Room Occupancy Count

Big Data Analytics Project

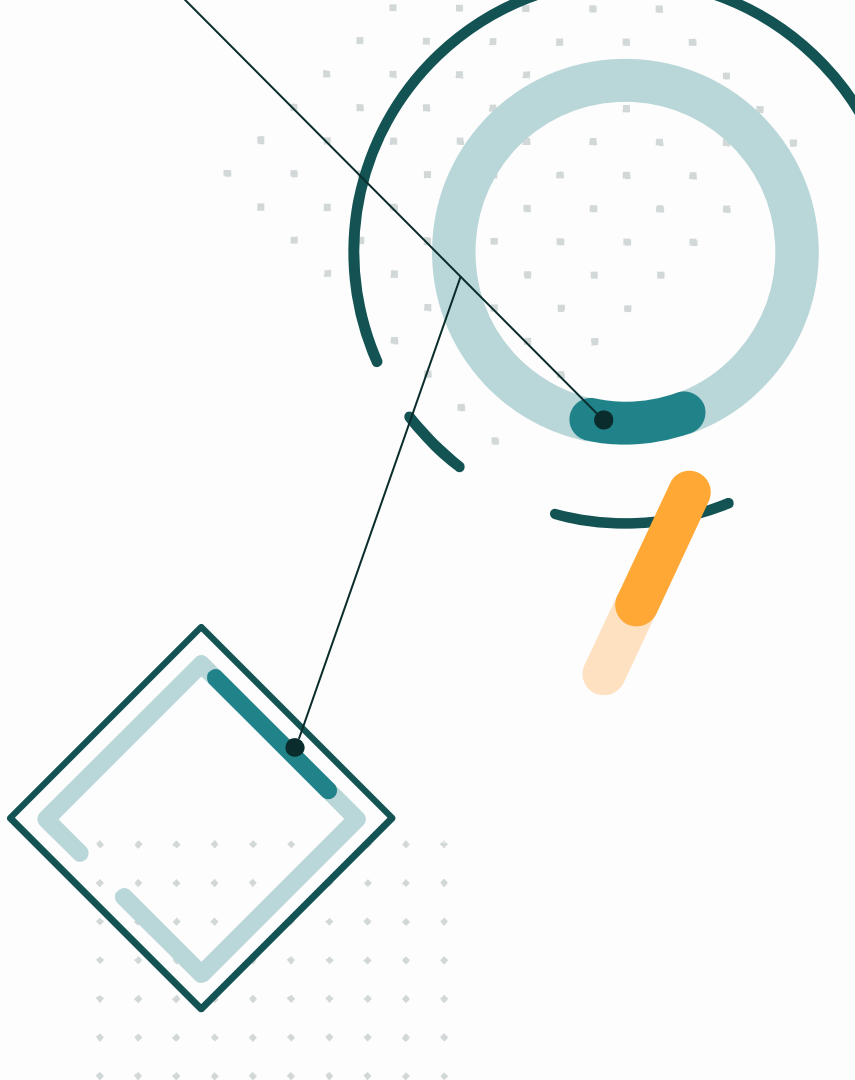
Table of contents

01

Problem and Data

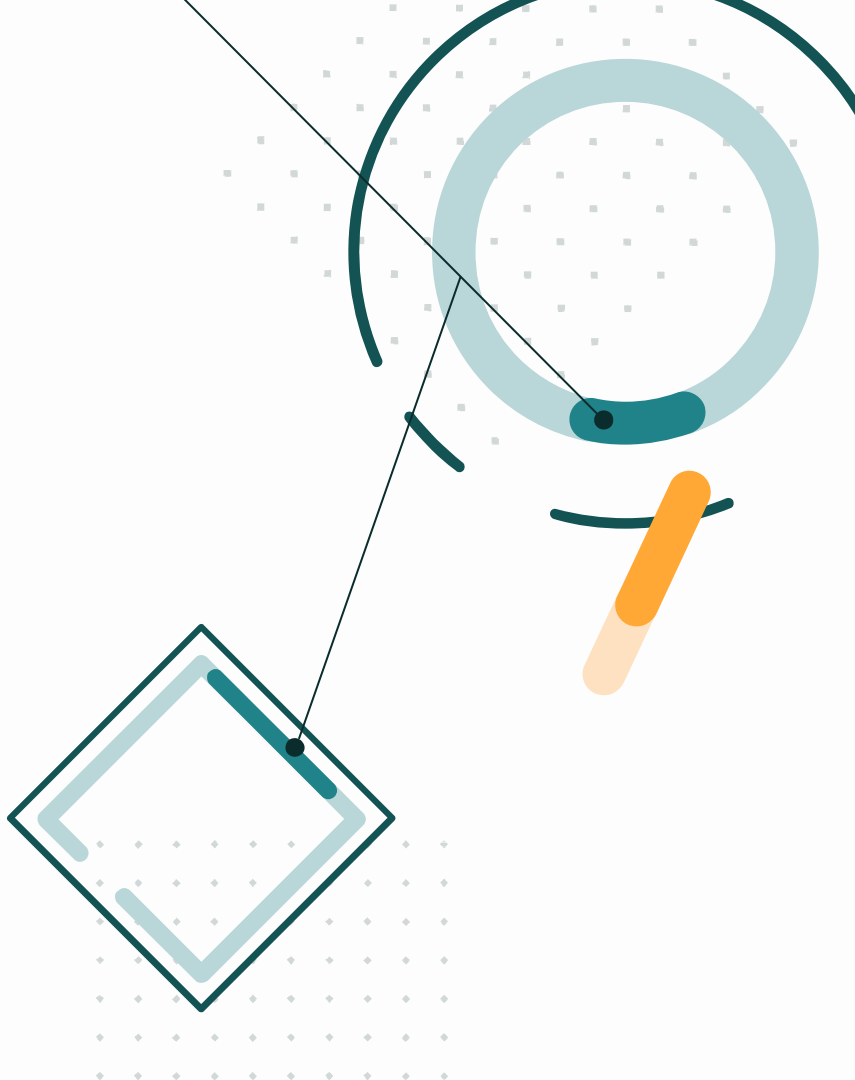
Objective

Predict number of people in a room through sensor data.



Problem Type

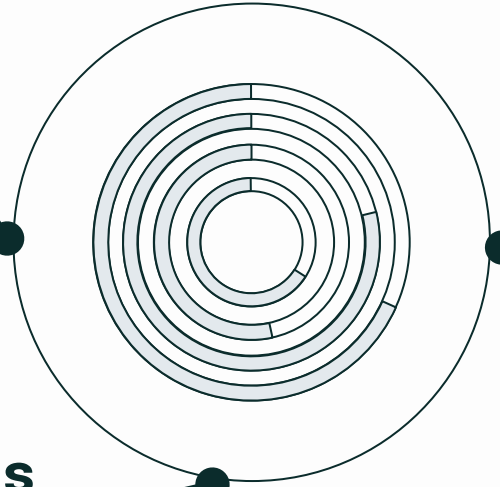
- Multilabel Classification
- Timeseries Data



The Dataset

10,129 Instances

Timeseries data points
covering a 7 days span



18 Features

- 2 Temporal Features
- 16 Sensor Features

4 Categorical Labels

Number of people in the room ranging
from 0 to 3

The Features

Temporal

- Date
- Time

Temperature

4 sensors measuring continuous values in °C

Light

4 Sensors measuring continuous values in Lux

Sound

4 sensors measuring continuous values in Volts

CO2

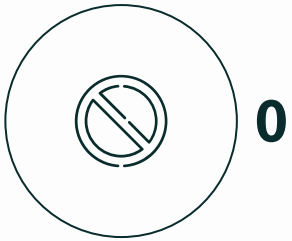
- 1 sensor measuring continuous values in parts per millions
- Slope of CO2 values taken in a sliding window of 25 CO2 points

PIR

2 sensors measuring in binary values whether motion is detected or not in the room

The Label

Count of people in the room at the given time in a categorical range

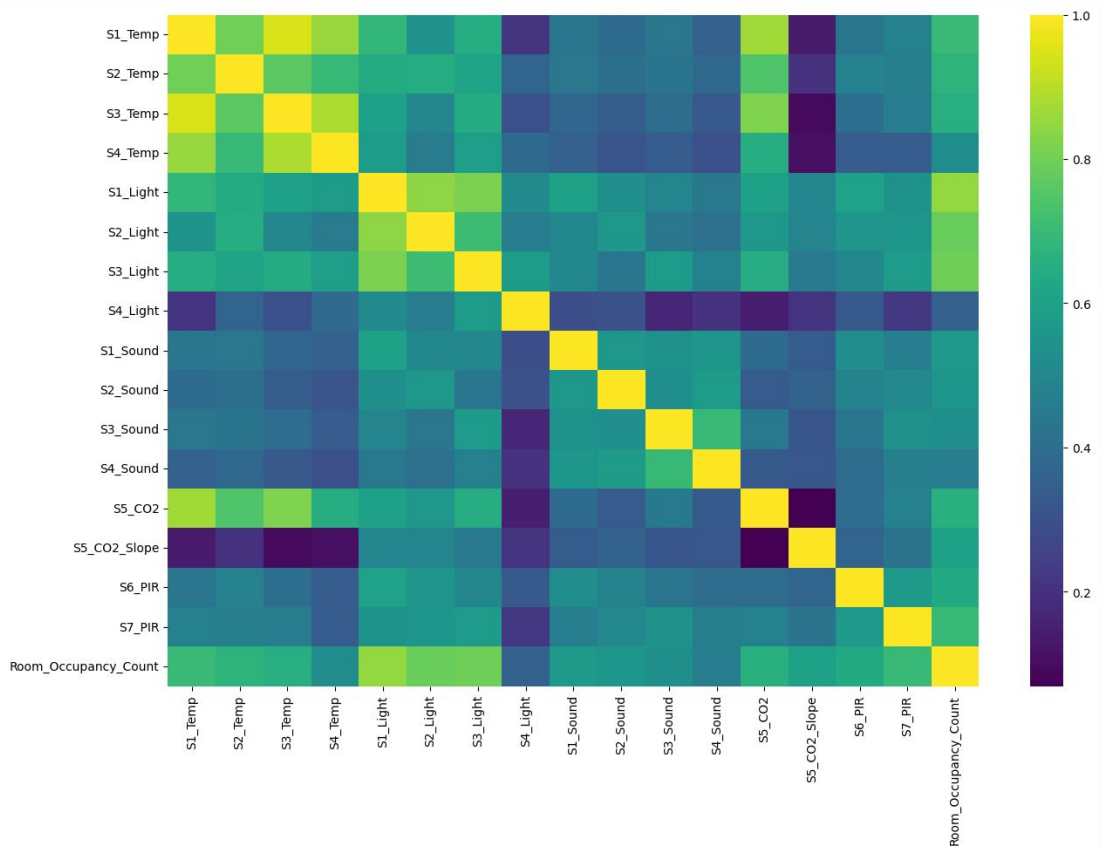

0


1

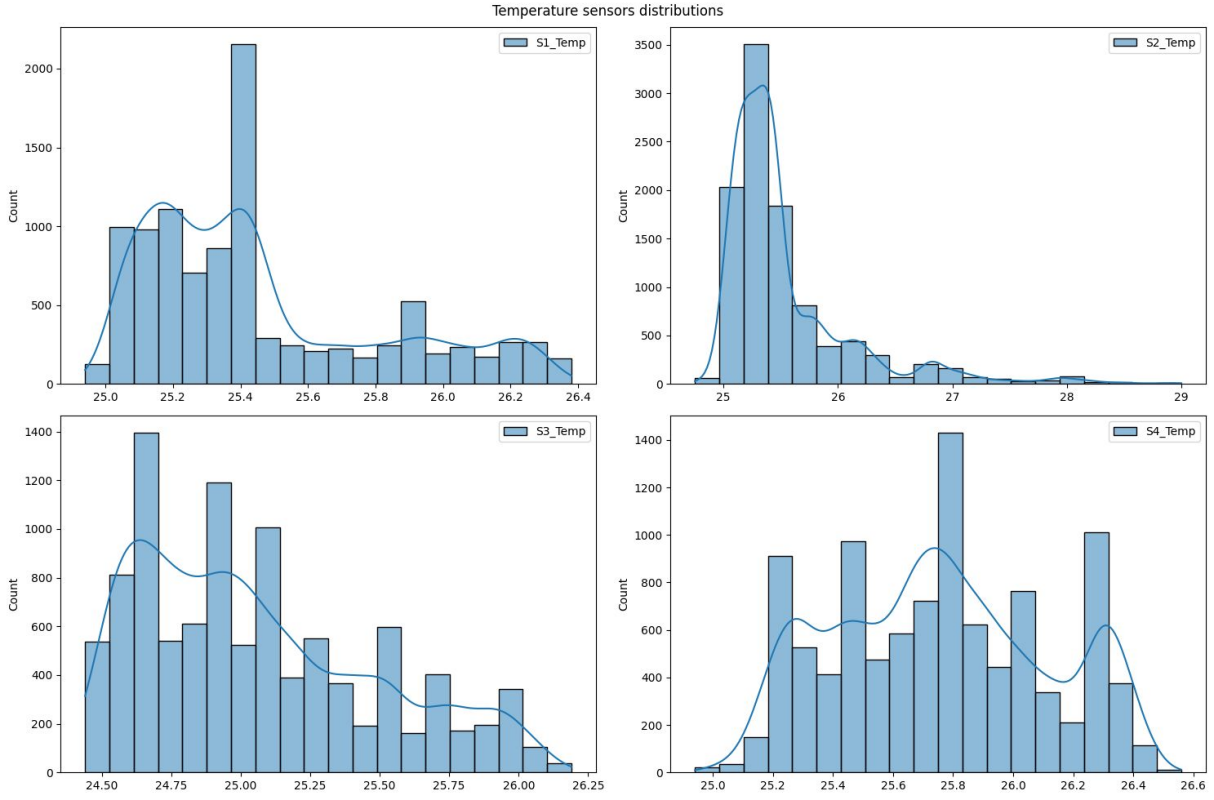

2


3

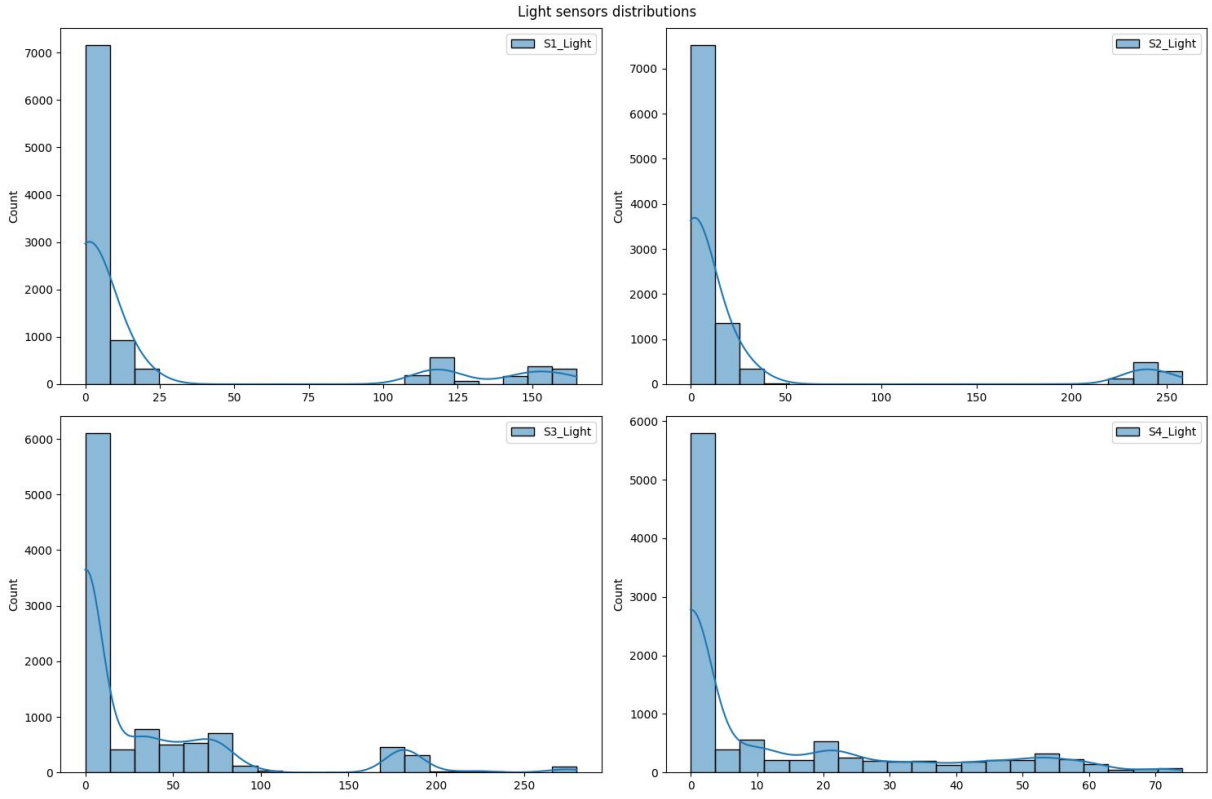
Features Correlation



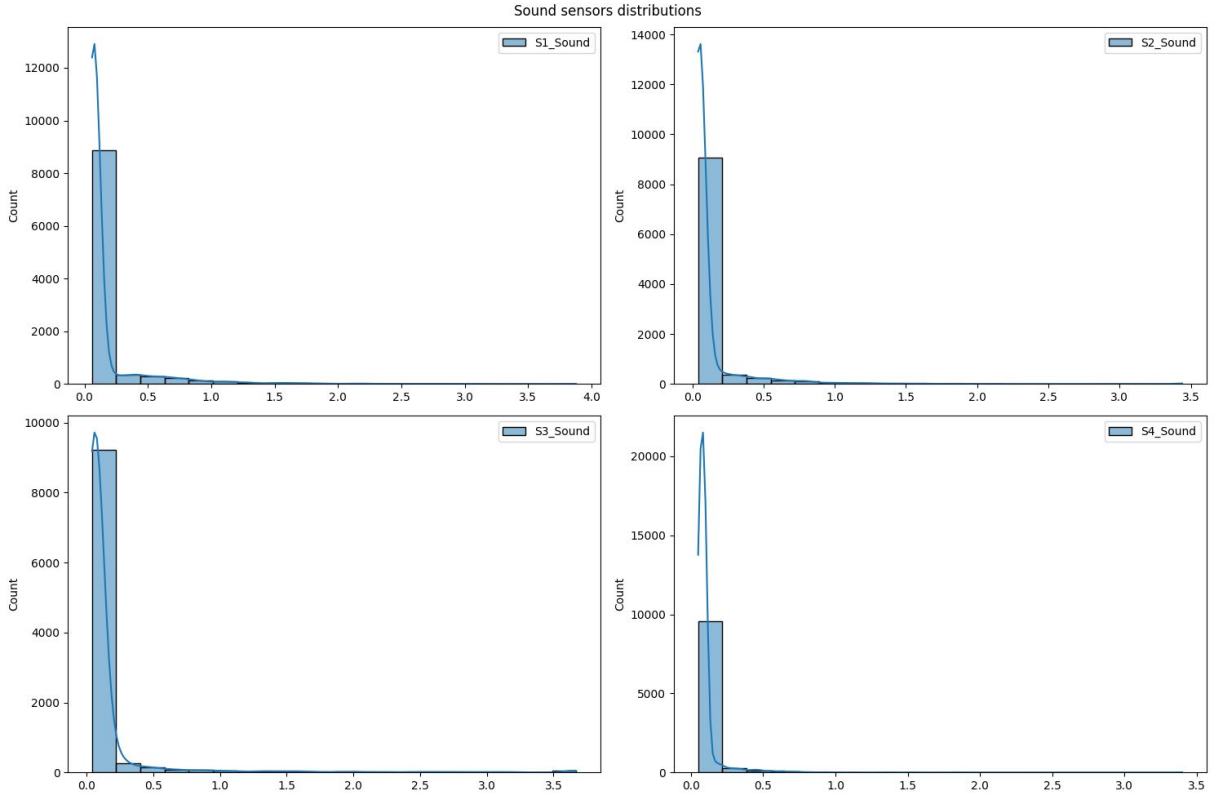
Temperature Sensors Distribution



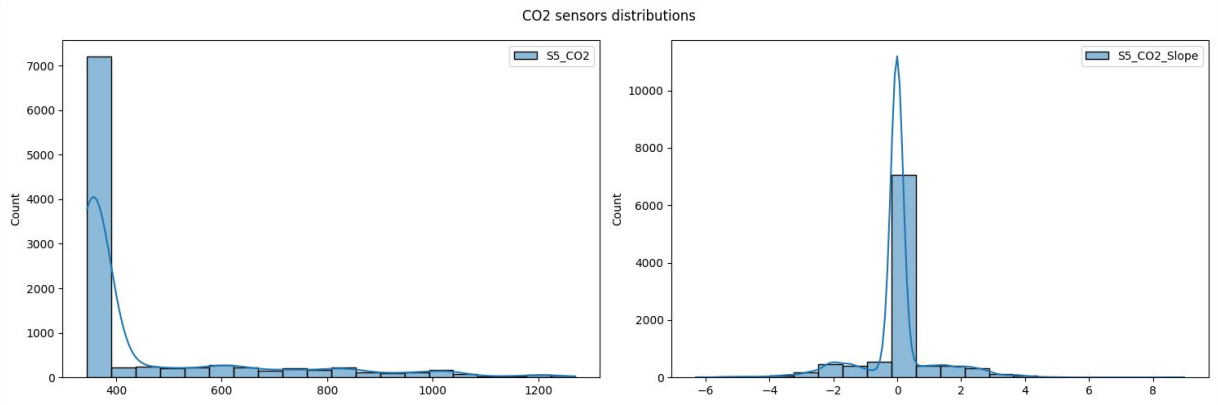
Light Sensors Distribution



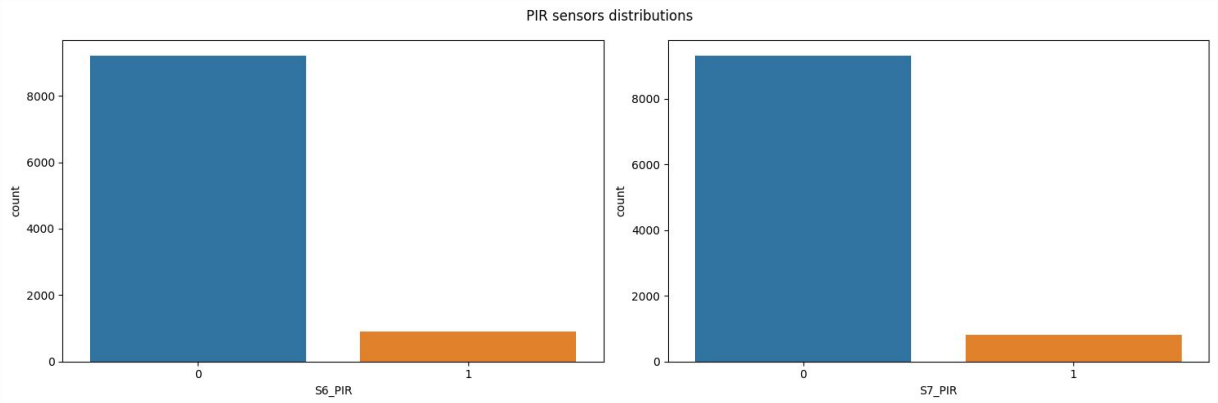
Sound Sensors Distribution



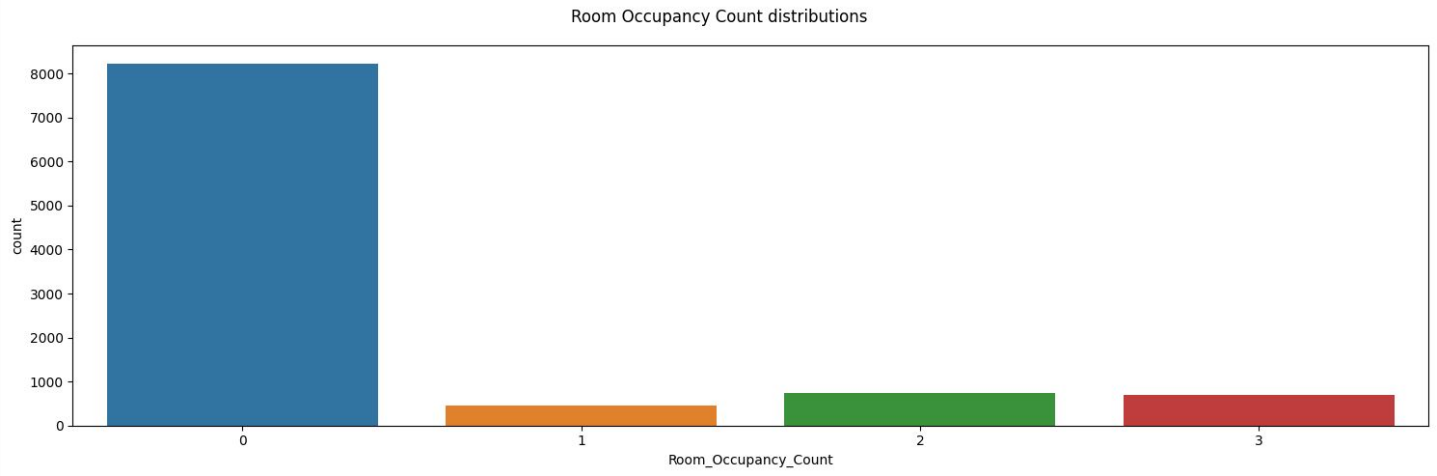
CO2 Sensors Distribution



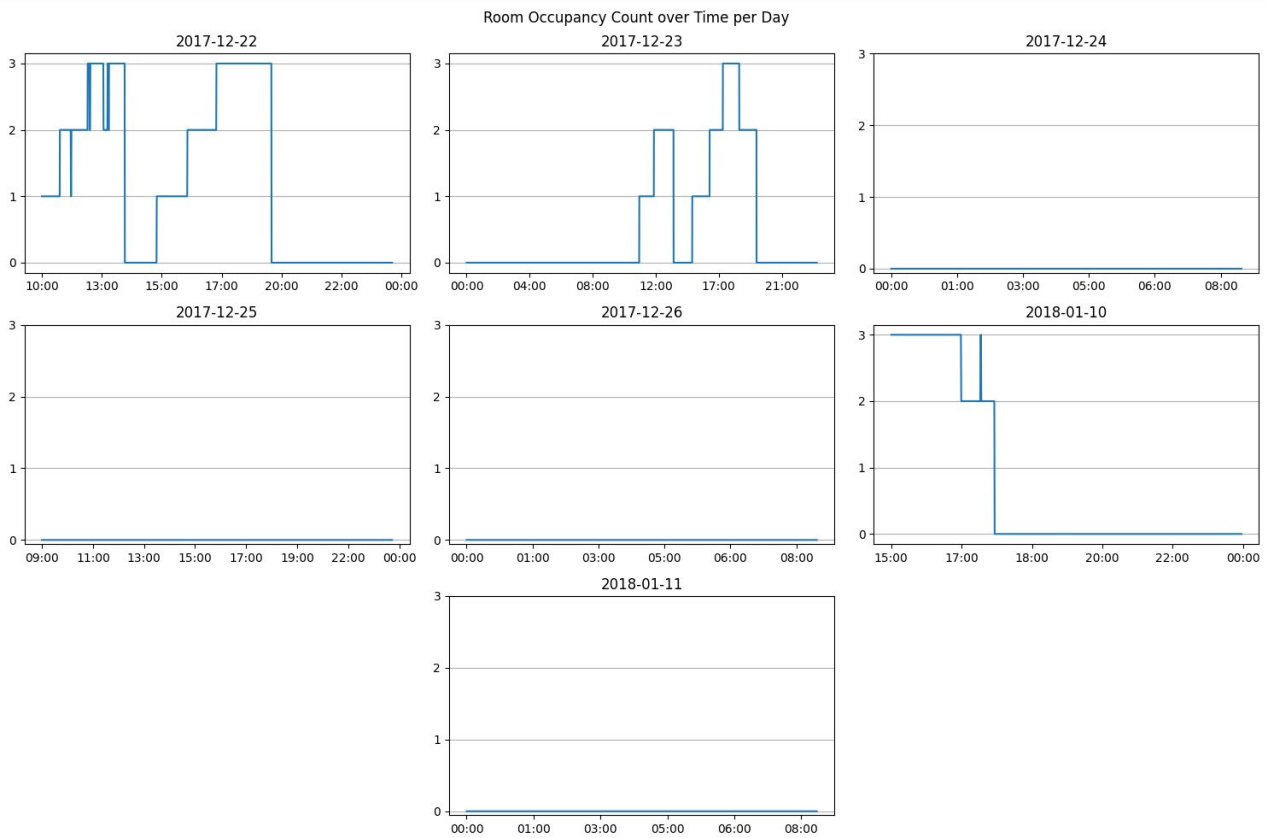
PIR Sensors Distribution



Label Distribution



Label Distribution through Time



02

Architecture

Resources

Experiment tested on both:

- **Local Cluster**
- **Google Colab**

Tested the correct functionality of the local cluster.

Actual experiment and result obtained by colab

- **Faster**
- More access to **memory**

Local Cluster Setup

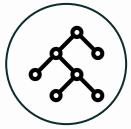
Master Node

4096 MB
IP: 192.168.33.10

Worker Node

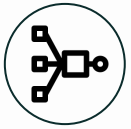
4096 MB
IP: 192.168.33.11

Tested Models



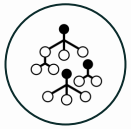
Gradient Boosted Tree

Logistic Regression



Multilayer Perceptron

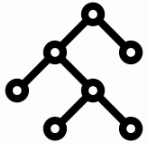
Naive Bayes



Random Forest

Support Vector Machine

Gradient Boosted Tree



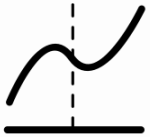
Fixed Parameters

- Seed: 42
- Subset Strategy: *sqrt*
- Impurity: *variance*
- Loss type: *logistic*

Tested Hyperparameters

- Step Size
- Validation Tolerance

Logistic Regression



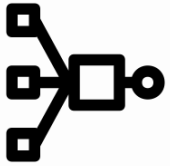
Fixed Parameters

- Family: *multinomial*

Tested Hyperparameters

- Max Iterations
- ElasticNet Parameter:
  - 0 = L2 PENALITY
  - 1 = L1 PENALITY
  - (0, 1) = COMBINATION OF BOTH
- Regularization Parameter

Multilayer Perceptron



Fixed Parameters

- Seed: 42

Tested Hyperparameters

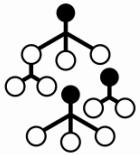
- Solver
- Max Iterations
- Step Size:
- Hidden Layers (Size= (|input features| + |output features|) / 2)
    - 1 HIDDEN LAYER
    - 2 HIDDEN LAYERS

Naive Bayes

Tested Hyperparameters

- Smoothing
- Model Type
  - MULTONOMIAL
  - GAUSSIAN

Random Forest



Fixed Parameters

- Seed: 42
- Subset Strategy: *sqrt*

Tested Hyperparameters

- Impurity
  - GINI
  - ENTROPY
- Number of Trees
- Maximum Depth
- Minimum Info Gain

Support Vector Machine

Tested Hyperparameters

- Regularization Parameter
- Fit Intercept
- Tolerance

Handling Multilabel Classification

Multilabel Classification natively implemented for most architecture in Sparks

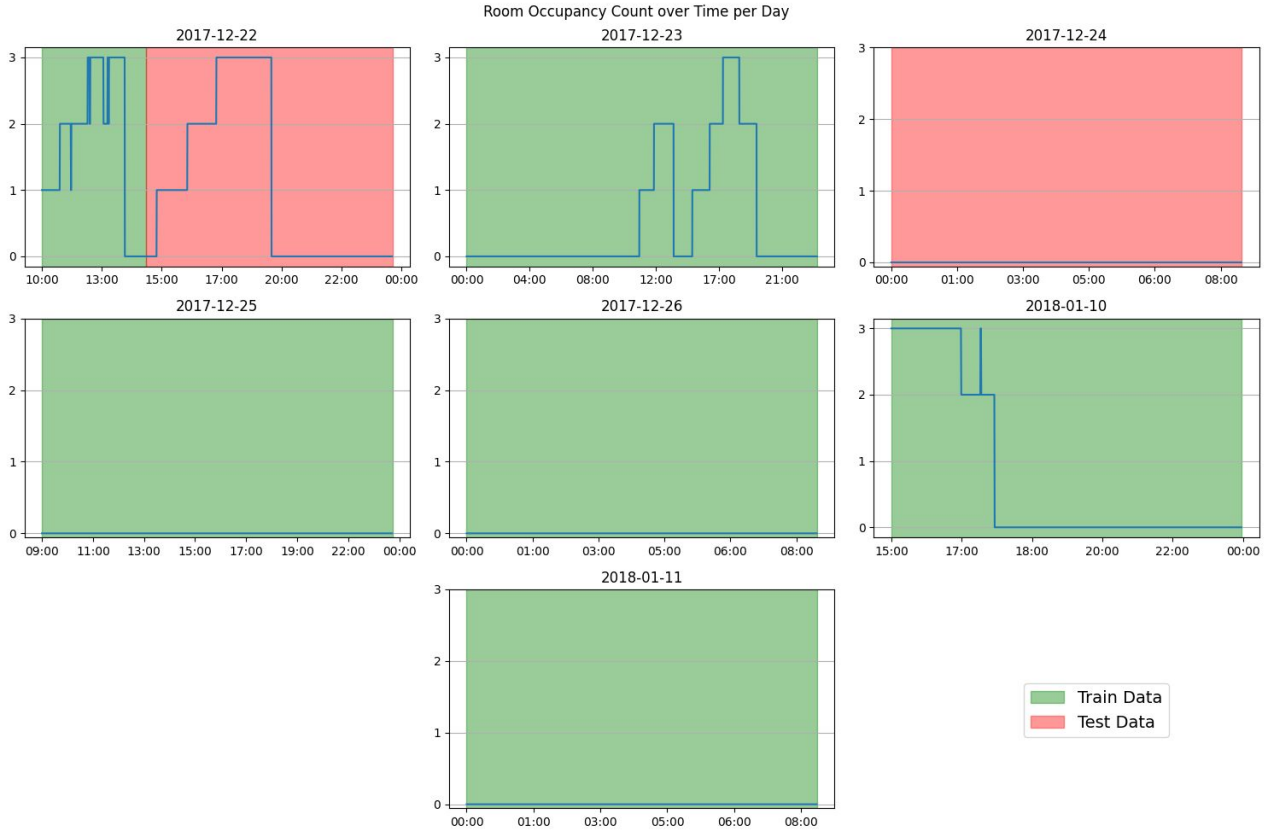
One Vs Rest Binary Classification for each label adopted for:

- **Gradient Boosted Tree**
- **Support Vector Machine**

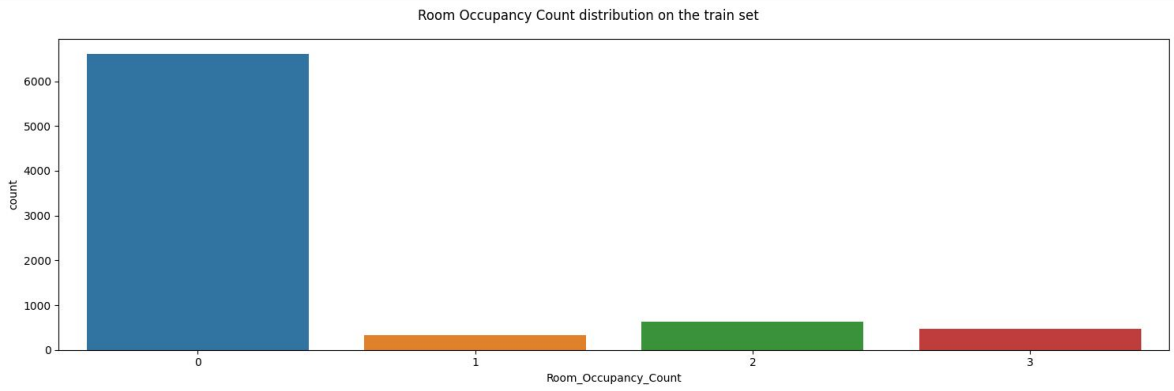
03

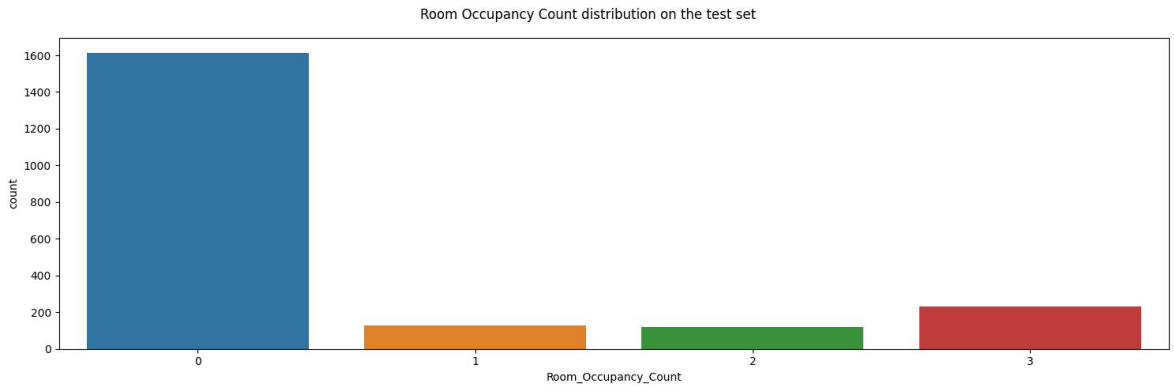
Experimental Setting

Train and Test Split



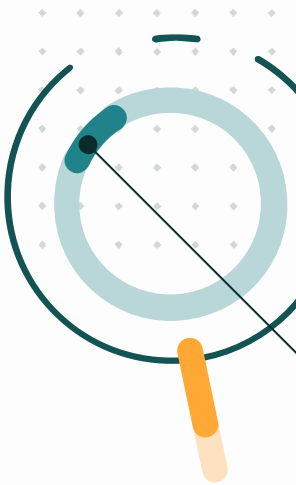
Train and Test Label Distribution





Training Pipeline

Data Processing



Feature Selection

- All **sensor data**
- Time features excluded
- **R-Formula** to vectorize data

Standardization

Standard scaling applied for:

- **Logistic Regression**
- **Multilayer Perceptron**
- **Support Vector Machine**

Feature Shifting

Naive Bayes using **multinomial distribution** accepts **non-negative** values

- **CO2 Slope** values were **shifted by 20**

Grid Search

Grid Search on the selected combination of hyperparameters validating on **F1 score**

Class sampling* with instance weight inversely proportional to their label distribution in order to account for class unbalance and insist on under-represented features

$$\text{weight}(x) = \frac{|\mathcal{D}|}{|\mathcal{C}| \cdot |\mathcal{D}_c|}$$

Where:

- x is an instance
- c is the label of x
- $\mathcal{D}$ is the training dataset
- $\mathcal{C}$ is the set of labels
- $\mathcal{D}_c$ is the subset of the training dataset with class c

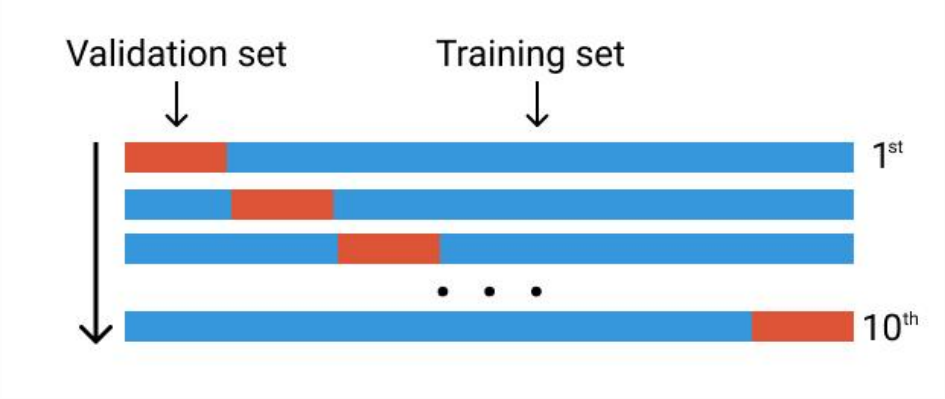
** Not applied for Multilayer Perceptron*

Cross Validation

10-Fold Cross Validation on the training set

- **Stratification** for equal class distribution on each fold
- Selected **contiguous instances** for each label in each fold to preserve time independency

Re-train the model on the whole train set with the best found hyperparameters according to grid search and cross validation scores



Threshold Selection*

Select for **each label** the **threshold** that maximizes its **average F1 score** for each validation fold.

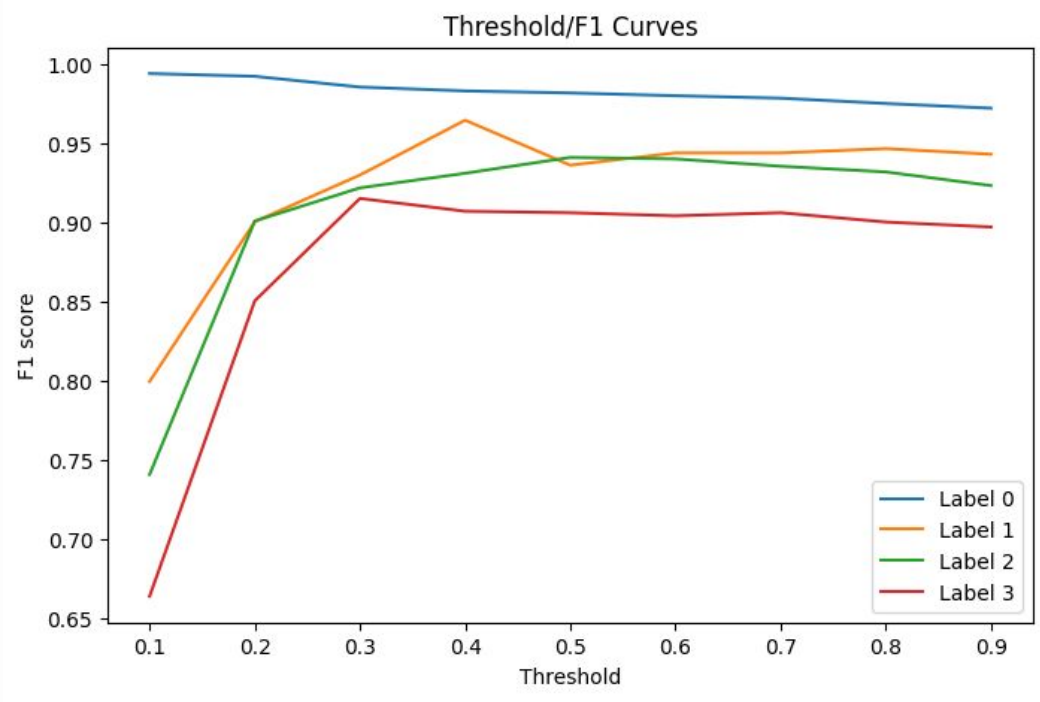
The selected threshold t for a class c changes for each instance x the **probability score** p of it being an instance of class c such that:

$$p_c(x) = \frac{p_c(x)}{t}$$

** Not applied for Gradient Boosted Tree and Support Vector Machine*

Threshold Selection

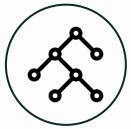
Example of threshold/F1 curves for the Random Forest Classifier



04

Results

Best Hyperparameters

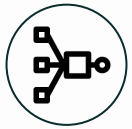


Gradient Boosted Tree

- Step Size: 0.1
- Validation Tolerance: 0.1

Logistic Regression

- Maximum Iterations: 50
- ElasticNet Parameter: 0 (L2 Penality)
- Regulation Parameter: 0.0001

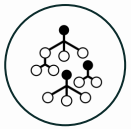


Multilayer Perceptron

- Solver: *l-bfgs*
- Maximum Iterations: 100
- Step Size: 0.3
- Hidden Layers: 1

Naive Bayes

- Smoothing: 1.0
- Model Type: *gaussian*



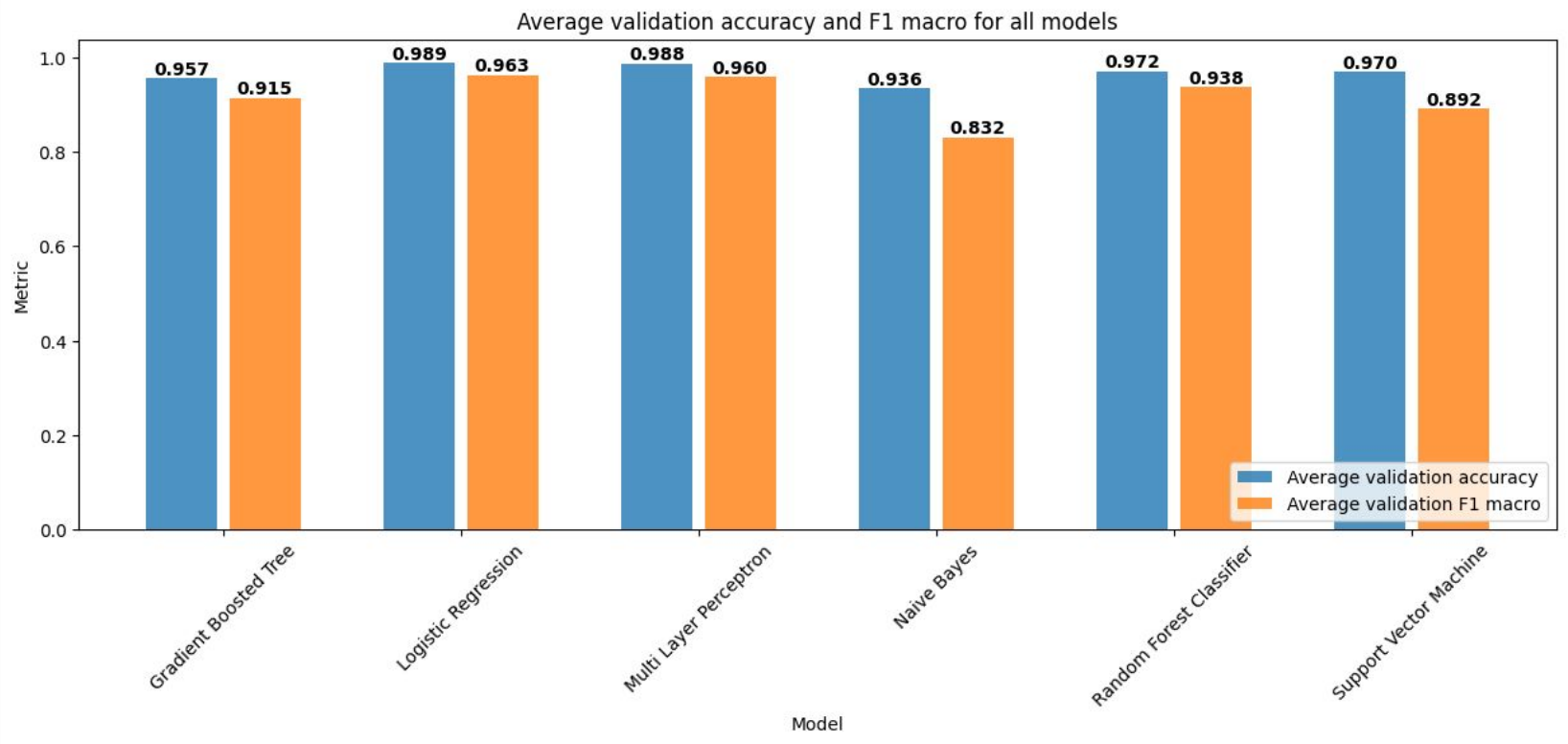
Random Forest

- Impurity: *entropy*
- Number of Trees: 100
- Maximum Depth: 10
- Minimum Info Gain: 0.2

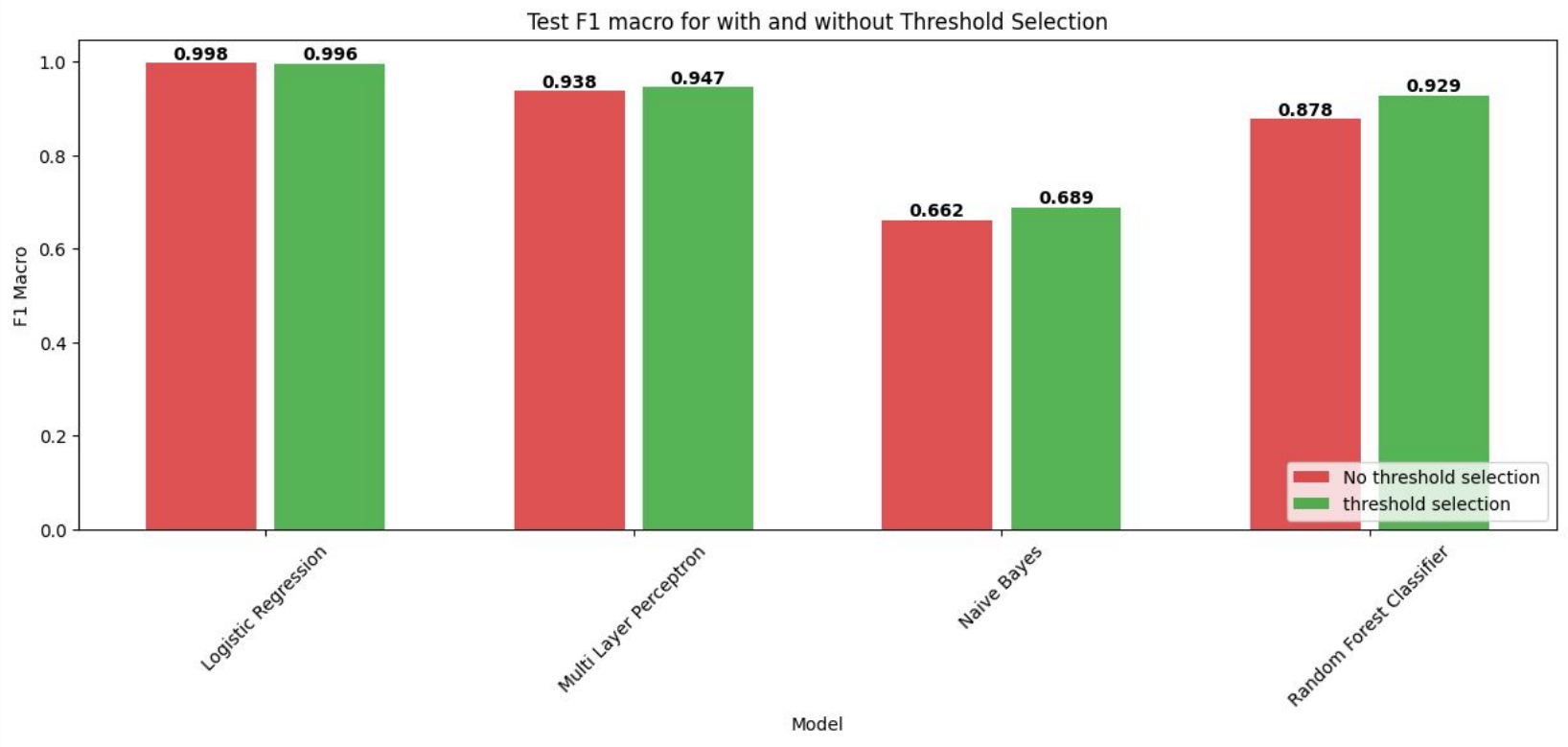
Support Vector Machine

- Regulation Parameter: 0.0001
- Fit Intercept: False
- Tolerance: 0.001

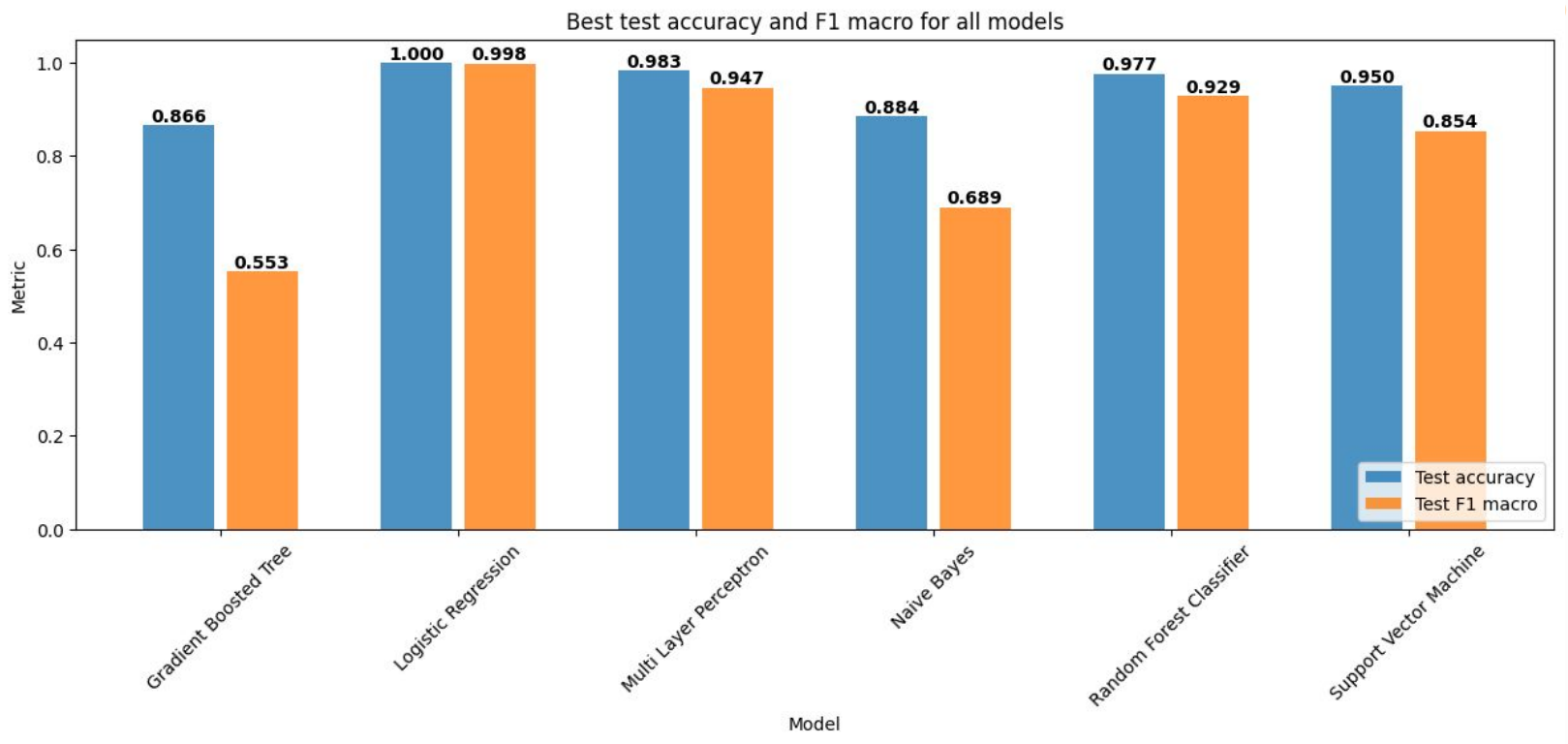
Average Validation Results



Threshold Selection Test Results



Test Results

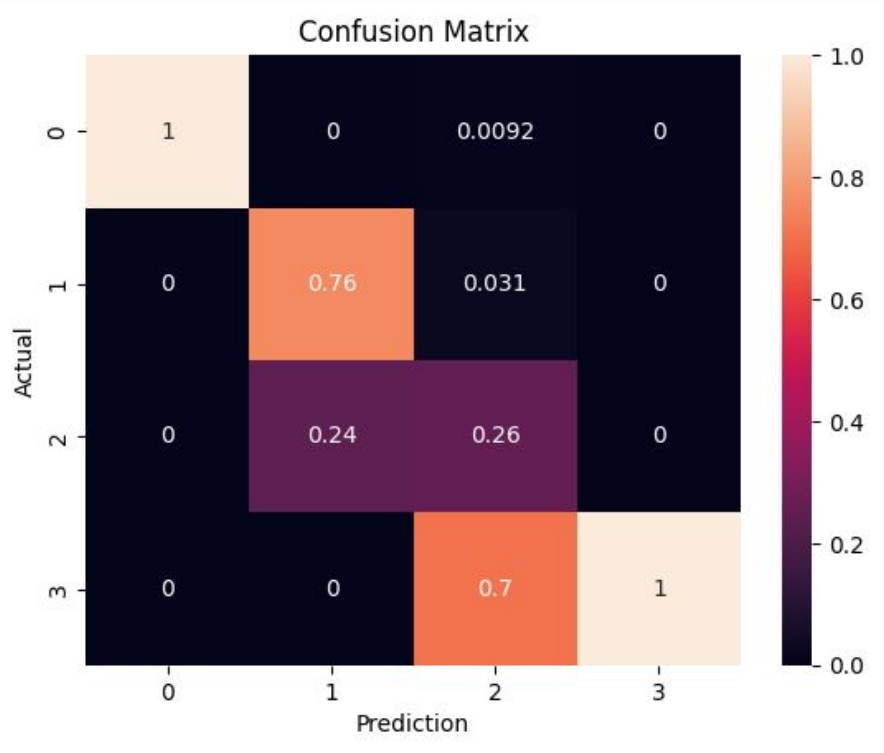


Gradient Boosted Tree

F1 scores:

- **Label 0:** 0.999
- **Label 1:** 0.831
- **Label 2:** 0.372
- **Label 3:** 0.009

F1 Macro: 0.553
Accuracy: 0.866

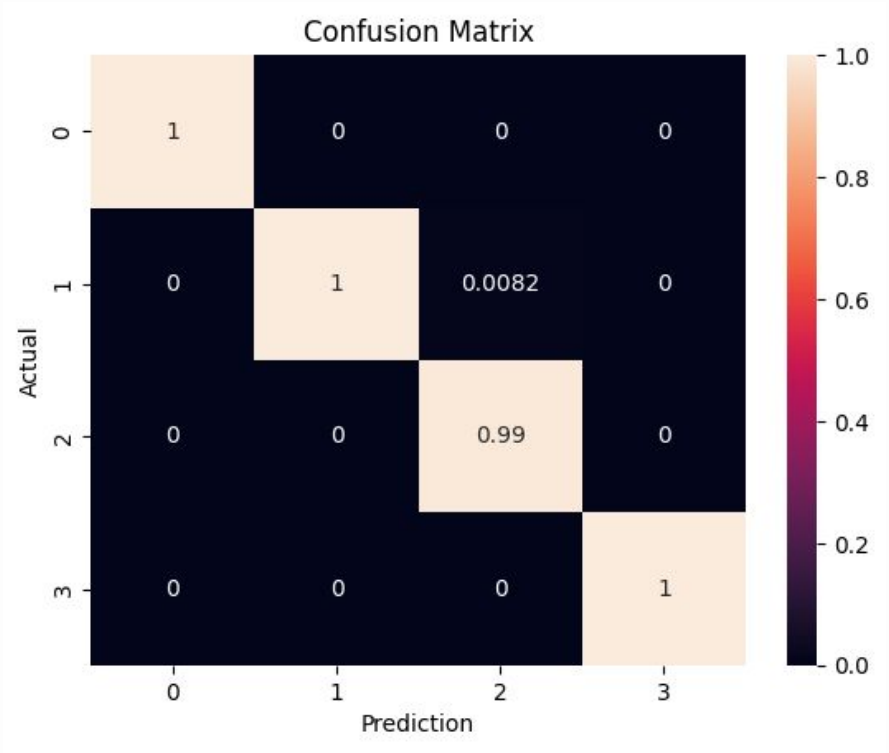


Logistic Regression

F1 scores:

- **Label 0:** 1.000
- **Label 1:** 0.996
- **Label 2:** 0.996
- **Label 3:** 1.000

F1 Macro: 0.998
Accuracy: 1.000

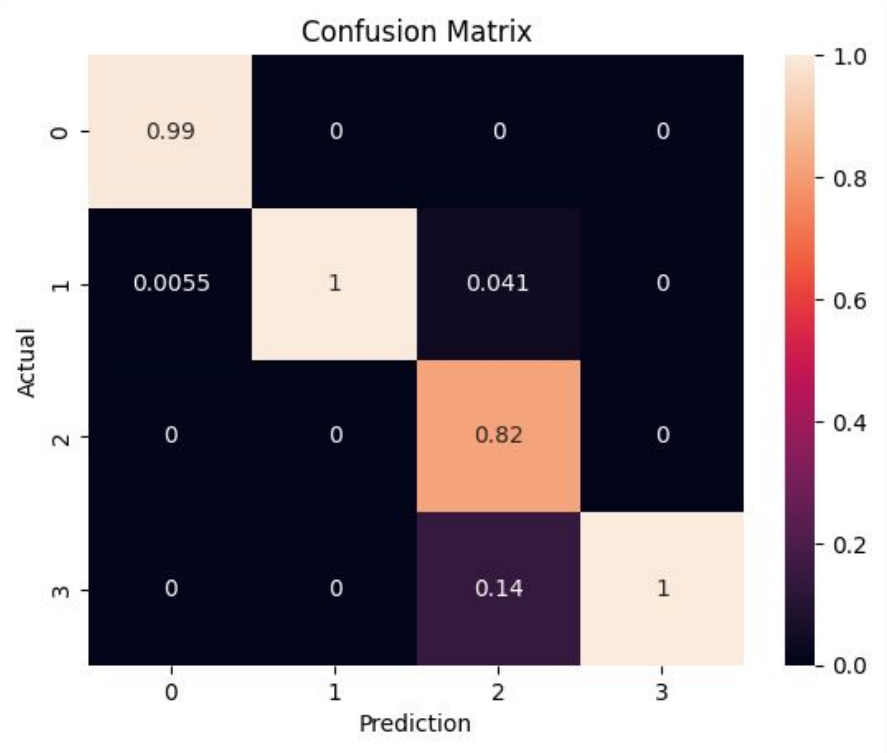


Multilayer Perceptron

F1 scores:

- **Label 0:** 0.997
- **Label 1:** 0.938
- **Label 2:** 0.900
- **Label 3:** 0.952

F1 Macro: 0.947
Accuracy: 0.983

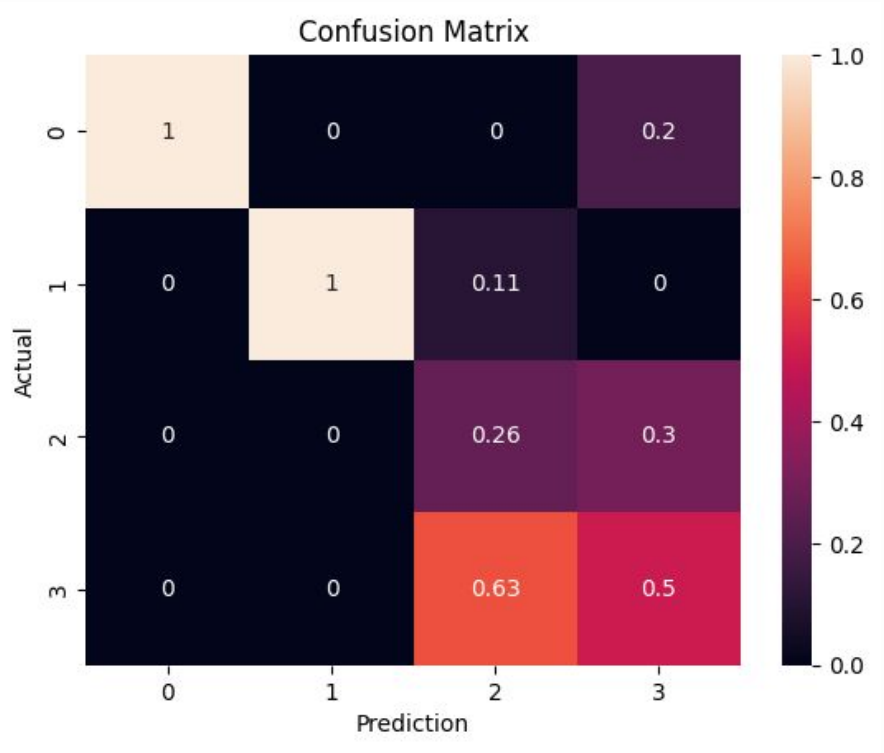


Naive Bayes

F1 scores:

- **Label 0:** 0.982
- **Label 1:** 0.942
- **Label 2:** 0.270
- **Label 3:** 0.563

F1 Macro: 0.689
Accuracy: 0.884

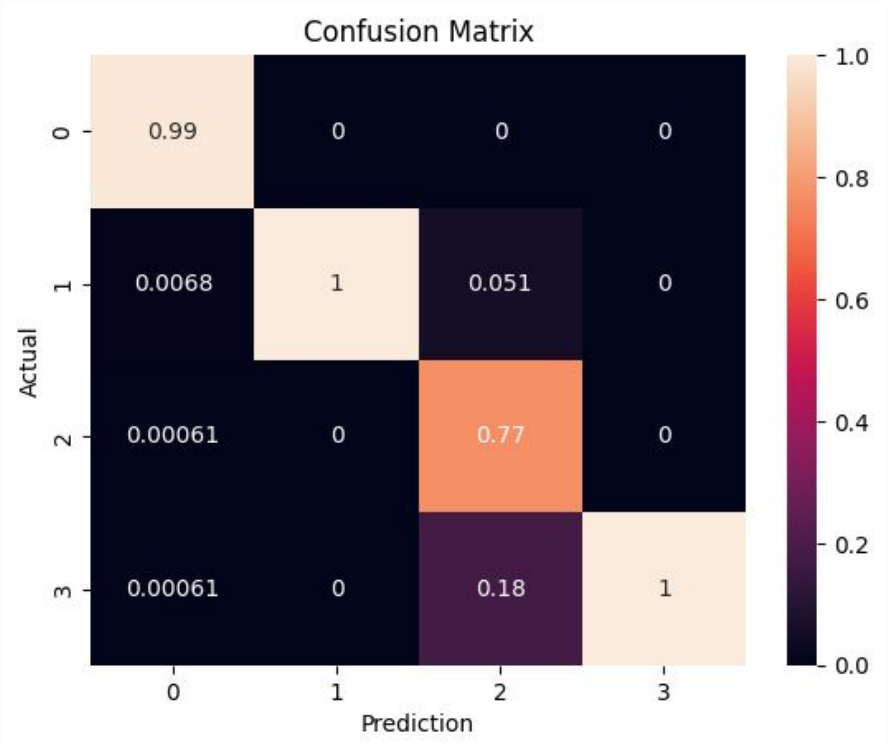


Random Forest

F1 scores:

- **Label 0:** 0.996
- **Label 1:** 0.920
- **Label 2:** 0.866
- **Label 3:** 0.933

F1 Macro: 0.929
Accuracy: 0.977

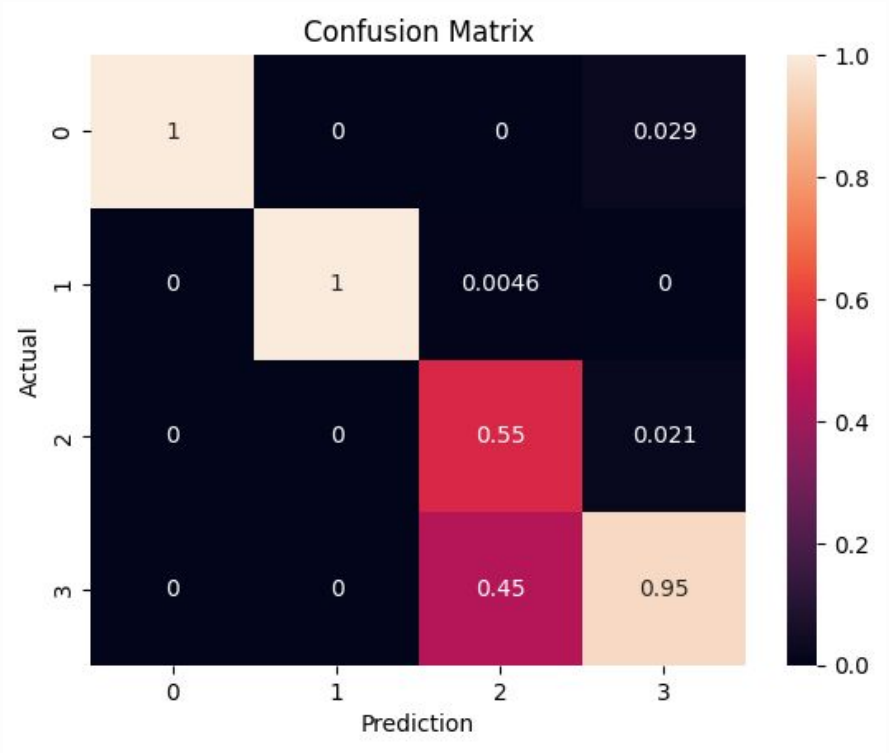


Support Vector Machine

F1 scores:

- **Label 0:** 0.999
- **Label 1:** 0.996
- **Label 2:** 0.700
- **Label 3:** 0.719

F1 Macro: 0.854
Accuracy: 0.950



Thanks!

Do you have any questions?