Heart Disease Risk Assessment - API Reference

Base URL

```
Production: https://heartdisease.duminduthushan.com
API Docs: https://heartdisease.duminduthushan.com/docs
```

Overview

RESTful API for cardiovascular risk assessment using machine learning ensemble models. All endpoints return JSON responses with comprehensive error handling.

Authentication

Currently open access for demonstration purposes. JWT authentication framework ready for implementation.

Rate Limiting

- **Standard Rate**: 100 requests per minute per IP address
- **Batch Processing**: Limited to 100 patients per request
- **Concurrent Connections**: Up to 50 simultaneous connections

Content Type

All requests and responses use `Content-Type: application/json`

API Endpoints

Health Check

Monitor service health and system status.

```http
GET /health
```

Response Example:

```json
```

```
{
  "status": "healthy",
  "timestamp": "2025-08-30T15:30:00Z",
  "version": "1.0.0",
  "model_loaded": true,
  "database_connected": true,
  "uptime_seconds": 86400
}
```

Response Fields:

- `status`: Service health status (healthy/degraded/unhealthy)
- `timestamp`: Current server timestamp in ISO format
- `version`: API version identifier
- `model_loaded`: ML model availability status
- `database_connected`: Database connectivity status
- `uptime_seconds`: Service uptime in seconds

Model Information

Retrieve detailed information about the ML model and its performance.

http

```
GET /model/info
```

Response Example:

json

```
{
  "model_name": "EnsembleClassifier",
  "model_version": "v1.0.0",
  "training_date": "2025-08-30",
  "performance_metrics": {
    "accuracy": 0.8689,
    "precision": 0.8125,
    "recall": 0.9286,
    "f1_score": 0.8667,
    "auc_roc": 0.9535
  },
  "feature_names": [
    "age", "sex", "cp", "trestbps", "chol", "fbs",
    "restecg", "thalach", "exang", "oldpeak",
    "slope", "ca", "thal", "age_group",
    "bp_category", "chol_category", "hr_reserve",
    "age_chol_interaction", "bp_age_interaction",
    "composite_risk"
  ],
  "total_features": 20,
  "training_dataset_size": 303
}
```

Single Patient Prediction

Assess cardiovascular risk for an individual patient.

http

```
POST /predict
Content-Type: application/json
```

Request Body:

json

```
{
  "age": 63,
  "sex": 1,
  "cp": 3,
  "trestbps": 145,
  "chol": 233,
  "fbs": 1,
  "restecg": 0,
  "thalach": 150,
  "exang": 0,
  "oldpeak": 2.3,
  "slope": 0,
  "ca": 0,
  "thal": 1
}
```

Parameter Definitions:

- `age`: Patient age (18-120 years)
- `sex`: Biological sex (0=Female, 1=Male)
- `cp`: Chest pain type (0=Typical angina, 1=Atypical angina, 2=Non-anginal pain, 3=Asymptomatic)
- `trestbps`: Resting blood pressure (80-250 mm Hg)
- `chol`: Serum cholesterol (100-600 mg/dl)
- `fbs`: Fasting blood sugar (0=≤120 mg/dl, 1=>120 mg/dl)
- `restecg`: Resting ECG (0=Normal, 1=ST-T abnormality, 2=LVH)
- `thalach`: Maximum heart rate achieved (60-220 bpm)
- `exang`: Exercise induced angina (0=No, 1=Yes)
- `oldpeak`: ST depression induced by exercise (0.0-10.0 mm)
- `slope`: Peak exercise ST segment slope (0=Upsloping, 1=Flat, 2=Downsloping)
- `ca`: Number of major vessels (0-4)
- `thal`: Thalassemia (1=Normal, 2=Fixed defect, 3=Reversible defect)

Response Example:

```
json
```

```
{
  "prediction": 1,
  "probability": 0.7845,
  "risk_percentage": 78.45,
  "risk_category": "high",
  "confidence": 89.23,
  "interpretation": {
    "sex": "Male",
    "cp": "Asymptomatic",
    "age_risk": "Advanced age (>65) - Higher risk factor",
    "cholesterol_risk": "High cholesterol (>240) - Major risk factor",
    "bp_risk": "High blood pressure (>140) - Major risk factor"
  },
  "recommendations": [
    "High risk detected - immediate medical consultation recommended",
    "Consider comprehensive cardiac evaluation",
    "High blood pressure detected - consult physician for management",
    "High cholesterol (>240) - Major risk factor",
    "Regular moderate exercise beneficial for heart health",
    "Maintain healthy diet rich in fruits and vegetables"
  ]
}
```

Response Fields:

- `prediction`: Binary prediction (0=No disease risk, 1=Disease risk)
- `probability`: Probability of heart disease (0.0-1.0)
- `risk_percentage`: Risk percentage (0-100%)
- `risk_category`: Risk stratification (low/medium/high)
- `confidence`: Model confidence score (0-100%)
- `interpretation`: Medical interpretation of key risk factors
- `recommendations`: Personalized health recommendations

Batch Patient Prediction

Process multiple patients simultaneously for population health analysis.

```http
POST /predict/batch
Content-Type: application/json
```

Request Body:

json

```
{
  "patients": [
    {
      "age": 63, "sex": 1, "cp": 3, "trestbps": 145, "chol": 233,
      "fbs": 1, "restecg": 0, "thalach": 150, "exang": 0,
      "oldpeak": 2.3, "slope": 0, "ca": 0, "thal": 1
    },
    {
      "age": 37, "sex": 0, "cp": 0, "trestbps": 110, "chol": 180,
      "fbs": 0, "restecg": 0, "thalach": 170, "exang": 0,
      "oldpeak": 0.5, "slope": 1, "ca": 0, "thal": 2
    }
  ],
  "return_detailed": true
}
```

Request Parameters:

- `patients`: Array of patient objects (max 100 patients)
- `return_detailed`: Include detailed analysis in response (optional, default: true)

Response Example:

json

```
{
  "total_patients": 2,
  "predictions": [
    {
      "prediction": 1,
      "probability": 0.7845,
      "risk_percentage": 78.45,
      "risk_category": "high",
      "confidence": 89.23,
      "interpretation": { /* detailed interpretation */ },
      "recommendations": [ /* personalized recommendations */ ]
    },
    {
      "prediction": 0,
      "probability": 0.1234,
      "risk_percentage": 12.34,
      "risk_category": "low",
      "confidence": 91.56,
      "interpretation": { /* detailed interpretation */ },
      "recommendations": [ /* personalized recommendations */ ]
    }
  ],
  "summary": {
    "total_processed": 2,
    "average_risk": 45.4,
    "risk_distribution": {
      "low": 1,
      "medium": 0,
      "high": 1
    },
    "high_risk_count": 1,
    "high_risk_percentage": 50.0
  }
}
```

Sample Data Endpoint

Retrieve sample patient data for testing and integration.

```http
GET /predict/sample
```

Response Example:

```json
{
  "sample_input": {
    "age": 63,
    "sex": 1,
    "cp": 3,
    "trestbps": 145,
    "chol": 233,
    "fbs": 1,
    "restecg": 0,
    "thalach": 150,
    "exang": 0,
    "oldpeak": 2.3,
    "slope": 0,
    "ca": 0,
    "thal": 1
  },
  "prediction": {
    "prediction": 1,
    "risk_percentage": 78.45,
    "risk_category": "high"
  }
}
```

Error Handling

HTTP Status Codes

- `200 OK`: Successful request
- `400 Bad Request`: Invalid input parameters
- `422 Unprocessable Entity`: Validation errors
- `429 Too Many Requests`: Rate limit exceeded
- `500 Internal Server Error`: Server processing error
- `503 Service Unavailable`: Service temporarily unavailable

Error Response Format

```json
```

```
{
  "error": "validation_error",
  "message": "Age must be between 18 and 120 years",
  "details": {
    "field": "age",
    "value": 150,
    "constraint": "must be <= 120"
  },
  "timestamp": "2025-08-30T15:30:00Z"
}
```

Common Error Types

- `validation_error`: Invalid input parameter values
- `rate_limit_exceeded`: Too many requests
- `model_unavailable`: ML model loading error
- `processing_error`: Internal prediction error

Integration Examples

Python Integration

```
python
```

```python
import requests
import json

def predict_heart_disease_risk(patient_data):
    """Predict heart disease risk for a patient."""
    url = "https://heartdisease.duminduthushan.com/predict"

    try:
        response = requests.post(url, json=patient_data, timeout=10)
        response.raise_for_status()
        return response.json()
    except requests.exceptions.RequestException as e:
        print(f"API request failed: {e}")
        return None

# Example usage
patient = {
    "age": 65,
    "sex": 1,
    "cp": 3,
    "trestbps": 160,
    "chol": 280,
    "fbs": 1,
    "restecg": 0,
    "thalach": 140,
    "exang": 1,
    "oldpeak": 2.5,
    "slope": 2,
    "ca": 1,
    "thal": 2
}

result = predict_heart_disease_risk(patient)
if result:
    print(f"Risk: {result['risk_percentage']:.1f}%")
    print(f"Category: {result['risk_category']}")
```

JavaScript Integration

```
javascript
```

```
async function predictHeartDiseaseRisk(patientData) {
  try {
    const response = await fetch('https://heartdisease.duminduthushan.com/predict', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify(patientData)
    });

    if (!response.ok) {
      throw new Error(`HTTP error! status: ${response.status}`);
    }

    const result = await response.json();
    return result;
  } catch (error) {
    console.error('API request failed:', error);
    return null;
  }
}

// Example usage
const patient = {
  age: 65,
  sex: 1,
  cp: 3,
  // ... other parameters
};

predictHeartDiseaseRisk(patient)
  .then(result => {
    if (result) {
      console.log(`Risk: ${result.risk_percentage.toFixed(1)}%`);
      console.log(`Category: ${result.risk_category}`);
    }
  });
```

cURL Examples

bash

```
# Health check
curl -X GET "https://heartdisease.duminduthushan.com/health"

# Single prediction
curl -X POST "https://heartdisease.duminduthushan.com/predict" \
  -H "Content-Type: application/json" \
  -d '{
   "age": 65,
   "sex": 1,
   "cp": 3,
   "trestbps": 160,
   "chol": 280,
   "fbs": 1,
   "restecg": 0,
   "thalach": 140,
   "exang": 1,
   "oldpeak": 2.5,
   "slope": 2,
   "ca": 1,
   "thal": 2
  }'

# Model information
curl -X GET "https://heartdisease.duminduthushan.com/model/info"
```

Performance Specifications

- **Response Time**: <2 seconds for single predictions
- **Batch Processing**: <1 second per patient
- **Availability**: 99.9% uptime SLA
- **Throughput**: 100+ concurrent requests supported

Security Considerations

- All communications encrypted with TLS 1.3
- Input validation prevents malicious payloads
- Rate limiting prevents abuse
- No patient data stored permanently
- CORS configured for web applications

Support and Feedback

For technical support, integration assistance, or to report issues:

- API Documentation: https://heartdisease.duminduthushan.com/docs
- Interactive testing available in Swagger UI
- Monitor service status via health endpoint