Predicting Maternal Health Risk

Introduction to Programming Capstone Project: Job Thomas (24MM60007), Kundan Singh Rathore (24MM60010), Suhrt KR (24MM60015) students of MMST in SMST

Data Summary

The dataset, sourced from the UC Irvine Machine Learning Repository (found at https://archive.ics.uci.edu/dataset/863/maternal+health+risk), contains 1,014 entries and is characterized by 7 attributes. These attributes are:

1. Age
2. Systolic BP (Blood Pressure)
3. Diastolic BP (Blood Pressure)
4. Blood Sugar (BS)
5. Body Temperature
6. Heart Rate
7. Risk Level

Out of the above all except Risk Level, were continuous data. Risk level was categorical data taking 3 values (low risk, mid risk, high risk)

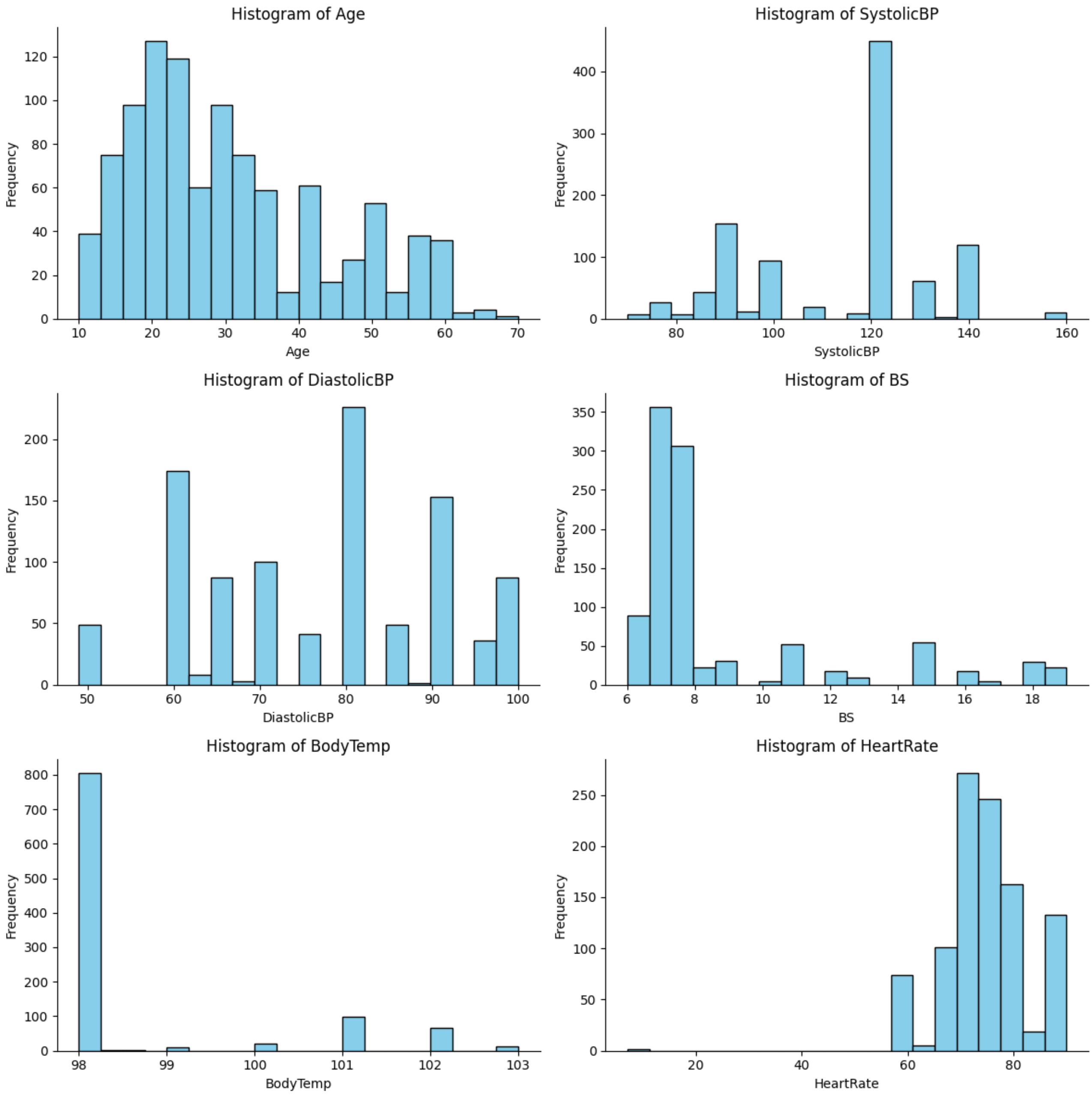
There were no missing data. Summary characteristic of the data are summarised bellow.

	Age	SystolicBP	DiastolicBP	BS	BodyTemp	HeartRate
count	1014.000000	1014.000000	1014.000000	1014.000000	1014.000000	1014.000000
mean	29.871795	113.198225	76.460552	8.725986	98.665089	74.301775
std	13.474386	18.403913	13.885796	3.293532	1.371384	8.088702
min	10.000000	70.000000	49.000000	6.000000	98.000000	7.000000
25%	19.000000	100.000000	65.000000	6.900000	98.000000	70.000000
50%	26.000000	120.000000	80.000000	7.500000	98.000000	76.000000
75%	39.000000	120.000000	90.000000	8.000000	98.000000	80.000000
max	70.000000	160.000000	100.000000	19.000000	103.000000	90.000000

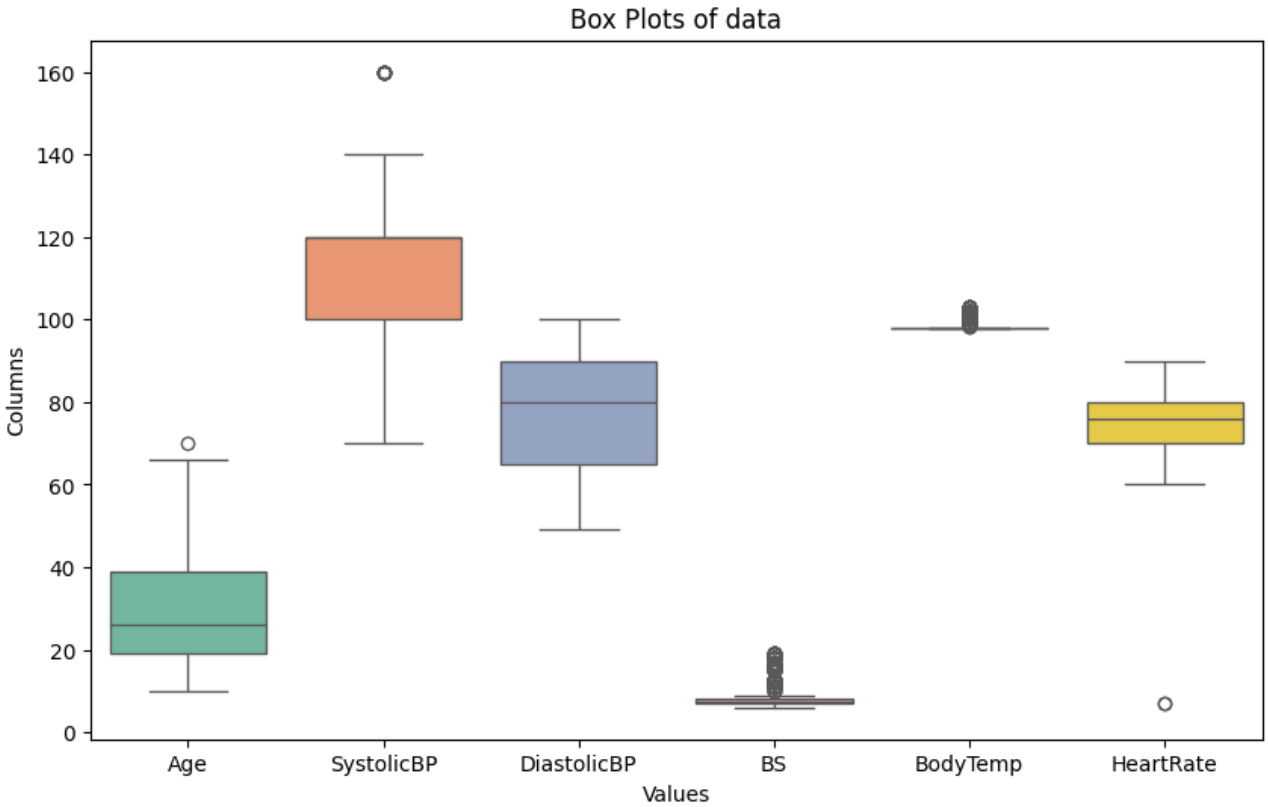
Risk Level
Low Risk: 406 | Med Risk: 336 | High Risk: 272

Data Visualisation

Univariate analysis

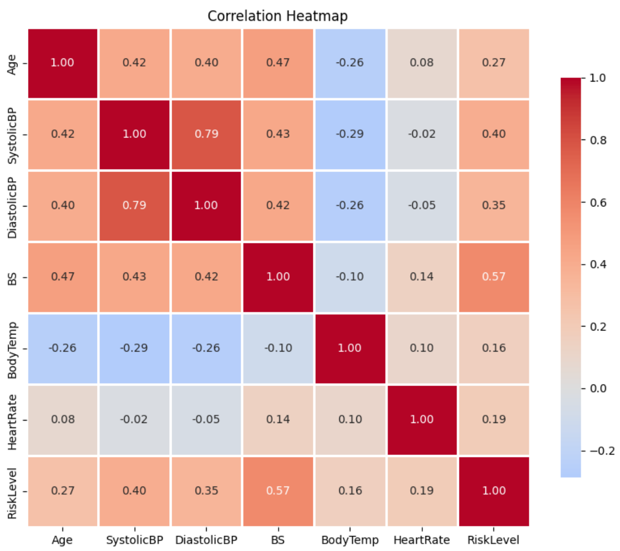


Histogram Plots of Data

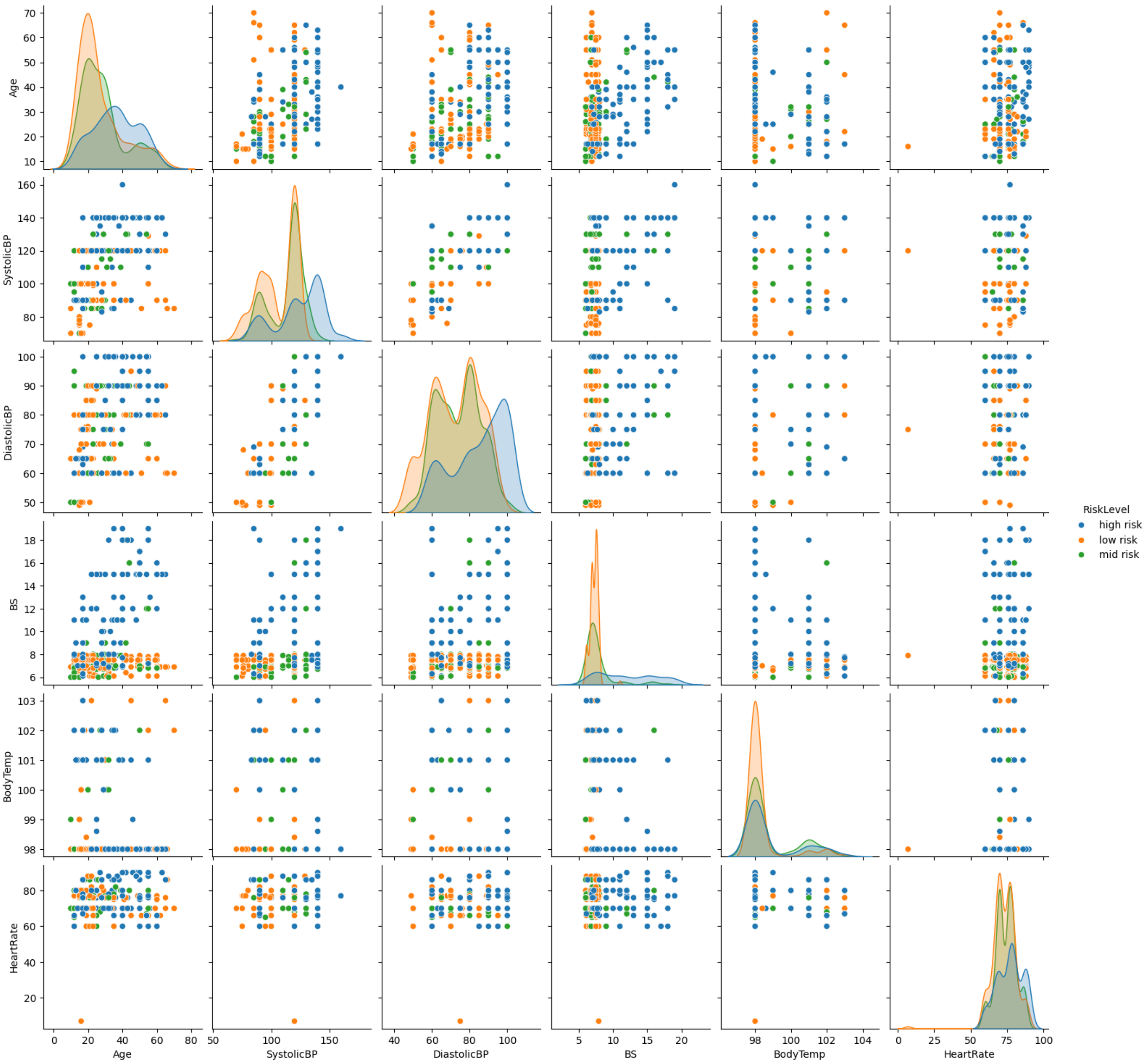


Box plot of data

Multivariate analysis



Correlation Heatmap



Scatter Plot With Risk Level As The Hue

Model Training

The following models were trained, using hyper parameters mentioned below along with the best params

Multinomial Logistic Regression

```
param_grid = [
    {
        "penalty": ["l1"],
        "solver": ["liblinear", "saga"],
        "C": np.logspace(-4, 4, 20),
        "max_iter": [100, 1000, 2500, 5000],
    },
    {
        "penalty": ["l2"],
        "solver": ["newton-cg", "lbfgs", "sag"],
        "C": np.logspace(-4, 4, 20),
        "max_iter": [100, 1000, 2500, 5000],
    },
]
```

Best parameters {'C': 0.012742749857031334, 'max_iter': 100, 'penalty': 'l2', 'solver': 'newton-cg'}

Decision Trees

```
param_grid = {
    "max_depth": [10, 20, 30, None],
    "min_samples_split": [2, 5, 10],
    "min_samples_leaf": [1, 2, 4],
    "criterion": ["gini", "entropy", "log_loss"],
    "ccp_alpha": [0.0, 0.01, 0.1],
}
```

Best parameters {'ccp_alpha': 0.0, 'criterion': 'gini', 'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 2}

Random Forest

```
param_grid = {
    "max_features": ["sqrt", "log2", None],
    "min_samples_split": [2, 5, 10],
    "min_samples_leaf": [1, 2, 4],
    "bootstrap": [True, False],
    "criterion": ["gini", "entropy"],
}
```

Best parameters {'ccp_alpha': 0.0, 'criterion': 'gini', 'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 2}

Support Vector Machines (SVM)

```
param_grid = {
    "C": [0.1, 1, 10, 100],
    "kernel": ["linear", "rbf", "poly", "sigmoid"],
    "gamma": ["scale", "auto"],
    "degree": [2, 3, 4],
    "class_weight": [None, "balanced"],
}
```

Best parameters {'bootstrap': False, 'criterion': 'gini', 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 2}

Gaussian Naive Bayes

param_grid = {"var_smoothing": [1e-9, 1e-8, 1e-7, 1e-6, 1e-5]}

Best parameters {'var_smoothing': 1e-09}

K-Nearest Neighbors (KNN)

```
param_grid = {
    'n_neighbors': [7, 9, 11, 15, 17, 19],
    'weights': ['uniform', 'distance'],
    'metric': [‘minkowski',],
    'p': [1, 2, 3 , 4],
    'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],  # Search algorithm
}
```

Best parameters {'algorithm': 'auto', 'metric': 'minkowski', 'n_neighbors': 9, 'p': 2, 'weights': 'distance'}

Model Evaluation

Models were evaluated using accuracy, precision, recall, and F1 score

Model	Accuracy	Prescision	Recall	F1 Score
Logistic Regression	0.635468	0.663357	0.647205	0.614348
Decision Tree	0.817734	0.821843	0.822723	0.820975
Random Forest	0.817734	0.827873	0.822723	0.823679
SVM	0.665025	0.677477	0.684929	0.676467
Gaussian Naive Bayes	0.576355	0.623634	0.578336	0.548915
KNN	0.817734	0.825429	0.82521	0.824968

Key Findings

Random Forest and KNN achieved the highest accuracy and F1 scores (81.8% accuracy and F1 ~0.825), making them the best-performing models.

Logistic Regression and SVM underperformed due to the categorical nature of the target variable and the complexity of relationships.

IMPROOVEMENTS

SMOTE

As the data was imbalanced, Synthetic Minority Oversampling Technique (SMOTE) was used. This increased the model accuracy. The following shows the improvement in scores using SMOTE.

Model	Accuracy	Prescision	Recall	F1 Score
Logistic Regression	0.668033	0.677562	0.67316	0.674605
Decision Tree	0.868852	0.881109	0.873959	0.874941
Random Forest	0.860656	0.872522	0.864416	0.867209
SVM	0.737705	0.757494	0.74209	0.747796
Gaussian Naive Bayes	0.602459	0.645654	0.603156	0.584975
KNN	0.860656	0.878333	0.862024	0.866239