health_analysis

November 22, 2025

1 Predicting risk factors for maternal mortality

by Apoorva Srivastava, Guaner (Gloria) Yi, Jeffrey Ding & Randall Lee 2025/11/22

[1]:
```
import numpy as np
import pandas as pd
import requests
import zipfile
import altair as alt
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.compose import make_column_transformer
from sklearn.preprocessing import StandardScaler, label_binarize
from sklearn.dummy import DummyClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.model_selection import cross_validate
from sklearn.pipeline import make_pipeline
from scipy.stats import loguniform
from sklearn.model_selection import RandomizedSearchCV, cross_val_predict
from sklearn.metrics import fbeta_score, make_scorer, recall_score,␣
↪ConfusionMatrixDisplay, confusion_matrix, RocCurveDisplay, roc_auc_score
```

2 Summary

In this project, we built a Support Vector Classifier (SVC) that draws from maternal health measurements to predict the risk intensity levels (low, medium, or high) of pregnant women. Data was sourced from the "Maternal Health Risk" dataset from the UCI Machine Learning Repository and was originally collected by Marzia Ahmed and her team. Our final classifier performed fairly well on an unseen test data set, with a weighted recall score of 0.77 and an overall accuracy of 0.77. Out of the 305 test data cases, it correctly predicted 235 cases. The model showed particularly strong performance in identifying high-risk pregnancies, achieving an AUV of 0.943 for the high-risk class, compared to 0.820 for low-risk class and 0.814 for medium-risk class. However, the model made notable errors where 13 high-risk cases were misclassified as 11 medium-risk and 2 low-risk. These false negatives are gaps where high-risk individuals may not receive the necessary care.

We recommend further research to improve the model's sensitivity to high-risk cases and better

differentiate between medium and low-risk categories before it is ready to be put into production in clinical settings. Additional feature engineering or exploring ensemble methods may help reduce these critical misclassifications. The implementation of refined classifiers would expand the capabilities of most healthcare systems and increase the efficacy of monitoring and interventions in underprivileged communities.

3 Introduction

Maternal mortality is a serious issue that predates human history and still affects many mothers today. Among women of reproductive age, 9% of global deaths are currently attributable to maternal causes such as hemorrhaging, hypertension, and unsafe abortion (Hassfurter, 2025). Fortunately, gradual improvements in medical understanding, policy, healthcare, and overall quality of life have led to steadily decreasing maternal mortality rates. Recent UNICEF reports from 2023 place the global maternal mortality ratio at 197 per 100,000 live births, which is approximately 40% less than the reported ratio from 2000. Despite these trends, many rural and underserved communities continue to experience higher rates of maternal mortality.

Bangladesh is a country that has an overall maternal mortality ratio of 196 per 100,000 live births; however, deeper investigation reveals significant differences in mortality rate between women of different socioeconomic backgrounds (Hossain et. al, 2023). Mortality rate was higher among women with no education, women in rural areas, and women in poor wealth categories. According to a paper published in The Lancet, focusing research on the biomedical causes of mortality is insufficient, and more attention should be directed towards needs such as "primary prevention, early identification, and adequate management of pregnancy, labour, and postpartum complications" (Souza et. al, 2024).

A welcome innovation has been the development of wearable technology and internet-enabled devices that have allowed patients and physicians to reliably monitor health conditions from their homes (Kashem et. al, 2020). Additionally, these devices enabled the collection of physiological data from otherwise underserved communities. We aim to predict maternal health risk levels using clinical measurements gathered from pregnant individuals in rural Bangladesh. The ability to accurately predict high-risk pregnancies would allow timely and focused medical interventions for vulnerable individuals.

4 Methods

4.1 Data

The data set used in this project is of health conditions of pregnant women from the rural areas of Bangladesh created by Marzia Ahmed at Daffodil International University. This dataset was sourced from the UC Irvine Machine Learning Repository and can be found here. Each observation in the dataset corresponds to a pregnant individual's health profile, comprising a risk intensity level (low, medium, or high risk) and associated clinical measurements including demographic information (age) and vital signs (systolic blood pressure, diastolic blood pressure, blood glucose concentration, body temperature, and resting heart rate). The data set was collected via an IoT-based risk monitoring system from hospitals, community clinics, and maternal health cares in rural Bangladesh.

4.1.1 Data dictionary

Column Name	Role	Type	Description
Age	Feature	Integer	Age of the patient during pregnancy (in years)
SystolicBP	Feature	Integer	Systolic (upper) blood pressure measured in mmHg
DiastolicBP	Feature	Integer	Diastolic (lower) blood pressure measured in mmHg
BS	Feature	Integer	Blood sugar level measured in mmol/L
BodyTemp	Feature	Integer	Body temperature of the patient measured in °F
HeartRate	Feature	Integer	Patient's resting heart rate measured in bpm
RiskLevel	Target	Categorical	Predicted pregnancy risk level based on clinical features

4.2 Analysis

SVC was used to build a classification model to predict risk levels for pregnant women in rural Bangladesh. With the exception of diastolic blood pressure, all variables from the original dataset were included for analysis. The features used were age, systolic blood pressure, blood glucose level (BS), body temperature, and heart rate. Data was partitioned into a 70:30 train-test split and the random_state was set to 123 for reproducibility. We performed hyperparameter tuning using randomized search with 10-fold cross-validation and recall score (weighted) as our evaluation metric to select the optimal values for C (regularization parameter), gamma (kernel coefficient). Recall score was selected to optimize the model for sensitivity in predicting high-risk cases. All explanatory variables were numerical and were standardized via StandardScalar prior to fitting. The Python programming language and the following Python packages were used to perform the analysis: requests, zipfile, numpy, Pandas, altair, seaborn, and scikit-learn.

4.3 Results & Discussion

4.3.1 EDA

Preliminary exploratory data analysis (EDA) was performed to briefly examine each explanatory variable. Previous research have described hypertension as a complication risk; therefore, we dropped diastolic BP for the more commonly significant systolic. Distributions of each explanatory variable were plotted using histograms and coloured according to risk levels (blue: high risk, green: medium risk, orange: low risk). The plotted distibutions were visually distinct across risk levels. Thus, we continued to fit our model with the remaining features.

[2]:
```
# download data as zip and extract
url = "https://archive.ics.uci.edu/static/public/863/maternal+health+risk.zip"

request = requests.get(url)
with open("../data/raw/maternal+health+risk.zip", 'wb') as f:
    f.write(request.content)

with zipfile.ZipFile("../data/raw/maternal+health+risk.zip", 'r') as zip_ref:
    zip_ref.extractall("../data/raw")
```

[3]:
```
health_data = pd.read_csv("../data/raw/Maternal Health Risk Data Set.csv",␣
 ↪header=0)
health_data
```

[3]:

	Age	SystolicBP	DiastolicBP	BS	BodyTemp	HeartRate	RiskLevel
0	25	130	80	15.0	98.0	86	high risk
1	35	140	90	13.0	98.0	70	high risk
2	29	90	70	8.0	100.0	80	high risk
3	30	140	85	7.0	98.0	70	high risk
4	35	120	60	6.1	98.0	76	low risk
…	…	…	…	…	…	…	…
1009	22	120	60	15.0	98.0	80	high risk
1010	55	120	90	18.0	98.0	60	high risk
1011	35	85	60	19.0	98.0	86	high risk
1012	43	120	90	18.0	98.0	70	high risk
1013	32	120	65	6.0	101.0	76	mid risk

[1014 rows x 7 columns]

[4]:
```
health_data.describe()
```

[4]:

	Age	SystolicBP	DiastolicBP	BS	BodyTemp	\
count	1014.000000	1014.000000	1014.000000	1014.000000	1014.000000	
mean	29.871795	113.198225	76.460552	8.725986	98.665089	
std	13.474386	18.403913	13.885796	3.293532	1.371384	
min	10.000000	70.000000	49.000000	6.000000	98.000000	
25%	19.000000	100.000000	65.000000	6.900000	98.000000	
50%	26.000000	120.000000	80.000000	7.500000	98.000000	
75%	39.000000	120.000000	90.000000	8.000000	98.000000	
max	70.000000	160.000000	100.000000	19.000000	103.000000	

	HeartRate
count	1014.000000
mean	74.301775
std	8.088702
min	7.000000
25%	70.000000

```
50%      76.000000
75%      80.000000
max      90.000000
```

[5]:
```
health_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1014 entries, 0 to 1013
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Age          1014 non-null   int64  
 1   SystolicBP   1014 non-null   int64  
 2   DiastolicBP  1014 non-null   int64  
 3   BS           1014 non-null   float64
 4   BodyTemp     1014 non-null   float64
 5   HeartRate    1014 non-null   int64  
 6   RiskLevel    1014 non-null   object 
dtypes: float64(2), int64(4), object(1)
memory usage: 55.6+ KB
```

[6]:
```
train_df, test_df = train_test_split(
    health_data, test_size=0.3, random_state=123
)
X_train, y_train = train_df.drop(columns=['RiskLevel']), train_df["RiskLevel"]
X_test, y_test = test_df.drop(columns=['RiskLevel']), test_df["RiskLevel"]

train_df.to_csv("../data/processed/maternal_health_risk_train.csv")
test_df.to_csv("../data/processed/maternal_health_risk_test.csv")
```

[7]:
```
train_df
```

[7]:

	Age	SystolicBP	DiastolicBP	BS	BodyTemp	HeartRate	RiskLevel
1003	50	130	100	16.0	98.0	76	high risk
243	32	120	65	6.0	101.0	76	mid risk
848	15	70	50	6.0	98.0	70	mid risk
202	23	90	60	7.5	98.0	76	low risk
300	15	75	49	7.7	98.0	77	low risk
…	…	…	…	…	…	…	…
988	25	120	90	12.0	101.0	80	high risk
322	65	90	60	6.9	98.0	70	low risk
382	17	90	65	7.8	103.0	67	high risk
365	22	120	90	7.8	98.0	82	mid risk
510	17	90	63	7.5	101.0	70	low risk

[709 rows x 7 columns]

[8]:
```
test_df
```

[8]:

	Age	SystolicBP	DiastolicBP	BS	BodyTemp	HeartRate	RiskLevel
50	25	120	80	7.0	98.0	66	low risk
784	35	100	70	6.8	98.0	60	mid risk
204	15	76	49	7.5	98.0	77	low risk
85	18	90	60	6.9	98.0	70	mid risk
802	42	130	80	18.0	98.0	70	mid risk
…	…	…	…	…	…	…	
619	29	130	70	7.5	98.0	78	mid risk
607	45	120	95	7.5	98.0	66	low risk
700	15	120	80	6.6	99.0	70	low risk
1005	17	90	65	7.7	103.0	67	high risk
178	40	120	95	11.0	98.0	80	high risk

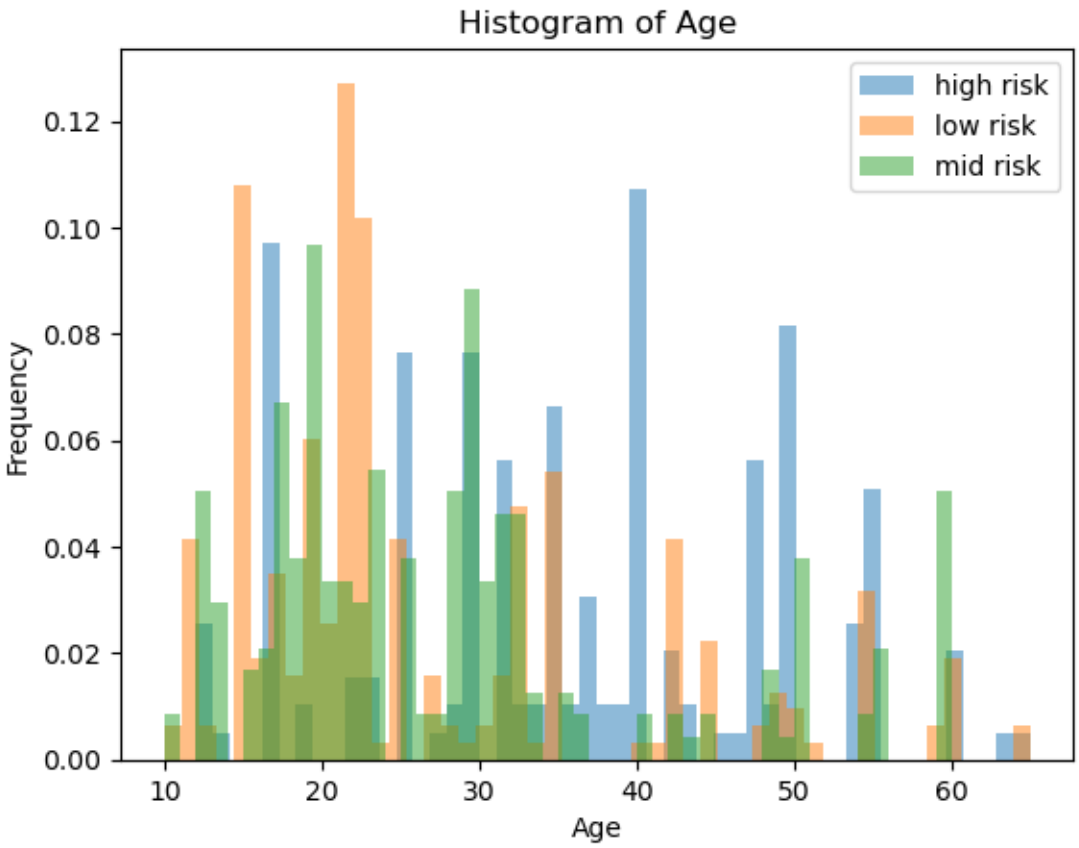
[305 rows x 7 columns]

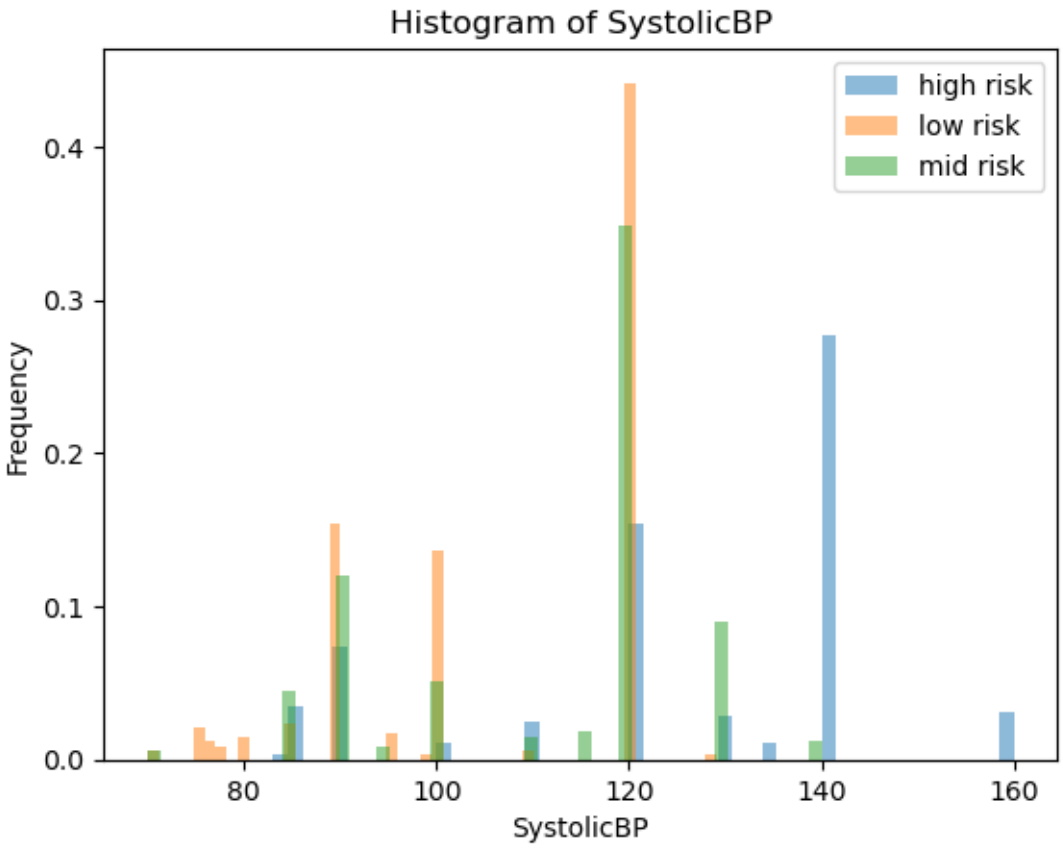
[9]:
```
y_train.value_counts()
```

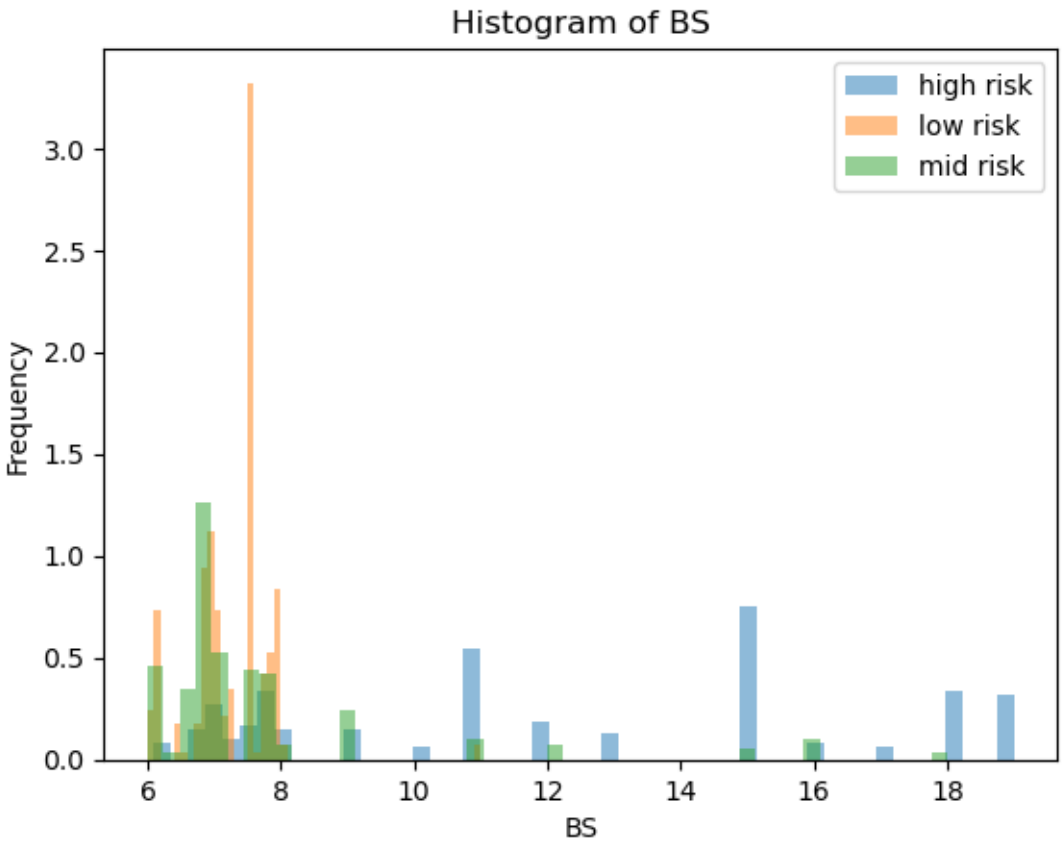
[9]:
```
RiskLevel
low risk     286
mid risk     238
high risk    185
Name: count, dtype: int64
```

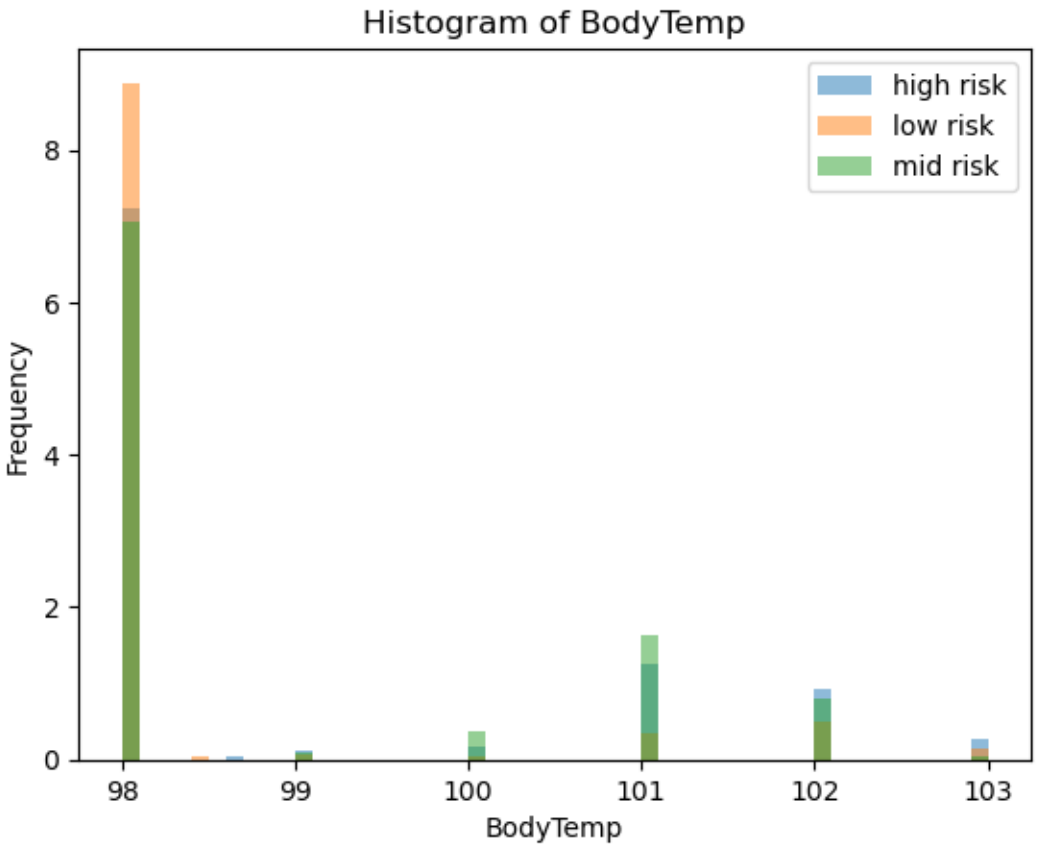
[10]:
```
feature_cols = ["Age", "SystolicBP", "BS", "BodyTemp", "HeartRate"]

for feature in feature_cols:
    train_df.groupby("RiskLevel")[feature].plot.hist(bins=50, alpha=0.5,␣
 ↪legend=True, density = True, title = "Histogram of " + feature);
    plt.xlabel(feature);
    plt.show()
```


Histogram of Age
high risk
low risk
mid risk
Frequency
0.00
0.02
0.04
0.06
0.08
0.10
0.12
10
20
30
40
50
60
Age


Histogram of SystolicBP
high risk
low risk
mid risk
Frequency
0.0
0.1
0.2
0.3
0.4
80
100
120
140
160
SystolicBP


Histogram of BS
high risk
low risk
mid risk
Frequency
BS
0.0
0.5
1.0
1.5
2.0
2.5
3.0
6
8
10
12
14
16
18


Histogram of BodyTemp
high risk
low risk
mid risk
Frequency
0
2
4
6
8
98
99
100
101
102
103
BodyTemp

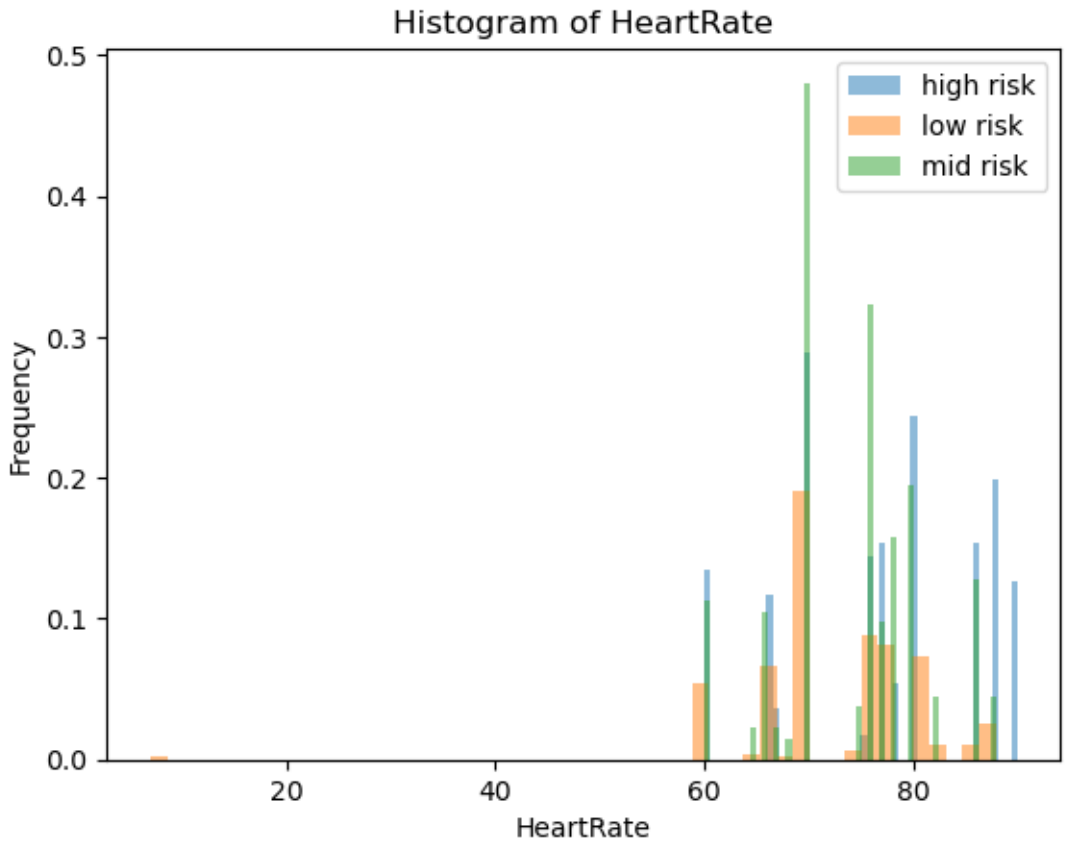


Figure 1. Comparison of the distributions of features contributing to the risk intensity level during pregnancy of an individual.

4.3.2 Model construction

We selected a Support Vector Classifier (SVC) model for this classification task. To identify the model configuration that best predicted maternal health risk levels, we performed hyperparameter tuning using randomized search with 10-fold cross-validation and recall score (weighted) as our evaluation metric to select the optimal values for C (regularization parameter), gamma (kernel coefficient). We found that the optimal hyperparameters were 760 for C and 5.8 for gamma.

[11]:
```
preprocessor = make_column_transformer(
    (StandardScaler(), feature_cols)
)
```

[12]:
```
preprocessor.fit(X_train)
```

[12]:
```
ColumnTransformer(transformers=[('standardscaler', StandardScaler(),
                                 ['Age', 'SystolicBP', 'BS', 'BodyTemp',
                                  'HeartRate'])])
```

[13]:
```
X_train_enc = pd.DataFrame(
    preprocessor.transform(X_train),
    index=X_train.index,
    columns=preprocessor.get_feature_names_out()
)

# Show the transformed data
X_train_enc
```

[13]:

	standardscaler__Age	standardscaler__SystolicBP	standardscaler__BS	standardscaler__BodyTemp	standardscaler__HeartRate
1003	1.539667	0.915275	2.278454	-0.499908	0.209123
243	0.182676	0.373190	-0.820502	1.630962	0.209123
848	-1.098927	-2.337235	-0.820502	-0.499908	-0.543507
202	-0.495820	-1.253065	-0.355659	-0.499908	0.209123
300	-1.098927	-2.066193	-0.293680	-0.499908	0.334561
…	…	…	…	…	…
988	-0.345043	0.373190	1.038872	1.630962	0.710876
322	2.670493	-1.253065	-0.541596	-0.499908	-0.543507
382	-0.948150	-1.253065	-0.262690	3.051541	-0.919822
365	-0.571208	0.373190	-0.262690	-0.499908	0.961753
510	-0.948150	-1.253065	-0.355659	1.630962	-0.543507

[709 rows x 5 columns]

[14]:
```
dc = DummyClassifier()
```

[15]:
```
dc_score = pd.DataFrame(cross_validate(dc, X_train, y_train, cv=5,␣
 ↪return_train_score=True))
```

[16]:
```
dc_score
```

[16]:

	fit_time	score_time	test_score	train_score
0	0.000631	0.000400	0.401408	0.403880
1	0.000305	0.000305	0.401408	0.403880

```
2  0.000314    0.000233    0.401408    0.403880
3  0.000261    0.000215    0.408451    0.402116
4  0.000252    0.000210    0.404255    0.403169
```

[17]:
```
svc = make_pipeline(preprocessor, SVC())
```

[18]:
```
svc_score = pd.DataFrame(cross_validate(svc, X_train, y_train, cv=5,␣
 ↪return_train_score=True))
```

[19]:
```
svc_score
```

[19]:
```
   fit_time  score_time  test_score  train_score
0  0.006774    0.002788    0.725352    0.707231
1  0.006165    0.002624    0.676056    0.719577
2  0.005901    0.002521    0.718310    0.719577
3  0.005892    0.002555    0.711268    0.708995
4  0.005446    0.002316    0.659574    0.727113
```

4.3.3 Hyperparameter tuning

Because we are attempting to classify clincal risk levels, we selected recall score as the preferred evaluation metric. This is because recall score measures the percentage correctly identified of actual high-risk pregnancies. This is critical for maternal health prediction where false negatives could result in missing high-risk, vulnerable individuals. Although prioritizing recall could increase the rate of false positive errors, it is the safer choice in this context.

[20]:
```
svc_score = pd.DataFrame(cross_validate(svc, X_train, y_train, cv=5,␣
 ↪return_train_score=True, scoring="recall_weighted"))
```

[21]:
```
svc_score
```

[21]:
```
   fit_time  score_time  test_score  train_score
0  0.006638    0.003394    0.725352    0.707231
1  0.005858    0.003036    0.676056    0.719577
2  0.005968    0.003039    0.718310    0.719577
3  0.005742    0.003326    0.711268    0.708995
4  0.005789    0.003167    0.659574    0.727113
```

[22]:
```
param_grid = {
    "svc__C": loguniform(1e-2, 1e3),
    "svc__gamma": loguniform(1e-4, 1e1)
}
```

[23]:
```
random_search = RandomizedSearchCV(svc,
                   param_distributions=param_grid,
                   n_iter=100,
                   n_jobs=-1,
                   return_train_score=True,
```

```
                   cv=10,
                   scoring='recall_weighted',
                   random_state=123)

random_search.fit(X_train, y_train)
```

[23]:
```
RandomizedSearchCV(cv=10,
                   estimator=Pipeline(steps=[('columntransformer',
ColumnTransformer(transformers=[('standardscaler',
StandardScaler(),
['Age',
'SystolicBP',
'BS',
'BodyTemp',
'HeartRate'])])),
                                             ('svc', SVC())]),
                   n_iter=100, n_jobs=-1,
                   param_distributions={'svc__C':
<scipy.stats._distn_infrastructure.rv_continuous_frozen object at 0x3050b7380>,
                                        'svc__gamma':
<scipy.stats._distn_infrastructure.rv_continuous_frozen object at 0x305a8ae90>},
                   random_state=123, return_train_score=True,
                   scoring='recall_weighted')
```

[24]:
```
random_search.best_score_
```

[24]:
```
np.float64(0.7629979879275655)
```

[25]:
```
result_grid = pd.DataFrame(random_search.cv_results_)
result_grid = result_grid[
    [
        "mean_test_score",
        "param_svc__gamma",
        "param_svc__C",
        "mean_fit_time",
        "rank_test_score",
    ]
].set_index("rank_test_score").sort_index().T.iloc[:, :10]

result_grid
```

[25]:
```
rank_test_score           1           2           3          4           5   \
mean_test_score    0.762998    0.760221    0.755956   0.753159    0.751771
param_svc__gamma   1.615660    8.282366    1.280916   2.749691    9.479652
param_svc__C     422.340226  323.263643  198.127782  21.103877   33.480671
mean_fit_time      0.013005    0.012636    0.010105   0.008967    0.009363
```

rank_test_score	6	7	8	9	10
mean_test_score	0.734849	0.733400	0.730624	0.729215	0.720644
param_svc__gamma	1.571147	8.468353	2.134628	1.766684	0.265551
param_svc__C	34.953203	2.601185	10.080750	14.859520	801.347695
mean_fit_time	0.014226	0.012603	0.008903	0.012903	0.025342

[26]:
```
plt.figure(figsize=(12, 5))
sns.heatmap(result_grid,
            annot=True,
            fmt='.3f',
            cmap='viridis',
            cbar_kws={'label': 'Value'},
            linewidths=0.5,
            linecolor='gray')

plt.title('Top 10 SVC Hyperparameter Combinations', fontsize=14, pad=20)
plt.xlabel('Rank', fontsize=12)
plt.ylabel('Parameter/Metric', fontsize=12)
plt.tight_layout()
plt.show()
```

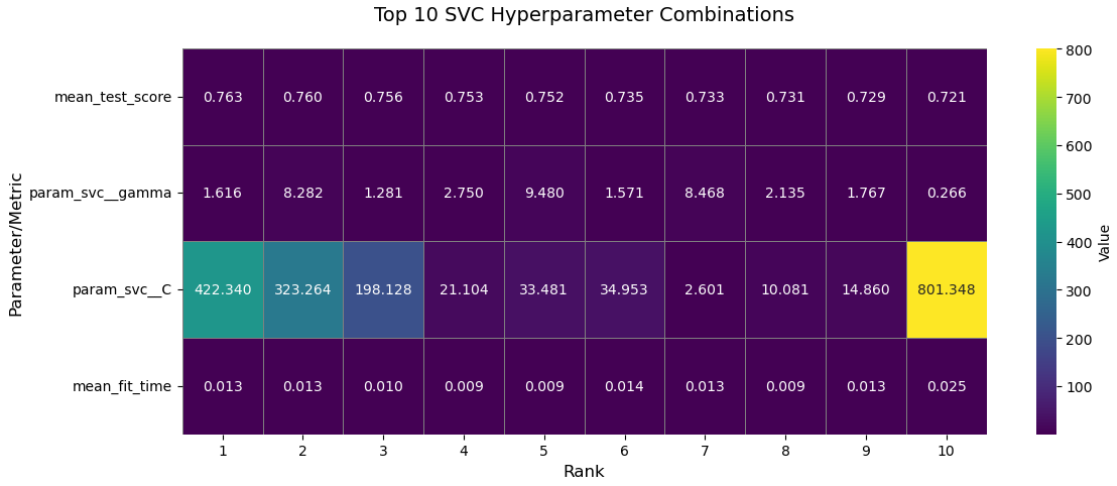


Figure 2. Results from hyperparameter optimization and 10-fold cross validation to choose gamma and C. Recall score was used as the classification metric as gamma and C was varied.

[27]:
```
accuracy_score = random_search.score(
    X_test, y_test
)
```

[28]:
```
accuracy_score
```

[28]: 0.7836065573770492

[29]:
```
maternal_preds = X_test.assign(
    predicted=random_search.best_estimator_.predict(X_test)
)

# Add the actual labels
maternal_preds['actual'] = y_test.values

# Compute recall score (weighted for multi-class)
recall = recall_score(
    maternal_preds['actual'],
    maternal_preds['predicted'],
    average='weighted'
)
```

[30]:
```
recall
```

[30]:
```
0.7836065573770492
```

[31]:
```
pd.crosstab(
    maternal_preds["actual"],
    maternal_preds["predicted"],
)
```

[31]:

actual	high risk	low risk	mid risk
high risk	74	2	11
low risk	4	95	21
mid risk	9	19	70

[32]:
```
confmat_logreg_bal = ConfusionMatrixDisplay.from_predictions(
    y_test,
    random_search.best_estimator_.predict(X_test),
    #normalize='all'
)
confmat_logreg_bal.ax_.set_title('Confusion Matrix - Test Data', fontsize=14,␣
↪fontweight='bold')
```

[32]:
```
Text(0.5, 1.0, 'Confusion Matrix - Test Data')
```

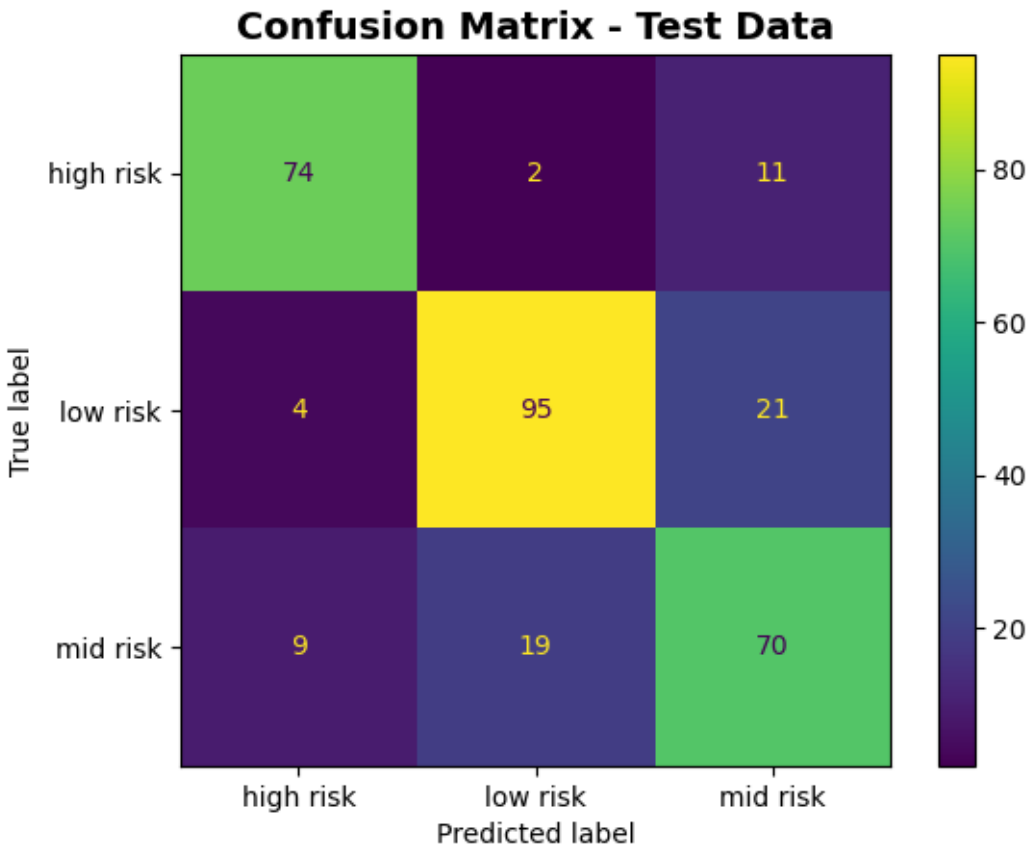


Figure 3. Confusion matrix of model performance on test data.

[33]:
```
y_score = random_search.best_estimator_.decision_function(X_test)
svc_classes = random_search.best_estimator_.named_steps['svc'].classes_
y_test_bin = label_binarize(y_test, classes=svc_classes)
fig, ax = plt.subplots(figsize=(6, 8))

for i, class_name in enumerate(svc_classes):
    auc = roc_auc_score(y_test_bin[:, i], y_score[:, i])
    RocCurveDisplay.from_predictions(
        y_test_bin[:, i],
        y_score[:, i],
        name=f'{class_name} (AUC = {auc:.3f})',
        ax=ax
    )
    print(f"{class_name}: AUC = {auc:.3f}")
```

```
high risk: AUC = 0.952
low risk: AUC = 0.885
mid risk: AUC = 0.854
```

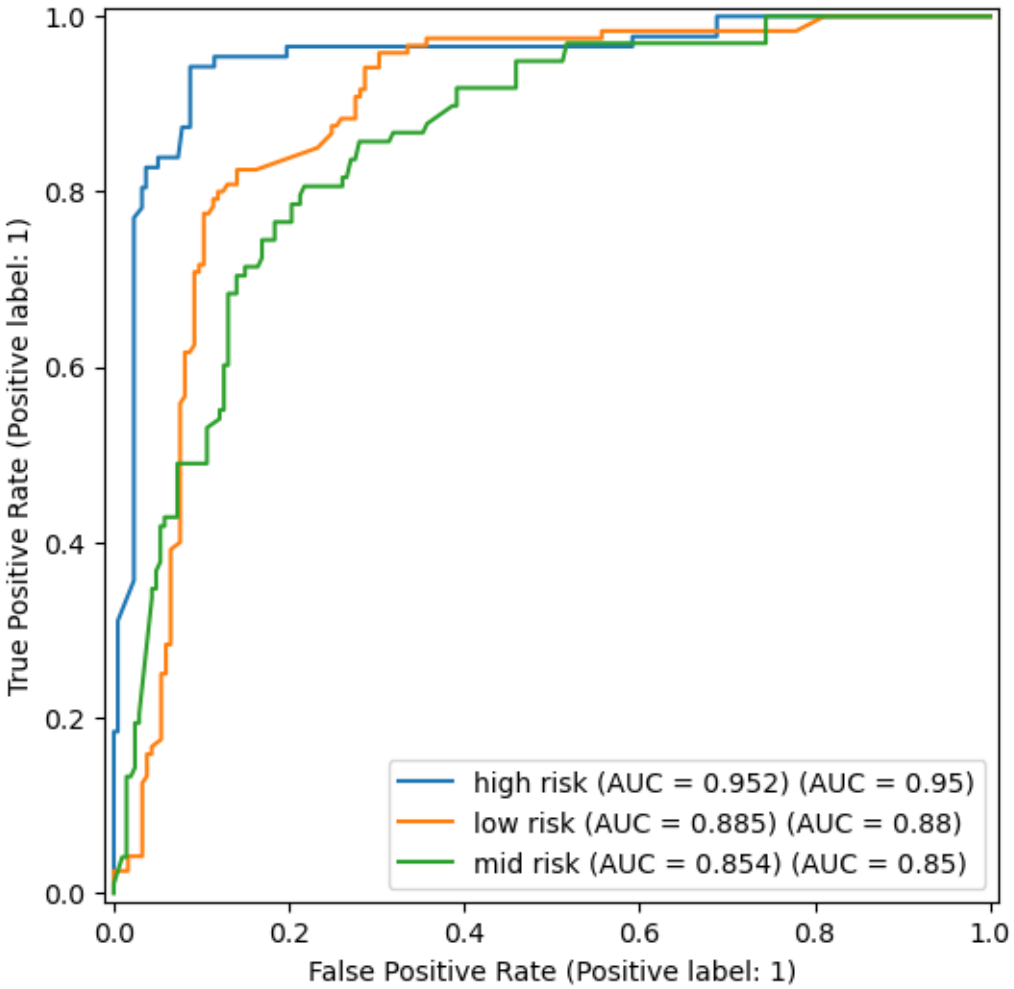


Figure 4. ROC curve of model performance on test data.

4.3.4 Summary of Model Results

The final SVC classifier performed fairly well on an unseen test data set, with a weighted recall score of 0.77 and an overall accuracy of 0.77. Out of the 305 test data cases, it correctly predicted 235 cases. The model showed particularly strong performance in identifying high-risk pregnancies, achieving an AUV of 0.943 for the high-risk class, compared to 0.820 for low-risk class and 0.814 for medium-risk class. It correctly identifies 74 out of 87 actual high-risk pregnancies resulting in a 85% recall for high risk. The 13 notable errors were true high-risk cases were misclassified as 11 medium-risk and 2 low-risk. These false negatives are gaps where high-risk individuals may not receive the necessary care. We recommend further research to improve the model's sensitivity to high-risk cases and better differentiate between medium and low-risk categories before it is ready to be put into production in clinical settings.

4.4 References

A.B.M. Sharif Hossain, Siddique, A., Jabeen, S., Khan, S., M Moinuddin Haider, Ameen, S., Tazeen Tahsina, Chakraborty, N., Nahar, Q., Jamil, K., Shams El Arifeen, & Ahmed Ehsanur Rahman. (2023). Maternal mortality in Bangladesh: Who, when, why, and where? A national survey-based analysis. Journal of Global Health, 13. https://doi.org/10.7189/jogh.13.07002

Ahmed, M., Kashem, M. A., Rahman, M., & Khatun, S. (2020). Review and Analysis of Risk Factor of Maternal Health in Remote Area Using the Internet of Things (IoT). Lecture Notes in Electrical Engineering, 357–365. https://doi.org/10.1007/978-981-15-2317-5_30

Hassfurter, K. (2025, April 6). Trends in maternal mortality 2000 to 2023 - UNICEF DATA. UNICEF DATA. https://data.unicef.org/resources/trends-in-maternal-mortality-2000-to-2023/

Souza, J. P., Day, L., Rezende-Gomes, A. C., Zhang, J., Mori, R., Baguiya, A., Jayaratne, K., Osoti, A., Vogel, J. P., Campbell, R., Mugerwa, K., Lumbiganon, P., Tunçalp, ., Cresswell, J. A., Say, L., Moran, A. C., & Oladapo, O. T. (2023). A Global Analysis of the Determinants of Maternal Health and Transitions in Maternal Mortality. The Lancet Global Health, 12(2). https://doi.org/10.1016/s2214-109x(23)00468-0