http

POST /score-life-insurance-lead

Content-Type: application/json

{

"lead\_id": "LIFE\_001",

"age": 35,

"income": 85000,

"marital\_status": "married",

"dependents\_count": 2,

"health\_status": "good",

"smoking\_status": "non\_smoker",

"coverage\_amount\_requested": 750000,

"consent\_given": true

}

```

\*\*Response:\*\*

```json

{

"lead\_id": "LIFE\_001",

"score": 92.3,

"life\_stage": "family\_building",

"mortality\_risk\_score": 3.2,

"recommended\_coverage": 750000,

"recommended\_policy\_type": "TERM\_LIFE",

"urgency\_level": "CRITICAL",

"compliance\_status": "PASSED"

}

```

**Batch Processing APIs**

Endpoint: `/score-leads` (All Models)

```http

POST /score-leads

Content-Type: application/json

{

"leads": [

{"lead\_id": "001", "age": 35, ...},

{"lead\_id": "002", "age": 42, ...}

]

}

```

**Utility APIs**

**Coverage Calculator**

```http

GET /life-insurance-coverage-calculator?income=85000&dependents=2&mortgage=300000

```

**Health Risk Assessment**

```http

GET /healthcare-risk-assessment?age=42&conditions=diabetes,hypertension

```

**Model Evaluation**

**Evaluation Framework**

Each model includes comprehensive evaluation across multiple dimensions:

**Accuracy Metrics**

* RÂ² Score: Coefficient of determination
* RMSE: Root Mean Square Error
* MAE: Mean Absolute Error
* Precision/Recall: For binary classification thresholds

**Fairness Metrics**

* Demographic Parity: Equal positive rates across groups
* Equalized Odds: Equal TPR/FPR across groups
* Individual Fairness: Similar individuals receive similar scores

**Business Metrics**

* Conversion Accuracy: Actual vs predicted conversions
* Revenue Impact: Lead value optimization
* Cost Efficiency: Reduced manual review requirements

**Evaluation Reports**

**Base Insurance Evaluation**

python

from models.insurance\_lead\_scoring.evaluate import InsuranceModelEvaluator

evaluator = InsuranceModelEvaluator()

report = evaluator.generate\_evaluation\_report('data/test\_leads.csv')

**Compliance & Ethics**

**Data Privacy**

**GDPR Compliance**

* Consent Management: Explicit consent tracking
* Right to Erasure: Data deletion capabilities
* Data Minimization: Only necessary data collection
* Anonymization: PII masking in processing

**CCPA Compliance**

* Consumer Rights: Data access and deletion
* Opt-out Mechanisms: Marketing communication controls
* Transparency: Clear data usage disclosure

**HIPAA Compliance (Healthcare Model)**

* Health Data Protection: Encrypted health information
* Access Controls: Role-based data access
* Audit Trails: Complete data access logging

**Fairness & Bias Prevention**

**Protected Classes**

* Age, Gender, Race, Religion, Disability Status
* Geographic Location, Socioeconomic Status
* Health Status (where legally permissible)

**Bias Detection**

**python**

**Automated bias monitoring**

fairness\_metrics = {

'age\_bias': 0.05, # < 0.1 threshold

'income\_bias': 0.03,

'location\_bias': 0.07,

'overall\_bias\_alert': False

}

```

**Mitigation Strategies**

* Pre-processing: Bias-aware feature selection
* In-processing: Fairness constraints during training
* Post-processing: Score adjustment for fairness
* Monitoring: Continuous bias detection

**Regulatory Compliance**

**Insurance Regulations**

* State Insurance Codes: Compliance with state requirements
* NAIC Guidelines: National Association of Insurance Commissioners
* Fair Credit Reporting Act: Credit data usage compliance
* Equal Credit Opportunity Act: Non-discriminatory practices

**Model Governance**

* Model Documentation: Complete model lineage
* Validation Framework: Independent model validation
* Risk Management: Model risk assessment
* Audit Trail: Complete decision audit logs

NYOHAKI GROUP   
[WWW.NYOHAKI.CO](http://WWW.NYOHAKI.CO)  
(253)-268-0427  
WA, USA.