Data Analytics & Visualization (CS/IT 312) Mini Project Submission & Feedback Form

Student Information

- Student Name: RAHUL GUPTA

- Mini Project Title: ECG Signal Classification Using Feature Engineering and Machine Learning Models (Random Forest and SVM)

- Student Roll No.: 202211069

- Are you working with anyone else on the same project (data)? Yes / No (If yes, mention their name and roll number.) : No

Mini Project Details

1. Type of Data Considered

- **Type**: 1D ECG Signal (Time-Series)
- **Description**: The dataset consists of electrocardiogram (ECG) signals, where each sample is a sequence of 140 voltage measurements (time points) representing a single ECG waveform, along with a binary label (0 = normal, 1 = abnormal).

First 5 rows of the dataset:

	signal_0	signal_1	signal_2	signal_3	signal_4	signal_5	signal_6	signal_7	signal_8	signal_9	...	signal_131	signal_132	signal_133	signal_134	signal_135	signal_136	signal_137	signal_138	si
0	-0.112522	-2.827204	-3.773897	-4.349751	-4.376041	-3.474986	-2.181408	-1.818286	-1.250522	-0.477492	...	0.792168	0.933541	0.796958	0.578621	0.257740	0.228077	0.123431	0.925286	
1	-1.100878	-3.996840	-4.285843	-4.506579	-4.022377	-3.234368	-1.566126	-0.992258	-0.754680	0.042321	...	0.538356	0.656881	0.787490	0.724046	0.555784	0.476333	0.773820	1.119621	
2	-0.567088	-2.593450	-3.874230	-4.584095	-4.187449	-3.151462	-1.742940	-1.490659	-1.183580	-0.394229	...	0.886073	0.531452	0.311377	-0.021919	-0.713683	-0.532197	0.321097	0.904227	
3	0.490473	-1.914407	-3.616364	-4.318823	-4.268016	-3.881110	-2.993280	-1.671131	-1.333884	-0.965629	...	0.350816	0.499111	0.600345	0.842069	0.952074	0.990133	1.086798	1.403011	
4	0.800232	-0.874252	-2.384761	-3.973292	-4.338224	-3.802422	-2.534510	-1.783423	-1.594450	-0.753199	...	1.148884	0.958434	1.059025	1.371682	1.277392	0.960304	0.971020	1.614392	

5 rows × 141 columns

2. Number of Observations / Subjects

- **Total Number of Samples**: 4,998
- **Categories (Samples per Class)**:
  - **Label 0 (Normal)**: 2,079 samples (41.6%)
  - **Label 1 (Abnormal)**: 2,919 samples (58.4%)

3. Project Type

- **Type**: Classification (Binary)
- **Description**: The goal is to classify ECG signals as normal (0) or abnormal (1) based on extracted features, making this a supervised binary classification task.

4. Data Source

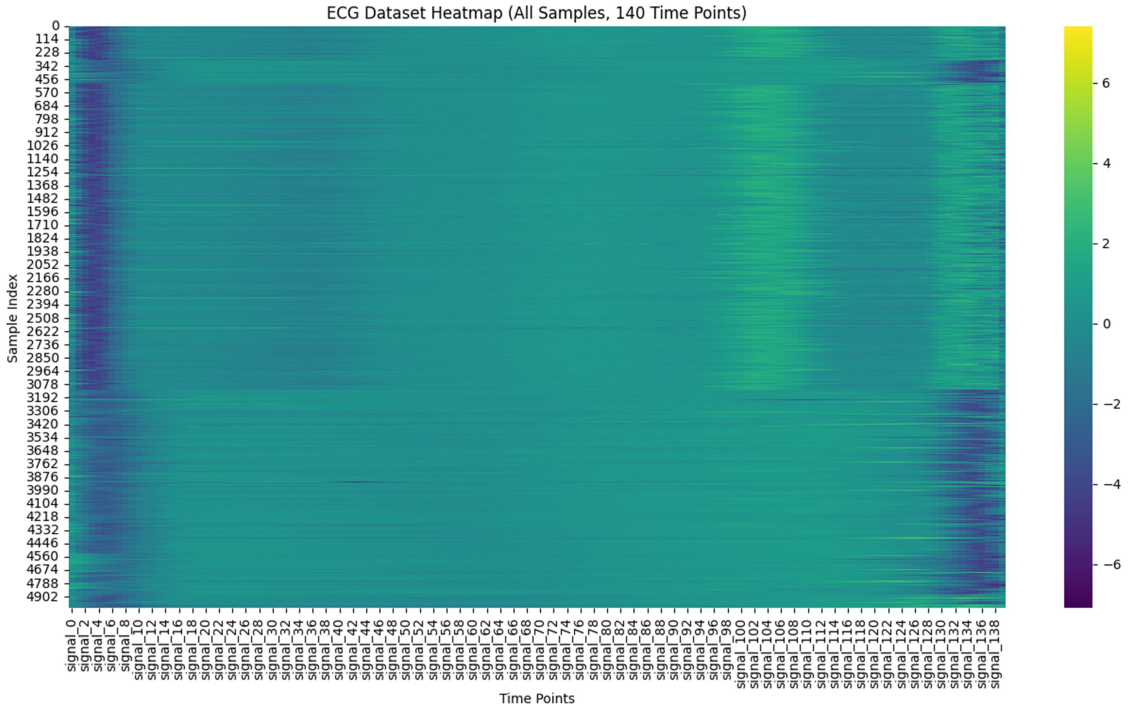
- **URL**: Kaggle Link (https://www.kaggle.com/datasets/devavratatripathy/ecg-dataset?select=ecg.csv)

5. Data Representation Before Feature Extraction

- **Raw Data Representation**:

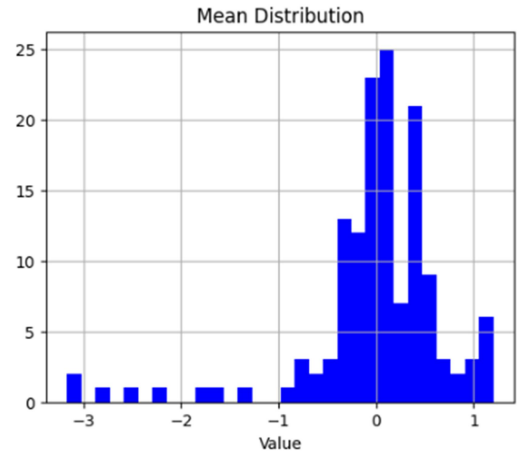
  ECG Dataset Heatmap (All Samples, 140 Time Points):

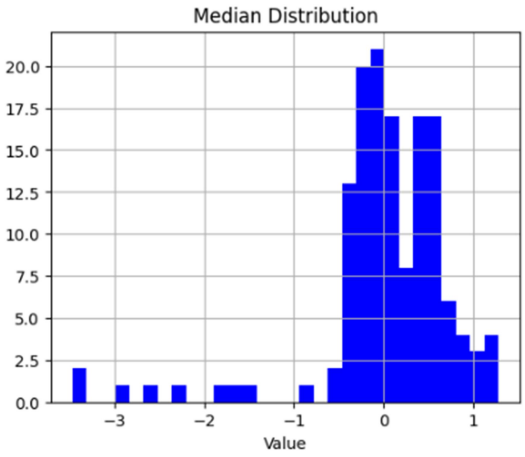
  - **Description**: A heatmap visualizing the entire dataset (4,998 samples × 140 signal columns).
  - **Reason**: To observe patterns in signal intensity across samples and time points, identifying variations that may correspond to normal or abnormal ECGs.

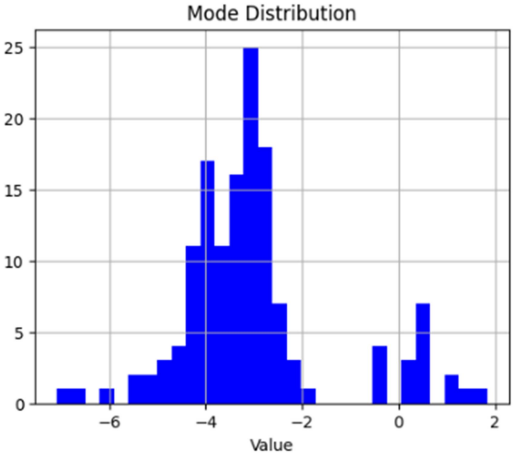


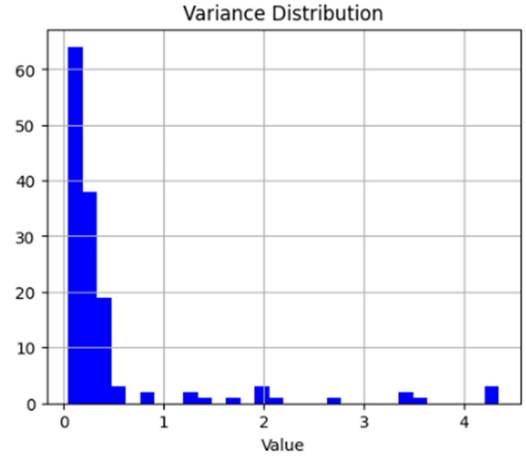
- **Descriptive Statistics Visualizations**:

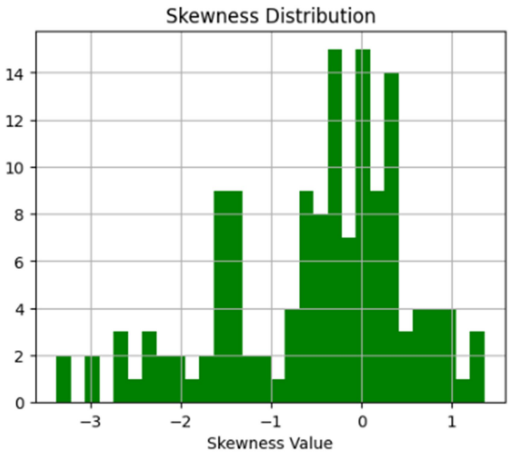
  - **Description**:
    - Histograms for mean, median, mode, variance, skewness, and kurtosis distributions across the 140 signal columns.

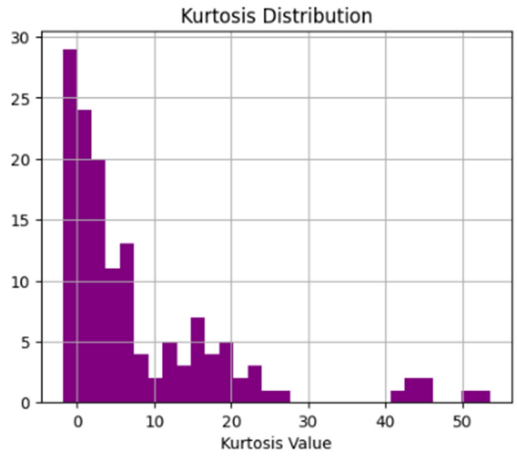




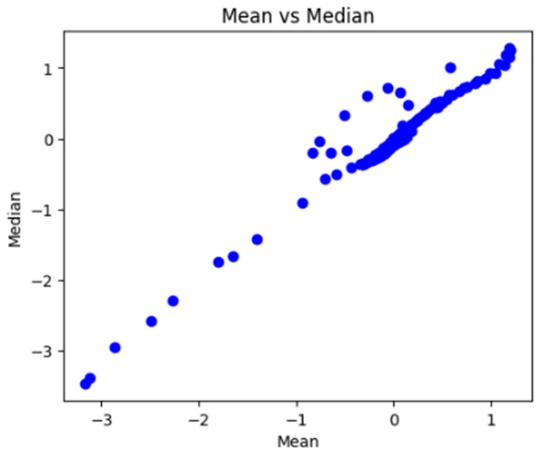


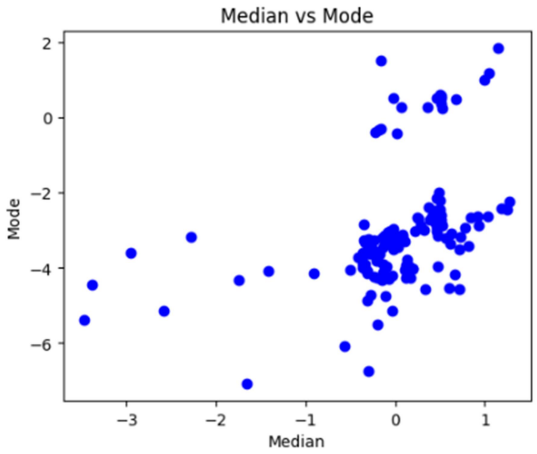


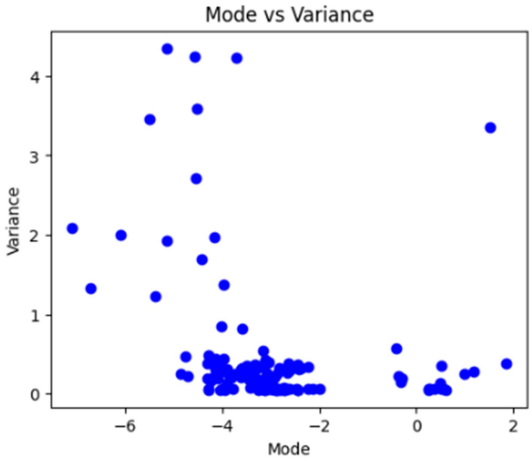


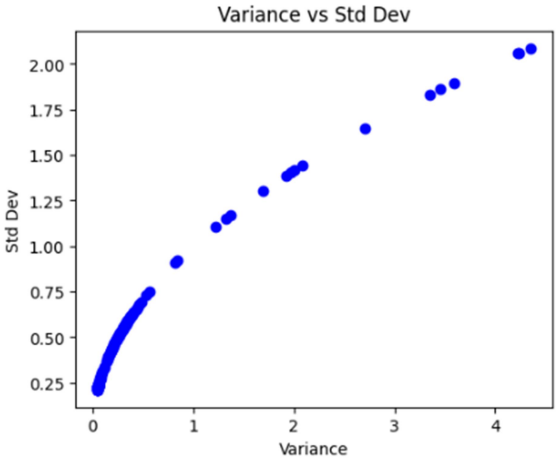


- Scatter plots (e.g., mean vs. median, median vs. mode, mode vs. variance, variance vs. std dev).

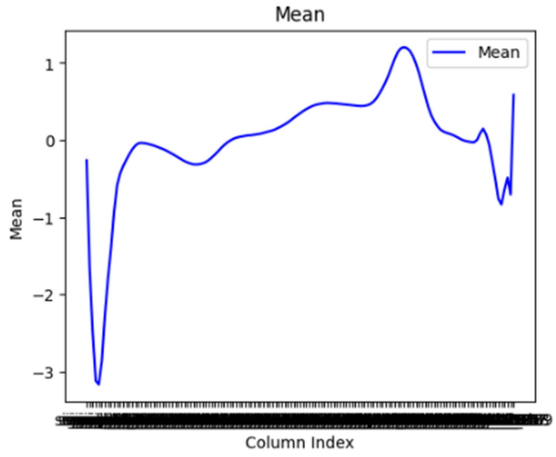


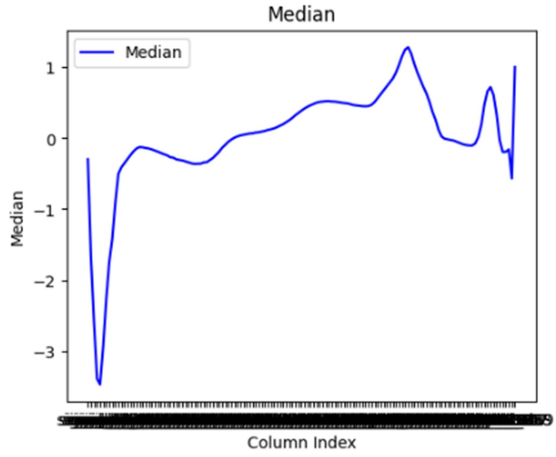


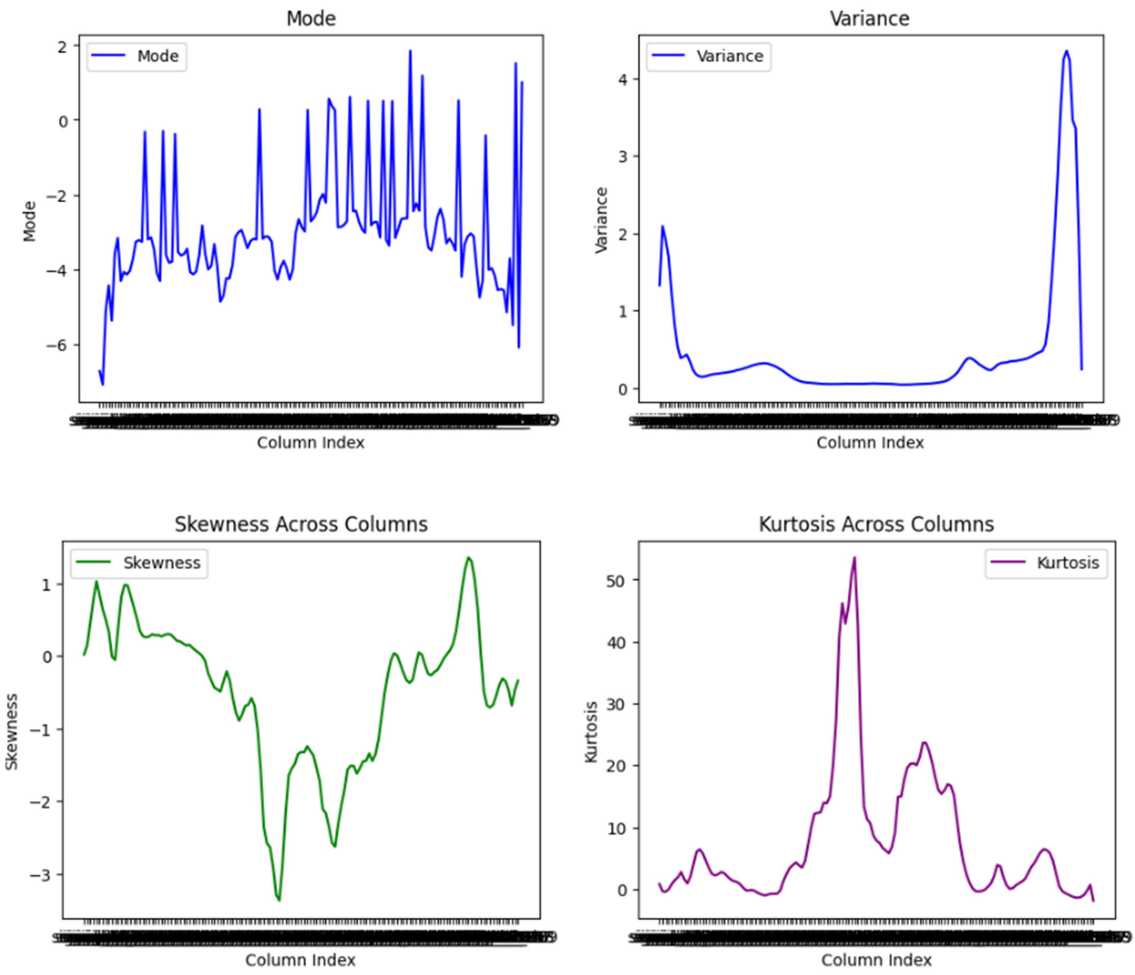




- Time-series plots for mean, median, mode, variance, skewness, and kurtosis across column indices.







- **Reason**: To explore statistical properties of the signal columns, assess data distribution, and identify potential outliers or trends relevant to feature extraction.

- **Total Graphs**: 17 (1 heatmap + 16 Statistics Visualizations).
- **Reason for Graphs**: These visualizations provide a high-level understanding of the raw data's structure, variability, and statistical characteristics, guiding feature engineering decisions.

6. Feature Extraction / Creation Details

- **Total Number of Features Extracted**: 24
- **Feature Names, Formulas, and Explanations**: Let x be the ECG signal with 140 values (x[1], x[2], ..., x[140]), and let avg(x) be the average of non-missing values. Let n be the count of non-missing values.
    1. **mean**:
        - **Formula**: avg(x) = sum of x[i] / n
        - **Explanation**: Average signal value, capturing the overall signal level.
    2. **std**:
        - **Formula**: std = sqrt(sum of (x[i] - avg(x))^2 / (n - 1))

- **Explanation**: Measures how much the signal varies from the average.

3. **skewness**:
   - **Formula**: skewness = (sum of (x[i] - avg(x))^3 / n) / ((sum of (x[i] - avg(x))^2 / n)^(3/2))
   - **Explanation**: Measures asymmetry of signal distribution, useful for detecting abnormal patterns.
4. **kurtosis**:
   - **Formula**: kurtosis = (sum of (x[i] - avg(x))^4 / n) / ((sum of (x[i] - avg(x))^2 / n)^2) - 3
   - **Explanation**: Measures tailedness, indicating extreme values in ECG signals.
5. **range**:
   - **Formula**: range = max(x) - min(x)
   - **Explanation**: Difference between maximum and minimum signal values, capturing signal amplitude.
6. **rms**:
   - **Formula**: rms = sqrt(sum of x[i]^2 / n)
   - **Explanation**: Root mean square, representing signal energy.
7. **zcr**:
   - **Formula**: zcr = (count of times x[i] * x[i+1] < 0) / (140 - 1)
   - **Explanation**: Zero-crossing rate, indicating how often the signal changes sign.
8. **peak_count**:
   - **Formula**: peak_count = number of peaks where x[i] > avg(x) + std and peaks are at least 10 samples apart
   - **Explanation**: Number of R-peaks, critical for heart rate estimation.
9. **psd_mean**:
   - **Formula**: psd_mean = avg(P), where P is power spectral density values from Welch's method (fs=360 Hz, segment length=140)
   - **Explanation**: Average power in frequency domain, capturing energy distribution.
10. **dominant_freq**:
    - **Formula**: dominant_freq = frequency where P is maximum
    - **Explanation**: Frequency with the highest power, indicating primary signal rhythm.
11. **psd_total**:
    - **Formula**: psd_total = sum of P
    - **Explanation**: Total power in frequency domain, representing overall energy.
12. **fft_max**:
    - **Formula**: fft_max = max of absolute FFT values for first half of frequencies
    - **Explanation**: Maximum amplitude in frequency domain, capturing dominant frequency components.
13. **band_energy_ratio**:
    - **Formula**: band_energy_ratio = (sum of P for frequencies 0.5 to 5 Hz) / (sum of P for 0.5 to 5 Hz + sum of P for 5 to 40 Hz), or 0 if denominator is 0

- **Explanation**: Ratio of low-frequency to total energy, relevant for heart rate vs. QRS complex.

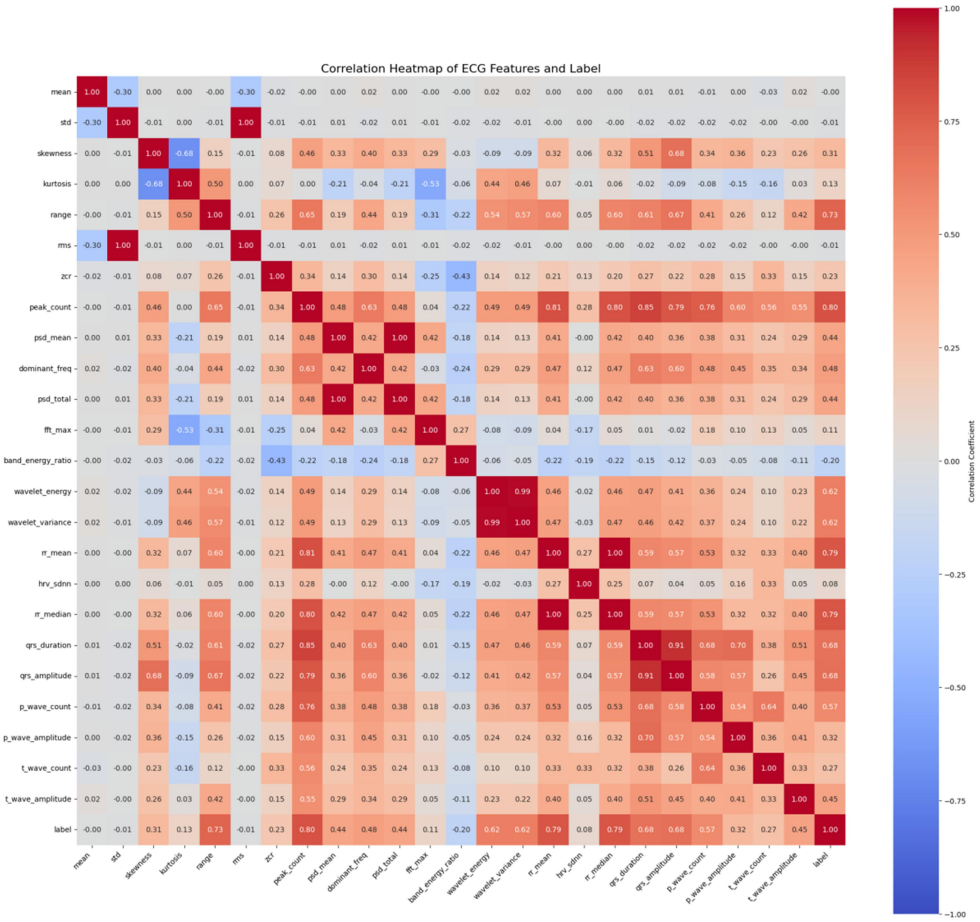
14. **wavelet_energy**:
    - **Formula**: wavelet_energy = sum of c[k]^2 for all wavelet coefficients c[k] from 4-level db4 decomposition
    - **Explanation**: Total energy of wavelet coefficients, capturing time-frequency patterns.
15. **wavelet_variance**:
    - **Formula**: wavelet_variance = avg of (sum of (c[k] - avg(c))^2 / (length(c) - 1)) for each wavelet level
    - **Explanation**: Average variance of wavelet coefficients, indicating signal complexity.
16. **rr_mean**:
    - **Formula**: rr_mean = avg of (p[i+1] - p[i]) for peak indices p, or 0 if fewer than 2 peaks
    - **Explanation**: Average time between R-peaks, related to heart rate.
17. **hrv_sdnn**:
    - **Formula**: hrv_sdnn = sqrt(sum of ((p[i+1] - p[i]) - rr_mean)^2 / (number of intervals - 1)), or 0 if fewer than 2 peaks
    - **Explanation**: Standard deviation of RR intervals, measuring heart rate variability.
18. **rr_median**:
    - **Formula**: rr_median = middle value of (p[i+1] - p[i]), or 0 if fewer than 2 peaks
    - **Explanation**: Median RR interval, robust to outliers.
19. **qrs_duration**:
    - **Formula**: qrs_duration = avg of (min(140, p[i] + 5) - max(0, p[i] - 5)) for all peaks, or 0 if no peaks
    - **Explanation**: Average QRS complex duration, critical for detecting abnormalities.
20. **qrs_amplitude**:
    - **Formula**: qrs_amplitude = avg of x[p[i]] for all peak indices, or 0 if no peaks
    - **Explanation**: Average R-peak amplitude, indicating QRS strength.
21. **p_wave_count**:
    - **Formula**: p_wave_count = sum of number of peaks in [max(0, p[i] - 20), max(0, p[i] - 5)] with height > 0.5 * avg(x), or 0 if no peaks
    - **Explanation**: Number of P-waves, related to atrial activity.
22. **p_wave_amplitude**:
    - **Formula**: p_wave_amplitude = avg of x[q] for all P-wave peaks q, or 0 if no P-waves
    - **Explanation**: Average P-wave amplitude, indicating atrial depolarization strength.
23. **t_wave_count**:
    - **Formula**: t_wave_count = sum of number of peaks in [min(140, p[i] + 5), min(140, p[i] + 30)] with height > 0.7 * avg(x), or 0 if no peaks
    - **Explanation**: Number of T-waves, related to ventricular repolarization.

24. **t_wave_amplitude**:
    - **Formula**: t_wave_amplitude = avg of x[t] for all T-wave peaks t, or 0 if no T-waves
    - **Explanation**: Average T-wave amplitude, indicating repolarization strength.

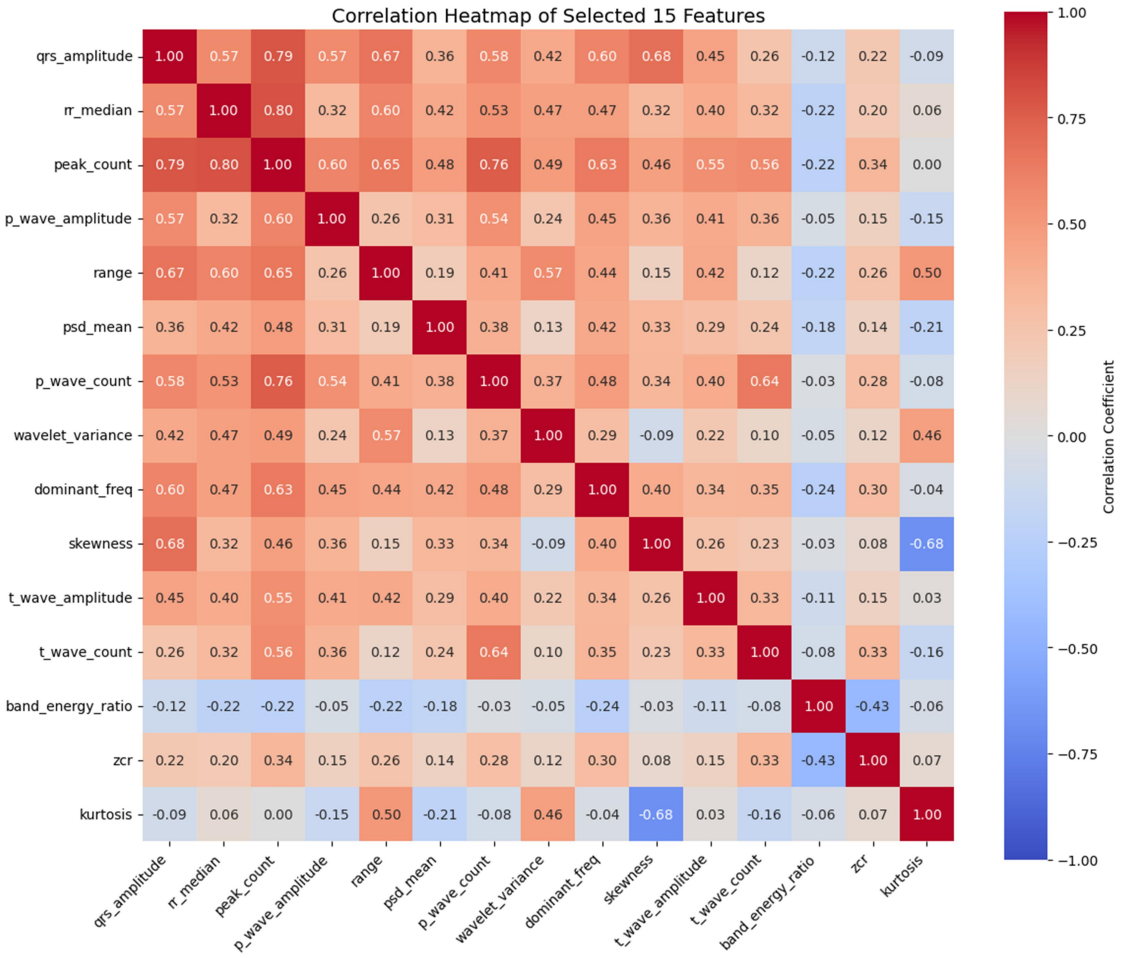
- **Use**: These features capture statistical, time-domain, frequency-domain, and ECG-specific characteristics, enabling robust classification of normal vs. abnormal ECGs.

7. Data Representation After Feature Extraction

- **Representation**:
    - The extracted features are stored in a Pandas DataFrame (feature_df) with 4,998 rows and 25 columns (24 features + 1 label).
- **Graphs Used** (Total: 2):
    1. **Correlation Heatmap of ECG Features and Label**:
        - **Description**: A 25×25 heatmap showing correlations among all 24 features and the label.
        - **Reason**: To identify highly correlated features for potential redundancy reduction and assess feature-label correlations for predictive power.



2. **Correlation Heatmap of Selected 15 Features**:
    - **Description**: A 15×15 heatmap showing correlations among the selected 15 features.
    - **Reason**: To confirm that selected features have low inter-feature correlations (< 0.9), ensuring minimal redundancy.



- **Total Graphs**: 2 heatmaps.
- **Reason for Choosing Graphs**:
    - Heatmaps are ideal for visualizing correlation matrices, helping to identify redundant or highly informative features.
    - The first heatmap provides a comprehensive view, while the second focuses on the selected features, validating the feature selection process.

8. Feature Selection Techniques Used

- **Methods Used**:
    - **Filter Method**: Mutual Information (MI) with mutual_info_classif to rank features by relevance to the label.

    - **Custom Correlation-Based Selection**: Iteratively selects features with high MI scores and low inter-feature correlation (< 0.9) to reduce redundancy.
- **Number of Features Selected**: 15
- **Selected Features**:
    - qrs_amplitude, rr_median, peak_count, p_wave_amplitude, range, psd_mean, p_wave_count, wavelet_variance, dominant_freq, skewness, t_wave_amplitude, t_wave_count, band_energy_ratio, zcr, kurtosis
- **Justification**:
    - MI ensures features are highly predictive of the label (normal vs. abnormal).
    - The correlation threshold (0.9) eliminates redundant features, improving model efficiency and reducing overfitting.
    - 15 features were chosen to balance model complexity and performance, as further reduction risked losing discriminative power.

9. Feature Transformation Techniques Used

- **Method**: Standardization (Z-score Normalization)
- **Description**:
    - Applied StandardScaler to the 15 selected features, transforming them to have zero mean and unit variance.
    - Formula: z = (x - mean) / std_dev
- **Purpose**:
    - Ensures all features are on the same scale, preventing features with larger ranges (e.g., wavelet_energy) from dominating distance-based algorithms (e.g., SVM).
    - Improves convergence and performance of machine learning models, especially those sensitive to feature scales (e.g., SVM, LDA).

10. Feature Reduction Techniques Used

- **Method**: Linear Discriminant Analysis (LDA)
- **Description**:
    - LDA reduces the 15 standardized features to 1 component, maximizing class separability for the binary classification task (normal vs. abnormal).
    - LDA projects the data onto a single dimension that best separates the two classes.
- **Explanation**:
    - LDA is ideal for binary classification, as it finds a linear combination of features that maximizes the ratio of between-class variance to within-class variance.
    - Reducing to one component simplifies the feature space, reduces computational cost, and mitigates overfitting while retaining discriminative information.
    - The explained variance ratio confirms LDA's effectiveness in capturing class differences.

11. Hypothesis Testing Methods Used

- **Method**: Independent Two-Sample t-test
- **Description**:
  - Performed t-tests for each of the 15 selected features to compare their distributions between classes (label 0 vs. 1).
  - Assumptions: Equal variances, normality (approximated for large samples).
- **Purpose**:
  - To identify features with statistically significant differences ($p < 0.05$) between normal and abnormal ECGs, validating their discriminative power.
  - Helps confirm that selected features are relevant for classification.

12. Models Employed

- **Models Used**:
  1. **Random Forest Classifier**:
     - sklearn.ensemble.RandomForestClassifier with n_estimators=100, random_state=42.
     - Ensemble method using decision trees, robust to noise and non-linear relationships.

```
Random Forest Training Accuracy: 0.9997
Random Forest model saved as '/content/drive/My Drive/Sem 6/DAV/random_forest_model.joblib'

Prediction for first row: 1.0, True Label: 1.0
Random Forest Test Set Accuracy: 0.9510
```

  2. **Support Vector Machine (SVM)**:
     - sklearn.svm.SVC with kernel='linear', random_state=42.
     - Linear kernel chosen due to the single LDA component, effective for linearly separable data.

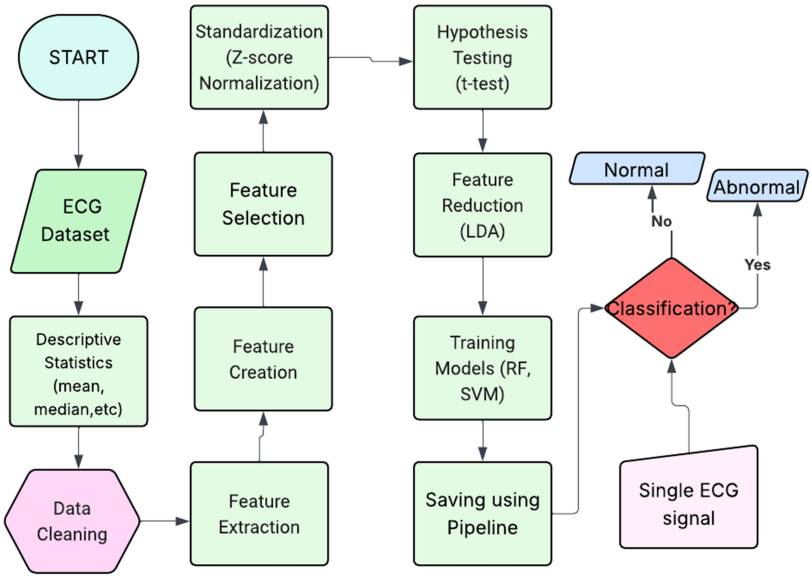
```
SVM Training Accuracy: 0.9687
SVM model saved as '/content/drive/My Drive/Sem 6/DAV/svm_model.joblib'

Prediction for first row: 1.0, True Label: 1.0
SVM Test Set Accuracy: 0.9700
```

13. Best Model Selection Criteria (Beyond Accuracy)

- **Metrics Considered**:
  - **F1-Score**: Balances precision and recall, critical for imbalanced datasets.
  - **Recall**: Measures the ability to detect abnormal ECGs (label 1), crucial for medical applications to minimize false negatives.
  - **False Negatives**: Number of missed abnormal cases, as false negatives are costly in diagnostics.
  - **AUC-ROC**: Measures overall discriminative ability across thresholds.

- **Evaluation**:
  - **5-Fold Cross-Validation Results**:
    - **Random Forest**:
      - Precision: 0.9533
      - Recall: 0.9510
      - F1-Score: 0.9522
      - AUC-ROC: 0.9428
      - False Negatives: 143
    - **SVM**:
      - Precision: 0.9669
      - Recall: 0.9801
      - F1-Score: 0.9735
      - AUC-ROC: 0.9665
      - False Negatives: 58
- **Best Model**: SVM
- **Reasons**:
  - **Primary Reason**: SVM has a higher F1-Score (0.9735 vs. 0.9522), indicating better balance of precision and recall.
  - **Secondary Reason**: SVM has higher recall (0.9801 vs. 0.9510) and fewer false negatives (58 vs. 143), critical for minimizing missed abnormalities in medical diagnostics.
  - **Additional Consideration**: SVM's higher AUC-ROC (0.9665 vs. 0.9428) suggests better overall discriminative power.
- **Justification**:
  - In medical applications, high recall and low false negatives are prioritized to ensure abnormal ECGs are detected, even at the cost of some false positives.

WORKFLOW:



Additional

- **Pipeline**: The entire preprocessing workflow (feature extraction, selection, standardization, LDA) is encapsulated in a sklearn.pipeline.Pipeline, saved to Google Drive, and tested on a new ECG signal, demonstrating reproducibility and deployability.