

# **COPD Risk Prediction Challenge ML Project Report**

## **Team Members**

**Kothamasu Naga Venkata Aditya — IMT2023033**

**Abhijit Dibbidi — IMT2023054**

**Adwait Priyadarshan — IMT2023502**

## **GitHub Link**

<https://github.com/erakin027/Chronic-Obstructive-Pulmonary-Disease-COPD-Risk/>

## Contents

<b>1 Task</b>	<b>4</b>
<b>2 Dataset Description</b>	<b>4</b>
2.1 Overview . . . . .	4
2.2 Variables . . . . .	4
<b>3 Pre-processing &amp; EDA</b>	<b>4</b>
3.1 Initial Data Exploration . . . . .	4
3.2 Handling Missing Values and Duplicates . . . . .	4
3.3 Outlier Detection and Treatment . . . . .	5
<b>4 Data Preprocessing</b>	<b>6</b>
4.1 Encoding Categorical Variables . . . . .	6
4.2 Dropping Useless Features . . . . .	6
4.3 Feature Correlation Analysis . . . . .	6
4.4 Standardization . . . . .	6
4.5 Dataset Splitting . . . . .	7
<b>5 Exploratory Data Analysis</b>	<b>7</b>
5.1 Correlation Heatmap . . . . .	7
5.2 Feature Distribution Analysis . . . . .	7
<b>6 Models Tested</b>	<b>7</b>
6.1 Evaluation Framework . . . . .	8
6.2 Clustering Models . . . . .	8
6.2.1 K-Means Clustering . . . . .	8
6.2.2 Gaussian Mixture Model (GMM) . . . . .	8
6.3 Classification Models . . . . .	8
6.3.1 Logistic Regression . . . . .	8
6.3.2 Support Vector Machine (SVM) . . . . .	9
6.3.3 Neural Network - Multilayer Perceptron (MLP) . . . . .	9
6.3.4 PyTorch Neural Network . . . . .	10
6.3.5 Ensemble - Bagging Neural Networks . . . . .	11
<b>7 Model Comparison and Results</b>	<b>11</b>
7.1 Validation Set Performance Rankings . . . . .	11
<b>8 Dataset Size Impact Analysis: 20% vs 100% Data</b>	<b>12</b>
8.1 20% Dataset Creation Methodology . . . . .	12
8.2 Model Performance Comparison: Full vs Reduced Dataset . . . . .	12
8.3 Key Findings from Dataset Size Experiments . . . . .	13
8.3.1 SVM Robustness to Data Scarcity . . . . .	13
8.3.2 Neural Network Sensitivity to Data Volume . . . . .	13
8.4 Extended Experiment: 5% Dataset . . . . .	13
8.5 Threshold Calibration Stability . . . . .	14
8.6 Conclusion from Dataset Size Analysis . . . . .	14
<b>9 Key Observations</b>	<b>14</b>

<b>10 Conclusion</b>	<b>14</b>
10.1 Key Findings Summary . . . . .	14
10.2 Final Statement . . . . .	15

## 1 Task

The primary objective of this project is to develop and compare multiple machine learning models to accurately predict COPD (Chronic Obstructive Pulmonary Disease) risk based on clinical, physiological, and lifestyle measurements. This represents a binary classification problem where the goal is to build predictive models using a training dataset containing patient health metrics with their corresponding COPD risk labels (0 = No risk, 1 = Risk present).

The trained models are then applied to a testing dataset containing input features without labels. The ultimate aim is to identify, tune, and deliver the best-performing classification model for predicting COPD risk with high F1 score, as this metric appropriately balances precision and recall for medical diagnosis applications.

## 2 Dataset Description

### 2.1 Overview

The COPD Risk Prediction Dataset is designed to examine the factors influencing COPD risk determination. It contains clinical measurements from patients, with each record including physiological measurements, laboratory test results, and lifestyle indicators, along with a ground-truth COPD risk classification.

The training dataset `train.csv` consists of 44,553 rows and 27 columns, with the testing dataset `test.csv` consisting of 11,139 rows and 26 columns (excluding the target variable).

### 2.2 Variables

## 3 Pre-processing & EDA

### 3.1 Initial Data Exploration

The dataset was analyzed by examining the first few rows, checking data types of each column, and identifying any missing values. A statistical summary of numerical columns was generated to assess central tendency, variability, and potential outliers.

#### Data Types Detected:

- Integer features: `patient_id`, `age_group`, `height_cm`, `weight_kg`, `dental_cavity_status`, `has_copd_risk`
- Float features: All physiological measurements (waist circumference, vision, hearing, blood pressure, cholesterol levels, enzyme levels, etc.)
- Categorical features (initially): `sex`, `oral_health_status`, `tartar_presence`

### 3.2 Handling Missing Values and Duplicates

A thorough examination of the dataset was conducted to identify missing values and duplicate records. The analysis revealed:

- **Missing Values:** 0 NaN values detected across all columns
- **Duplicates:** 0 duplicate records found
- **Data Quality:** The dataset is complete and requires no data imputation or removal

Table 1: Dataset Features and Description

Feature	Description
patient_id	Unique identifier for each patient
sex	Biological sex (M/F)
age_group	Age categorized in 5-year bands
height_cm	Height in centimeters (int)
weight_kg	Weight in kilograms (int)
waist_circumference_cm	Waist circumference (float)
vision_left	Visual acuity for left eye (float)
vision_right	Visual acuity for right eye (float)
hearing_left	Hearing test result for left ear (float)
hearing_right	Hearing test result for right ear (float)
bp_systolic	Systolic blood pressure in mmHg (float)
bp_diastolic	Diastolic blood pressure in mmHg (float)
fasting_glucose	Blood glucose after fasting in mg/dL (float)
total_cholesterol	Total cholesterol level in mg/dL (float)
triglycerides	Triglyceride level in mg/dL (float)
hdl_cholesterol	HDL cholesterol in mg/dL (float)
ldl_cholesterol	LDL cholesterol in mg/dL (float)
hemoglobin_level	Hemoglobin concentration in g/dL (float)
urine_protein_level	Urine protein presence level (float)
serum_creatinine	Creatinine in blood in mg/dL (float)
ast_enzyme_level	AST enzyme level in U/L (float)
alt_enzyme_level	ALT enzyme level in U/L (float)
ggt_enzyme_level	GGT enzyme level in U/L (float)
oral_health_status	Oral hygiene rating (categorical)
dental_cavity_status	Dental cavities presence (int)
tartar_presence	Tartar buildup indicator (categorical)
<b>Target Variable</b>	
has_copd_risk	COPD risk: 1 = Risk present, 0 = No risk

### 3.3 Outlier Detection and Treatment

Outlier detection was performed using the IQR (Interquartile Range) method on all numerical features. A custom `outlierremoval` class was implemented that:

1. Calculates Q1 (25th percentile) and Q3 (75th percentile) for each feature
2. Computes  $IQR = Q3 - Q1$
3. Defines whiskers: Lower =  $Q1 - 1.5 \times IQR$ , Upper =  $Q3 + 1.5 \times IQR$
4. Caps outliers at whisker boundaries rather than removing them

This approach was applied to preserve data points while reducing the impact of extreme values on model training. Outlier treatment improved model performance, especially for SVM and Logistic Regression.

## 4 Data Preprocessing

### 4.1 Encoding Categorical Variables

Categorical variables were encoded using mapping to convert them into numerical format suitable for machine learning algorithms:

Table 2: Categorical Variable Encoding

Feature	Original Value	Encoded Value
sex	F	1
sex	M	0
oral_health_status	Y	1
oral_health_status	N	0
tartar_presence	Y	1
tartar_presence	N	0

### 4.2 Dropping Useless Features

Two features were identified as non-informative and removed:

- **patient\_id:** Serves only as an identifier with correlation of 0.0129 to target
- **oral\_health\_status:** Found to be constant across all training samples (all values were identical), providing zero variance and no predictive power

### 4.3 Feature Correlation Analysis

Correlation analysis was performed to understand relationships between features and the target variable (has\_copd\_risk):

Table 3: Top Features by Correlation with COPD Risk

Feature	Correlation
hemoglobin_level	0.4023
height_cm	0.3942
weight_kg	0.3015
triglycerides	0.2534
ggt_enzyme_level	0.2398
serum_creatinine	0.2256
waist_circumference_cm	0.2247
sex	-0.5106
hdl_cholesterol	-0.1775
age_group	-0.1633

Note: Sex shows strong negative correlation, indicating gender differences in COPD risk patterns.

### 4.4 Standardization

Numerical features were standardized using **StandardScaler** after the train-test split to avoid data leakage. Standardization ensures all features have similar scales with mean 0 and standard deviation 1, which is crucial for:

- Distance-based algorithms (SVM, K-Means, GMM)
- Neural networks and gradient-based optimization
- Ensuring fair feature importance across different scales

**Important Note:** Standardization is performed after train-test split to prevent data leakage. The scaler is fit on the training set and then applied to validation and test sets.

## 4.5 Dataset Splitting

The dataset was divided into training and validation sets using stratified train-test split with an 80/20 ratio:

- Training set: 35,642 samples (80%)
- Validation set: 8,911 samples (20%)

Stratification ensures that each split maintains the same class distribution as the original dataset, which is critical for binary classification with potentially imbalanced classes.

# 5 Exploratory Data Analysis

## 5.1 Correlation Heatmap

A comprehensive correlation heatmap was generated to visualize relationships between all features and identify potential multicollinearity issues. Key findings:

- Strong positive correlations: height, weight, and hemoglobin with COPD risk
- Strong negative correlation: sex with COPD risk
- Moderate correlations: Various metabolic markers (triglycerides, enzyme levels)
- No severe multicollinearity detected between predictors

## 5.2 Feature Distribution Analysis

Distribution analysis through histograms and boxplots revealed:

- Most features follow approximately normal distributions
- Some features (enzyme levels, triglycerides) show right-skewed distributions
- Outliers present in multiple features, addressed through capping
- Blood pressure and cholesterol levels show expected physiological ranges

# 6 Models Tested

Multiple machine learning models were trained and evaluated to identify the optimal approach for predicting COPD risk. An evaluation framework was developed to consistently measure model performance using F1 score with optimal threshold selection.

## 6.1 Evaluation Framework

A custom `evaluate_model` function was implemented to:

1. Generate probability predictions (when available)
2. Search for optimal classification threshold (0.0 to 1.0 in 0.01 steps)
3. Select threshold that maximizes F1 score on validation data
4. Return Accuracy Score, F1 score, Macro F1 score

For clustering models, a `clusters_to_labels` function maps cluster assignments to class labels using majority voting.

## 6.2 Clustering Models

### 6.2.1 K-Means Clustering

K-Means clustering was applied with 2 clusters to match the binary classification:

- Algorithm: K-Means with  $k=2$
- Initialization: `k-means++`
- Distance metric: Euclidean
- Label mapping: Majority voting per cluster

**Validation F1 Score: 0.6724**

### 6.2.2 Gaussian Mixture Model (GMM)

GMM with 2 components was employed for probabilistic cluster assignment:

- Components: 2
- Covariance type: full
- Optimal threshold: 0.01

**Validation F1 Score: 0.6966**

GMM showed marginally better performance than K-Means, suggesting some benefit from probabilistic modeling.

## 6.3 Classification Models

### 6.3.1 Logistic Regression

Logistic regression was trained with increased iterations:

- `max_iter`: 300
- Solver: default (lbfgs)
- Regularization: L2 (default)
- Optimal threshold: 0.37

**Validation F1 Score: 0.7090**



### 6.3.2 Support Vector Machine (SVM)

#### SVM without probability calibration:

- Kernel: RBF
- probability: False
- Decision function used for predictions

**Validation F1 Score: 0.6976**

**Calibrated SVM:** To enable probability predictions and threshold optimization, `CalibratedClassifierCV` was applied:

- Base estimator: SVC with RBF kernel
- Calibration method: Sigmoid (Platt scaling)
- Cross-validation: 3-fold
- Optimal threshold: 0.26

**Validation F1 Score: 0.7168**

The calibration improved F1 score by approximately 2%, demonstrating the value of probability calibration for threshold optimization.

**Linear SVM:** A linear kernel SVM was tested:

- Kernel: Linear
- C: 1.0
- max\_iter: 5000

**Validation F1 Score: 0.6807**

#### GridSearch-optimized Linear SVM:

- Parameter grid:  $C = [0.01, 0.1, 1, 10, 100]$ ,  $\text{loss} = [\text{'squared\_hinge'}]$
- Cross-validation: 3-fold stratified
- Optimal parameters:  $C = 0.01$ ,  $\text{loss} = \text{'squared\_hinge'}$

**Validation F1 Score: 0.6803**

The linear SVM achieved lower performance than RBF kernel, suggesting non-linear decision boundaries are important for this dataset.

### 6.3.3 Neural Network - Multilayer Perceptron (MLP)

#### MLP Configuration 1 (Single Hidden Layer):

- Architecture: (64,)
- Activation: ReLU
- Solver: Adam
- Alpha (L2): 0.0001
- Learning rate: 0.001
- Max iterations: 300

- Optimal threshold: 0.23

**Validation F1 Score: 0.7102**

**MLP Configuration 2 (Deep Network):**

- Architecture: (128, 64, 32)
- Activation: tanh
- Solver: Adam
- Alpha (L2): 0.001
- Learning rate: 0.001
- Max iterations: 400
- Early stopping: True
- Optimal threshold: 0.33

**Validation F1 Score: 0.7142**

**Hyperparameter-tuned MLP:** RandomizedSearchCV was performed:

- hidden\_layer\_sizes: [(64,32), (128,64), (128,64,32)]
- alpha: [1e-4, 1e-3, 1e-2]
- learning\_rate\_init: [1e-3, 1e-2]
- max\_iter: [300, 400]
- Activation: tanh
- Early stopping: True
- CV folds: 3
- Optimal parameters: hidden\_layer\_sizes=(128,64), alpha=0.001, learning\_rate\_init=0.001, max\_iter=300
- Optimal threshold: 0.37

**Validation F1 Score: 0.7140**

#### 6.3.4 PyTorch Neural Network

A custom PyTorch MLP classifier was implemented with scikit-learn compatibility:

**Architecture:**

- Variable hidden layers with dropout
- Binary cross-entropy loss
- Adam optimizer
- Batch processing with DataLoader

**GridSearchCV Configuration:**

- hidden\_dim: [64]

- hidden\_layers: [1]
- dropout: [0.2]
- lr: [0.001]
- epochs: [15]
- Batch size: 128
- CV folds: 2 (stratified)

**Optimal Configuration:**

- Hidden dim: 64
- Hidden layers: 1
- Dropout: 0.2
- Learning rate: 0.001
- Epochs: 15
- Optimal threshold: 0.31

**Validation F1 Score: 0.7120**

### 6.3.5 Ensemble - Bagging Neural Networks

Bagging was applied to MLPs to reduce variance:

- Base estimator: MLP (64, 32) with 300 iterations
- n\_estimators: 10
- max\_samples: 0.8
- Bootstrap: True
- Optimal threshold: 0.38

**Validation F1 Score: 0.7209**

The bagging ensemble achieved the highest F1 score among all models, demonstrating the benefit of model aggregation.

## 7 Model Comparison and Results

### 7.1 Validation Set Performance Rankings

Table 4: Model Comparison by Validation F1 Score

Rank	Model	F1 Score	Accuracy
1	Bagging Neural Networks	0.7224	0.7537
2	Neural Network (128,64,32) - tanh	0.7138	0.7401
3	Calibrated SVM (RBF)	0.7129	0.7532
4	PyTorch Neural Network	0.7105	0.7347
5	Neural Network (64)	0.7099	0.7348
6	Neural Network (128,64) - Tuned	0.7093	0.7416
7	Logistic Regression	0.7057	0.7275
8	SVM (RBF, uncalibrated)	0.6917	0.7624
9	GMM	0.6891	0.6952
10	K-Means	0.6809	—
11	Linear SVM (Tuned)	0.6770	0.7473

**Note:** Rankings based on F1 score, which balances precision and recall. Accuracy values provided for reference but may be misleading for imbalanced medical datasets.

## 8 Dataset Size Impact Analysis: 20% vs 100% Data

A critical investigation was conducted to understand how model performance varies with dataset size. A reduced dataset containing only 20% of the training data (approximately 8,910 samples) was created through stratified sampling to maintain class distribution. Both SVM and Neural Network models were trained and evaluated on this smaller dataset to compare performance degradation.

### 8.1 20% Dataset Creation Methodology

The reduced dataset was created using a two-stage stratified sampling approach:

1. First stage: 20% of total data extracted using stratified split
2. Second stage: The 20% subset further split into 80/20 train/validation
3. Final sizes:
  - Training: 7,128 samples (80% of 20% = 16% of original)
  - Validation: 1,782 samples (20% of 20% = 4% of original)

This approach ensures representative class distribution at all levels while significantly reducing computational requirements.

### 8.2 Model Performance Comparison: Full vs Reduced Dataset

Table 5: Performance Comparison: 100% vs 20% Dataset

Model	100% Data F1	20% Data F1	Performance Drop
Calibrated SVM (RBF)	0.7129	0.7082	0.0047 (0.66%)
Neural Network (64,)	0.7099	0.6922	0.0177 (2.49%)
<b>Relative Degradation</b>	—	—	<b>SVM: 3.8x better</b>

### 8.3 Key Findings from Dataset Size Experiments

#### 8.3.1 SVM Robustness to Data Scarcity

##### Calibrated SVM (RBF):

- Full dataset:  $F1 = 0.7129$  (threshold = 0.30)
- 20% dataset:  $F1 = 0.7082$  (threshold = 0.29)
- Performance drop: 0.0047 (0.66%)
- Accuracy maintained: 0.7424 on reduced data

The SVM demonstrated remarkable stability with only a **0.66% degradation** in F1 score despite using only 20% of training data. This robustness can be attributed to:

1. **Support vector dependency:** Only support vectors influence the model, making it less sensitive to total dataset size
2. **Kernel efficiency:** RBF kernel effectively captures non-linear patterns even with limited data
3. **Regularization strength:** The C parameter controls complexity, preventing overfitting on smaller datasets

#### 8.3.2 Neural Network Sensitivity to Data Volume

##### Neural Network (64, hidden units):

- Full dataset:  $F1 = 0.7099$  (threshold = 0.32)
- 20% dataset:  $F1 = 0.6922$  (threshold = 0.16)
- Performance drop: 0.0177 (2.49%)
- Accuracy on reduced data: 0.7110

The neural network showed a **2.49% degradation**, which is **3.8 times larger** than SVM's drop. This increased sensitivity reflects:

Notably, the optimal threshold shifted dramatically from 0.32 (full data) to 0.16 (reduced data), indicating calibration instability with limited samples.

### 8.4 Extended Experiment: 5% Dataset

A further experiment with only 5% of original data (not shown in detail) revealed:

- **SVM:** Similar performance drop ( $\sim 0.005$  additional decline)
- **Neural Network:** Catastrophic degradation ( $\sim 0.07$  total drop from baseline)

This extreme case confirms that neural networks require substantially more data to achieve stable performance.

## 8.5 Threshold Calibration Stability

An important observation is the stability of optimal thresholds:

- **SVM:** Threshold shifted minimally ( $0.30 \rightarrow 0.29$ ), indicating stable probability calibration
- **Neural Network:** Threshold shifted drastically ( $0.32 \rightarrow 0.16$ ), suggesting unreliable probability estimates on small data

This threshold instability in neural networks poses risks for medical applications where calibrated probabilities are critical for decision-making.

## 8.6 Conclusion from Dataset Size Analysis

### Winner for Small Datasets: Support Vector Machines

SVMs demonstrate superior robustness and reliability when training data is limited.

However, with sufficient data (40,000+ samples as in this study), neural networks can match or exceed SVM performance while offering greater modeling flexibility.

## 9 Key Observations

1. **Dataset Size Matters:** SVMs demonstrate 3.8x better robustness to data reduction compared to neural networks. With only 20% of data, SVM performance dropped by 0.66% while neural networks degraded by 2.49%.
2. **Ensemble Value:** Bagging neural networks achieved the highest validation F1 (0.7224), confirming that ensemble methods reduce variance and improve robustness.
3. **Threshold Optimization:** Systematic threshold search (0.0-1.0 in 0.01 steps) significantly improved F1 scores across all models, with optimal thresholds ranging from 0.16 to 0.40 depending on model calibration and dataset size.
4. **Calibration Impact:** SVM calibration using Platt scaling improved F1 from 0.6917 to 0.7129, demonstrating the importance of probability calibration for medical prediction tasks.
5. **Clustering vs. Classification:** Supervised learning ( $F1 > 0.70$ ) substantially outperforms unsupervised clustering ( $F1 \approx 0.68$ -0.69), as classification models directly optimize for the prediction target.
6. **Deep Learning Benefits:** PyTorch and scikit-learn neural networks achieved comparable performance on large datasets, validating the architecture design. The flexibility of custom PyTorch implementation allows for future extensions.
7. **Linear vs Non-linear:** RBF kernel SVM significantly outperforms linear SVM (F1: 0.7129 vs 0.6771), confirming that non-linear decision boundaries are essential for this dataset.

## 10 Conclusion

### 10.1 Key Findings Summary

#### 1. Model Performance Hierarchy:

- Neural networks (ensemble and deep) achieve highest F1 scores (0.7224-0.7138)

- Calibrated SVM competitive at 0.7129 F1 with superior robustness
- Traditional ML (Logistic Regression) achieves respectable 0.7057 F1
- Clustering methods (K-Means, GMM) underperform at 0.6809-0.6891 F1

## 2. Dataset Size Impact (Critical Finding):

- **SVM:** Remarkably robust with only 0.66% performance drop on 20% data
- **Neural Networks:** 2.49% degradation on 20% data ( $3.8\times$  worse than SVM)
- **Implication:** For medical datasets  $<10,000$  samples, prioritize SVMs
- **Threshold stability:** SVMs maintain calibration, NNs show instability on small data

## 10.2 Final Statement

This comprehensive study successfully demonstrates that:

- Neural networks excel with sufficient data (44,000+ samples)
- SVMs are remarkably robust for small datasets (data-scarce scenarios)
- Proper preprocessing, threshold optimization, and calibration are crucial