Phase_1

Submitted to:

Dr. Mahmoud Mounir

TA: Esraa Karam

Submitted by:

Nagy Ahmed Nagy	23p0365
Abdelrhman Mohammed Mahmoud Salah	23p0370
Ahmed Mosta Gomaa	23p0375
John George Mikhael	23P0266
Peter Maged Shokry	23p0192

Table Of Content

1. Introduction

1.1 Project Background and Significance

Diabetes mellitus represents one of the most pressing global health challenges of the 21st century, affecting approximately 537 million adults worldwide according to the International Diabetes Federation. As a chronic metabolic disorder characterized by elevated blood glucose levels, diabetes poses significant risks for severe complications including cardiovascular disease, kidney failure, neuropathy, and retinopathy. The early detection and accurate prediction of diabetes are therefore critical for implementing preventive measures, optimizing treatment strategies, and reducing healthcare costs.

This comprehensive data mining project addresses the crucial need for systematic analysis of clinical and laboratory parameters to identify patterns and risk factors associated with diabetes. By leveraging modern data science techniques, we aim to develop insights that can support healthcare professionals in early diagnosis and personalized treatment planning.

1.2 Project Objectives and Scope

The primary objectives of this project encompass multiple dimensions of healthcare data analytics:

Clinical Understanding:

- Investigate the complex relationships between various clinical parameters and diabetes status
- Identify the most significant biomarkers for diabetes prediction and classification
- Understand how different risk factors interact across patient demographics

Methodological Approach:

- Implement robust data preprocessing pipelines suitable for medical data
- Develop comprehensive outlier detection and treatment strategies
- Create informative visualizations that facilitate clinical interpretation
- Establish reproducible analytical workflows for healthcare datasets

Practical Applications:

- Support clinical decision-making through data-driven insights
- Enable risk stratification based on multiple clinical parameters
- Provide visualization tools for patient education and clinician training
- Contribute to the development of predictive models for diabetes screening

1.3 Dataset Context and Clinical Relevance

The dataset employed in this analysis comprises comprehensive medical records from 1000 patients, incorporating both demographic information and detailed laboratory measurements. This rich collection of clinical data enables multidimensional analysis of diabetes risk factors:

Demographic Factors: Age and gender distributions provide context for population-level risk assessment

Renal Function Markers: Urea and Creatinine (Cr) levels offer insights into kidney function, often compromised in diabetic patients

Glycemic Control: HbA1c (Glycated Hemoglobin) serves as a crucial indicator of long-term blood glucose management

Lipid Metabolism: Complete lipid profile including Cholesterol, Triglycerides, HDL, LDL, and VLDL captures cardiovascular risk factors

Body Composition: BMI (Body Mass Index) reflects adiposity, a well-established diabetes risk factor

The target variable CLASS categorizes patients into three distinct groups: Non-Diabetic, Predict-Diabetic (indicating pre-diabetes or high risk), and Diabetic, enabling nuanced analysis across the diabetes spectrum.

2. Dataset Overview

2.1 Original Dataset Structure

- **Size**: 1000 rows × 14 columns (including ID and Patient Number)
- **Target Variable**: CLASS (Diabetes classification)
- **Classes**: Non-Diabetic (N), Predict-Diabetic (P), Diabetic (Y)

2.2 Features Description

- **Demographic**: Gender, AGE
- **Blood Tests**: Urea, Cr (Creatinine), HbA1c (Glycated Hemoglobin)
- **Lipid Profile**: Chol (Cholesterol), TG (Triglycerides), HDL, LDL, VLDL
- **Anthropometric**: BMI (Body Mass Index)
- **Target**: CLASS (Diabetes status)

3. Data Preprocessing Techniques

3.1 Data Cleaning and Preparation

3.1.1 Duplicate Removal

Code:

```
duplicates = data[data.duplicated()]
print("🔍 Duplicate rows found:\n", duplicates)
db = data.drop(data.columns[[0, 1]], axis=1)
print(db)
```

Output:

No duplicate rows found in the dataset.

```
🔍 Duplicate rows found:
 Empty DataFrame
Columns: [ID, No_Pation, Gender, AGE, Urea, Cr, HbA1c, Chol, TG, HDL, LDL, VLDL, BMI, CLASS]
Index: []
    Gender  AGE  Urea  Cr  HbA1c  Chol   TG  HDL  LDL  VLDL   BMI CLASS
0        F   50   4.7  46    4.9   4.2  0.9  2.4  1.4   0.5  24.0     N
1        M   26   4.5  62    4.9   3.7  1.4  1.1  2.1   0.6  23.0     N
2        F   50   4.7  46    4.9   4.2  0.9  2.4  1.4   0.5  24.0     N
3        F   50   4.7  46    4.9   4.2  0.9  2.4  1.4   0.5  24.0     N
4        M   33   7.1  46    4.9   4.9  1.0  0.8  2.0   0.4  21.0     N
..     ...  ...   ...  ..    ...   ...  ...  ...  ...   ...   ...   ...
995      M   71  11.0  97    7.0   7.5  1.7  1.2  1.8   0.6  30.0     Y
996      M   31   3.0  60   12.3   4.1  2.2  0.7  2.4  15.4  37.2     Y
997      M   30   7.1  81    6.7   4.1  1.1  1.2  2.4   8.1  27.4     Y
998      M   38   5.8  59    6.7   5.3  2.0  1.6  2.9  14.0  40.5     Y
999      M   54   5.0  67    6.9   3.8  1.7  1.1  3.0   0.7  33.0     Y

[1000 rows x 12 columns]
```

3.1.2 Data Quality Assessment

Code:

```
print(f"Dataset shape: {db.shape}")
print(f"\nMissing values:\n{db.isnull().sum()}")
print(f"\nData types:\n{db.dtypes}")
```

Output:

- No missing values in any column
- Mixed data types: Object (categorical) and numerical

```
Dataset shape: (1000, 12)

Missing values:
Gender    0
AGE       0
Urea      0
Cr        0
HbA1c     0
Chol      0
TG        0
HDL       0
LDL       0
VLDL      0
BMI       0
CLASS     0
dtype: int64
```

```
Data types:
Gender     object
AGE         int64
Urea      float64
Cr          int64
HbA1c     float64
Chol      float64
TG        float64
...
VLDL      float64
BMI       float64
CLASS      object
dtype: object
```

3.1.3 Categorical Variable Encoding

Code:

```
db['Gender'] = db['Gender'].str.strip().str.upper()
db['CLASS'] = db['CLASS'].str.strip().str.upper()


db['Gender'] = db['Gender'].map({'M': 0, 'F': 1,'m':0})
db['CLASS'] = db['CLASS'].map({'N': 0, 'P': 1, 'Y': 2})
numerical_cols = ['AGE', 'Urea', 'Cr', 'HbA1c', 'Chol', 'TG', 'HDL', 'LDL', 'VLDL', 'BMI']
numerical_cols_exclude_Age = ['Urea', 'Cr', 'HbA1c', 'Chol', 'TG', 'HDL', 'LDL', 'VLDL', 'BMI']
print(db.dtypes)
print(db)
print(f"\nMissing values:\n{db.isnull().sum()}")
```

Output:

```
     Gender  AGE  Urea  Cr  HbA1c  Chol   TG  HDL  LDL  VLDL   BMI  CLASS
0         1   50   4.7  46    4.9   4.2  0.9  2.4  1.4   0.5  24.0      0
1         0   26   4.5  62    4.9   3.7  1.4  1.1  2.1   0.6  23.0      0
2         1   50   4.7  46    4.9   4.2  0.9  2.4  1.4   0.5  24.0      0
3         1   50   4.7  46    4.9   4.2  0.9  2.4  1.4   0.5  24.0      0
4         0   33   7.1  46    4.9   4.9  1.0  0.8  2.0   0.4  21.0      0
..      ...  ...   ...  ..    ...   ...  ...  ...  ...   ...   ...    ...
995       0   71  11.0  97    7.0   7.5  1.7  1.2  1.8   0.6  30.0      2
996       0   31   3.0  60   12.3   4.1  2.2  0.7  2.4  15.4  37.2      2
997       0   30   7.1  81    6.7   4.1  1.1  1.2  2.4   8.1  27.4      2
998       0   38   5.8  59    6.7   5.3  2.0  1.6  2.9  14.0  40.5      2
999       0   54   5.0  67    6.9   3.8  1.7  1.1  3.0   0.7  33.0      2

[1000 rows x 12 columns]
```

```
Gender      int64
AGE         int64
Urea      float64
Cr          int64
HbA1c     float64
Chol      float64
TG        float64
HDL       float64
LDL       float64
VLDL      float64
BMI       float64
CLASS       int64
dtype: object
```

```
Missing values:
Gender    0
AGE       0
Urea      0
Cr        0
HbA1c     0
Chol      0
TG        0
HDL       0
LDL       0
VLDL      0
BMI       0
CLASS     0
dtype: int64
```

4. Outlier Detection and Treatment

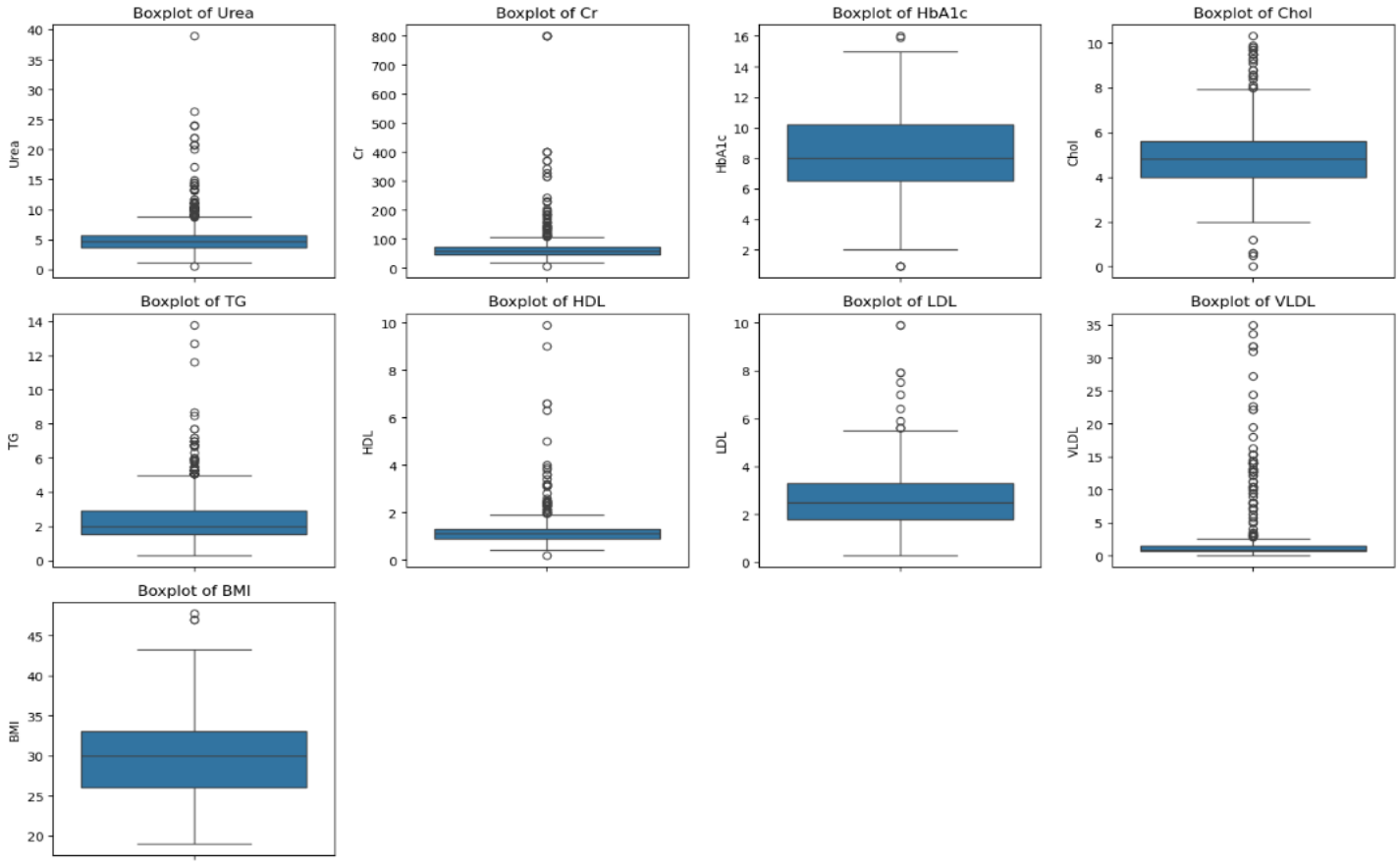
4.1 Statistical Approach for Outlier Detection

4.1.1 Interquartile Range (IQR) Method

Code:

```
def detect_outliers_iqr(data):
    Q1 = data.quantile(0.25)
    Q3 = data.quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    return data[(data < lower_bound) | (data > upper_bound)]

iqr_outliers = {}
for col in numerical_cols_exclude_Age:

    iqr_outliers[col] = detect_outliers_iqr(db[col])

plt.figure(figsize=(15, 10))
for i, col in enumerate(numerical_cols_exclude_Age, 1):
    plt.subplot(3, 4, i)
    sns.boxplot(y=db[col])
    plt.title(f'Boxplot of {col}')
plt.tight_layout()
plt.show()
```

Output:



4.1.2 Skewness and Kurtosis Analysis

Code:

```
for col in numerical_cols_exclude_Age:
    skewness = db[col].skew()
    kurtosis = db[col].kurtosis()
    print(f"{col}: Skewness={skewness:.2f}, Kurtosis={kurtosis:.2f}")
```

Output:

- Highly skewed features: Urea (4.30), Cr (8.47), TG (2.30), HDL (6.28), VLDL (5.35).
- Moderately skewed: LDL (1.15), Chol (0.62).
- Near normal: HbA1c (0.22), BMI (0.13).

```
Urea: Skewness=4.30, Kurtosis=30.43
Cr: Skewness=8.47, Kurtosis=91.71
HbA1c: Skewness=0.22, Kurtosis=-0.25
Chol: Skewness=0.62, Kurtosis=1.92
TG: Skewness=2.30, Kurtosis=10.26
HDL: Skewness=6.28, Kurtosis=62.63
LDL: Skewness=1.15, Kurtosis=4.28
VLDL: Skewness=5.35, Kurtosis=33.51
BMI: Skewness=0.13, Kurtosis=-0.30
```

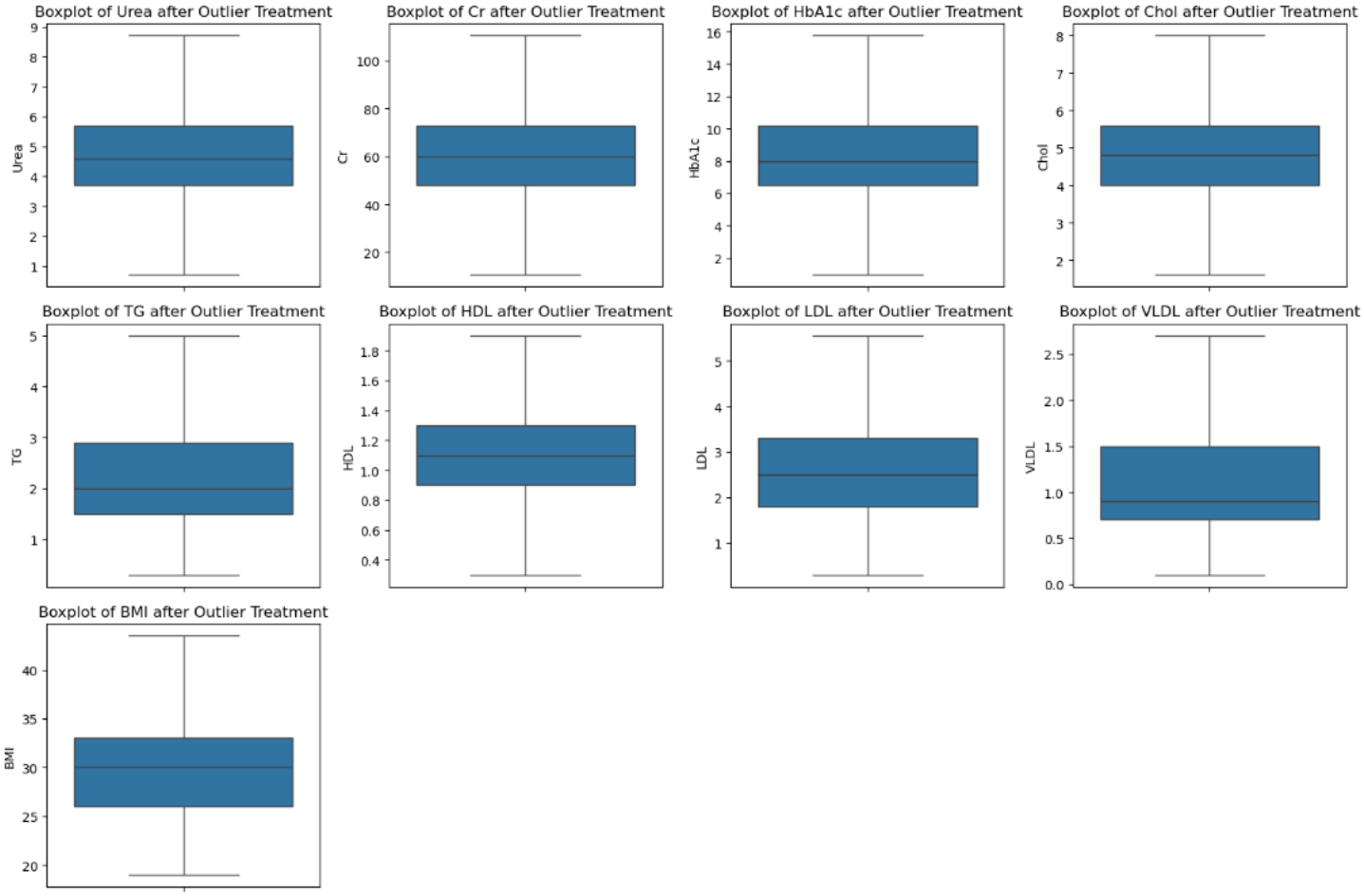
4.2 Outlier Treatment Strategy

4.2.1 Capping Method

Code:

```
def treat_outliers_iqr(db, column):
    Q1 = db[column].quantile(0.25)
    Q3 = db[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    # Cap outliers
    db[column] = np.where(db[column] < lower_bound, lower_bound, db[column])
    db[column] = np.where(db[column] > upper_bound, upper_bound, db[column])
    return db

for col in numerical_cols_exclude_Age:
    db = treat_outliers_iqr(db, col)
plt.figure(figsize=(15, 10))
for i, col in enumerate(numerical_cols_exclude_Age, 1):
    plt.subplot(3, 4, i)
    sns.boxplot(y=db[col])
    plt.title(f'Boxplot of {col} after Outlier Treatment')
plt.tight_layout()
plt.show()
```

Output:



5. Data Visualization and Correlation Analysis

5.1 Distribution Analysis

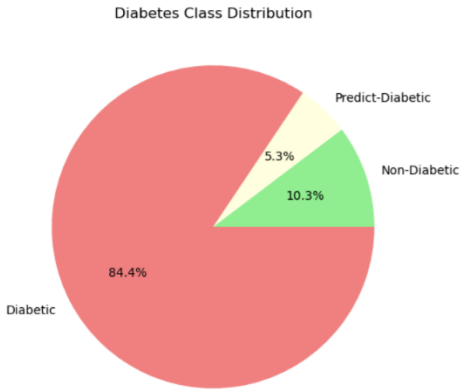
5.1.1 Class Distribution

Code:

```
plt.figure(figsize=(8, 6))
class_counts = db['CLASS'].value_counts().sort_index()
plt.pie(class_counts.values, labels=['Non-Diabetic', 'Predict-Diabetic', 'Diabetic'],
        autopct='%1.1f%%', colors=['lightgreen', 'lightyellow', 'lightcoral'])
plt.title('Diabetes Class Distribution')
plt.show()
```

Output:

- Diabetic: 84.4%
- Non-Diabetic: 10.3%
- Predict-Diabetic: 5.3%



5.2 Correlation Analysis

5.2.1 Overall Correlation Matrix

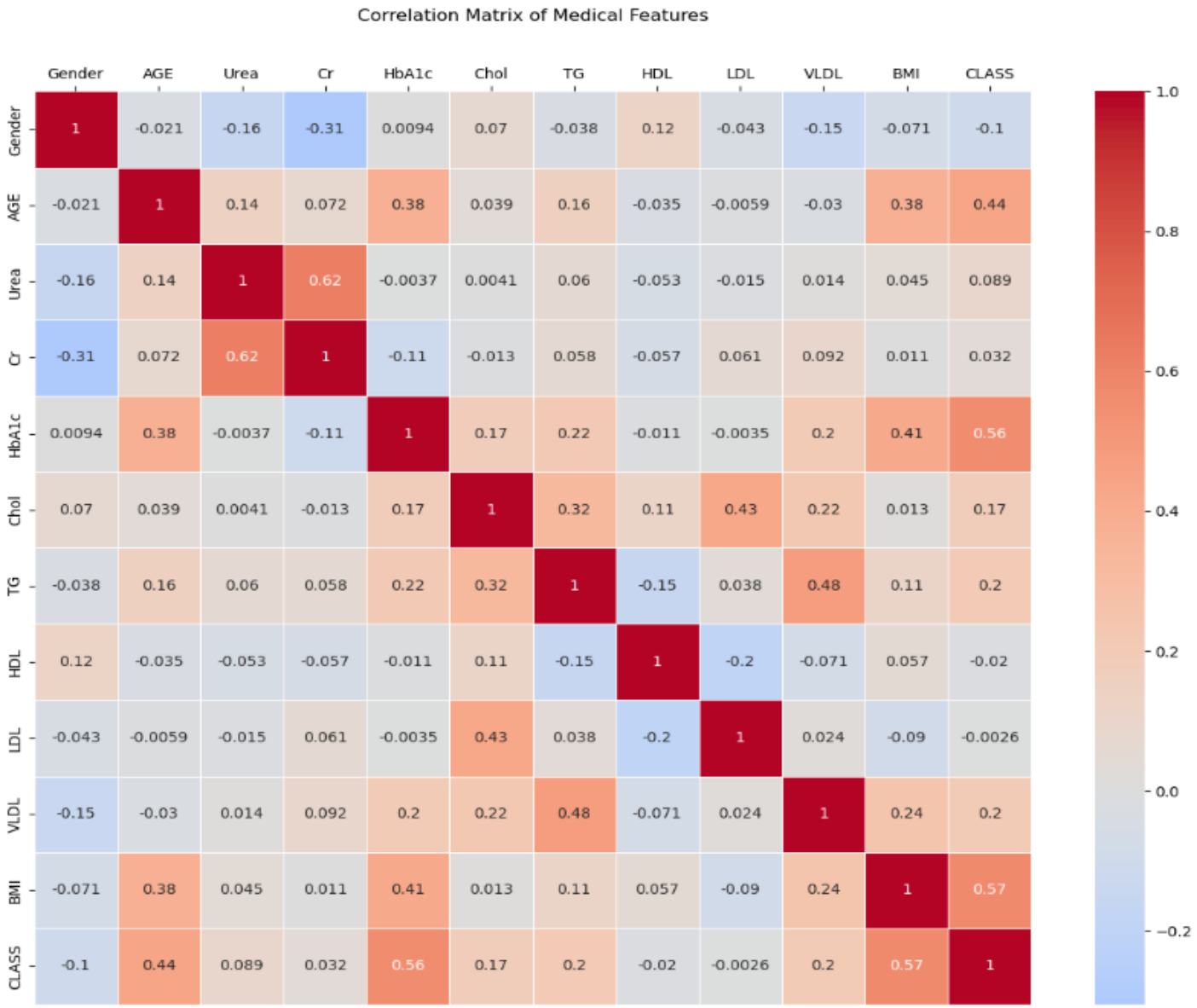
Code:

```
plt.figure(figsize=(12, 10))
correlation_matrix = db.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', center=0,square=True, linewidths=0.5)
plt.gca().xaxis.tick_top()
plt.title('Correlation Matrix of Medical Features', pad=50)
plt.tight_layout()
plt.show()
```

Output:

Key Correlations with CLASS:

- Moderate positive: BMI (0.57), AGE (0.44), HbA1c (0.56)
- Weak correlations: VLDL (0.20), TG (0.20), Gender (0.10), Urea (0.09)



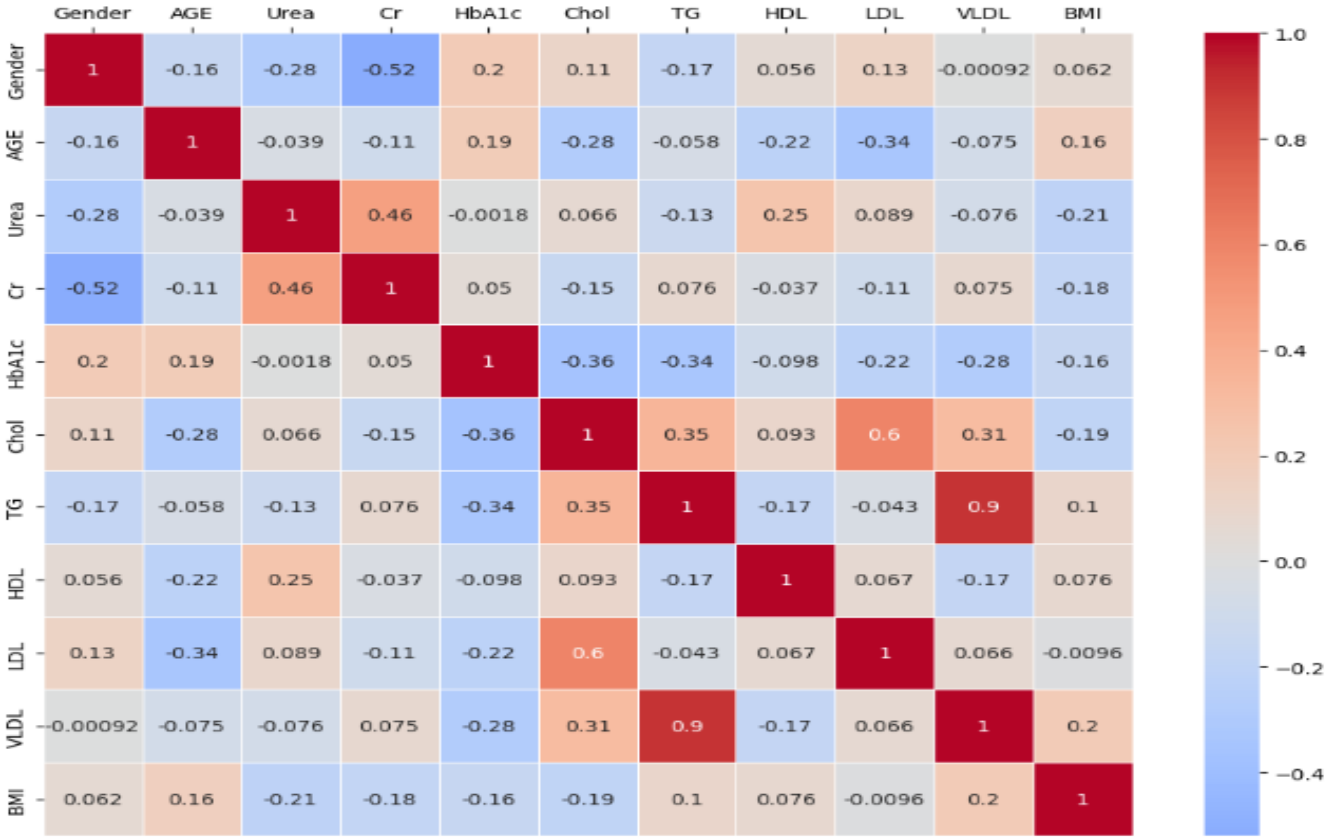
5.2.2 Class-wise Correlation Analysis

Code:

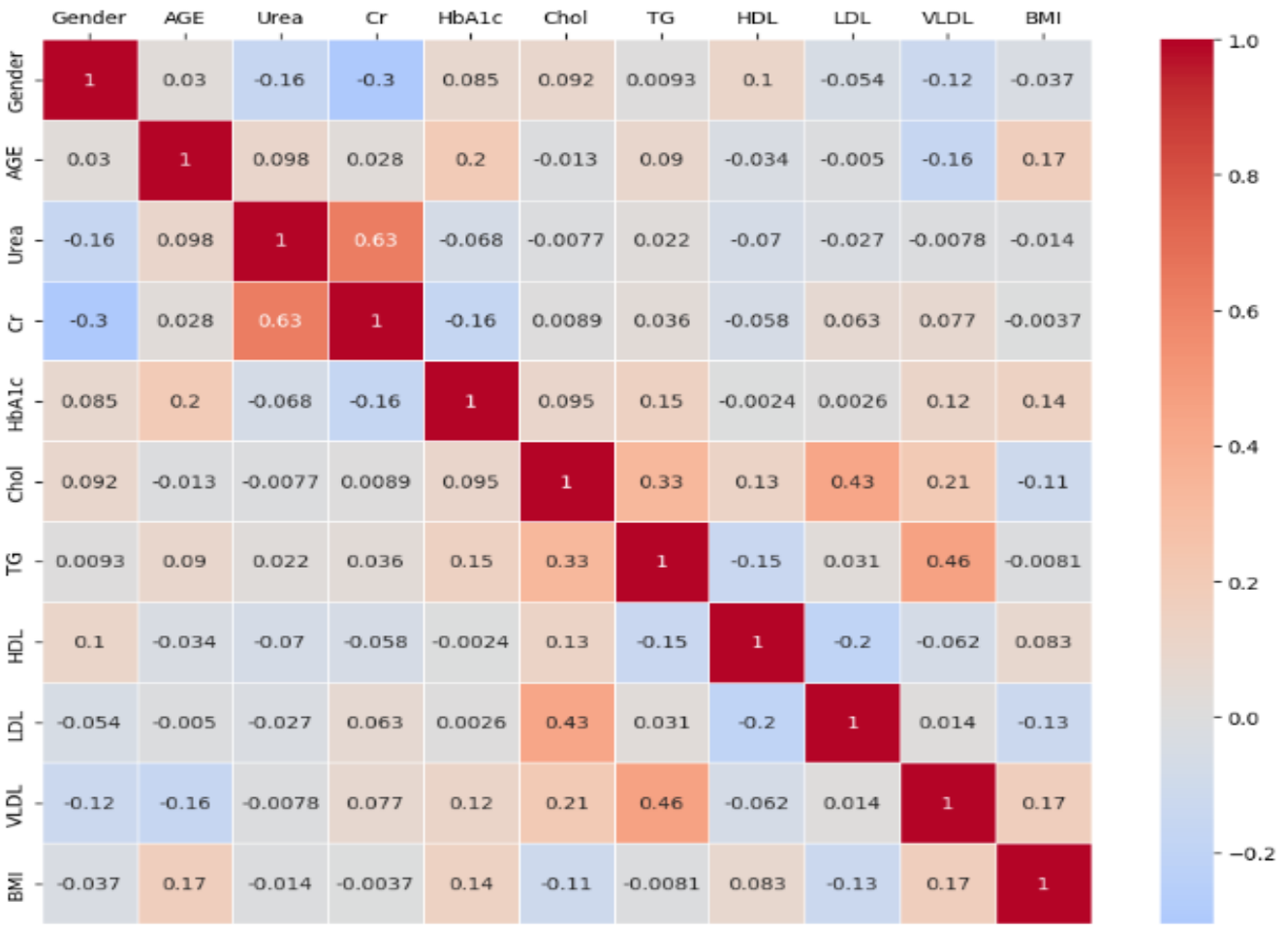
```
for cls in db['CLASS'].unique():
    subset = db[db['CLASS'] == cls]

    # Drop CLASS because it has no meaning inside a single class
    subset_numeric = subset.drop(columns=['CLASS'])

    plt.figure(figsize=(10, 8))
    corr_matrix = subset_numeric.corr()

    sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', center=0, square=True, linewidths=0.5)
    plt.gca().xaxis.tick_top()
    plt.title(f'Correlation Matrix for Class {cls}\n', pad=20)
    plt.tight_layout()
    plt.show()
```

Output:

Correlation Matrix for Class 1



Correlation Matrix for Class 2



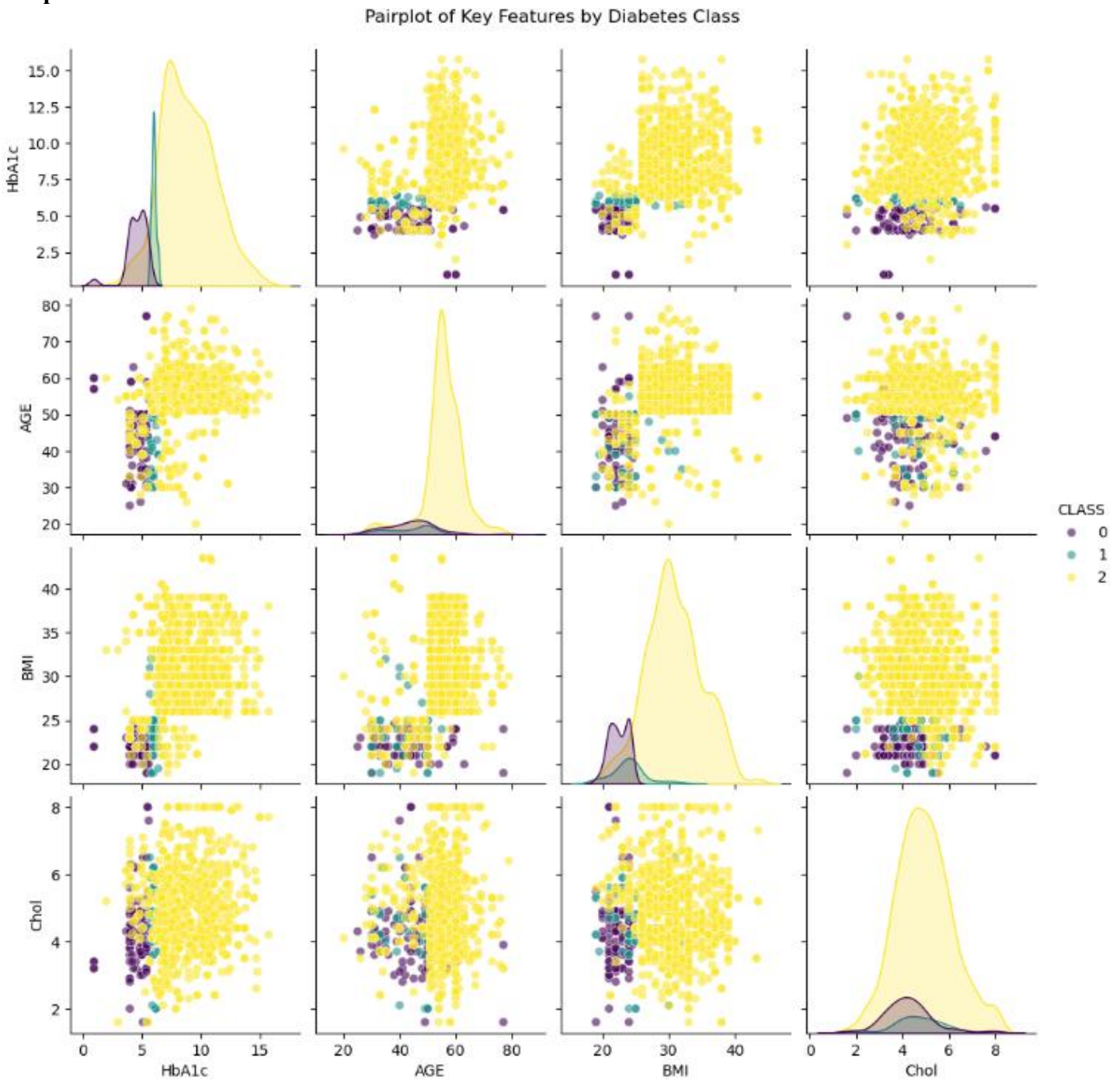
5.3 Feature Relationship Visualization

5.3.1 Bivariate Analysis

Code:

```
key_vars = ['HbA1c', 'AGE', 'BMI', 'Chol', 'CLASS']
sns.pairplot(db[key_vars], hue='CLASS', palette='viridis', plot_kws={'alpha': 0.6})
plt.suptitle('Pairplot of Key Features by Diabetes Class', y=1.02)
plt.show()
```

Output:

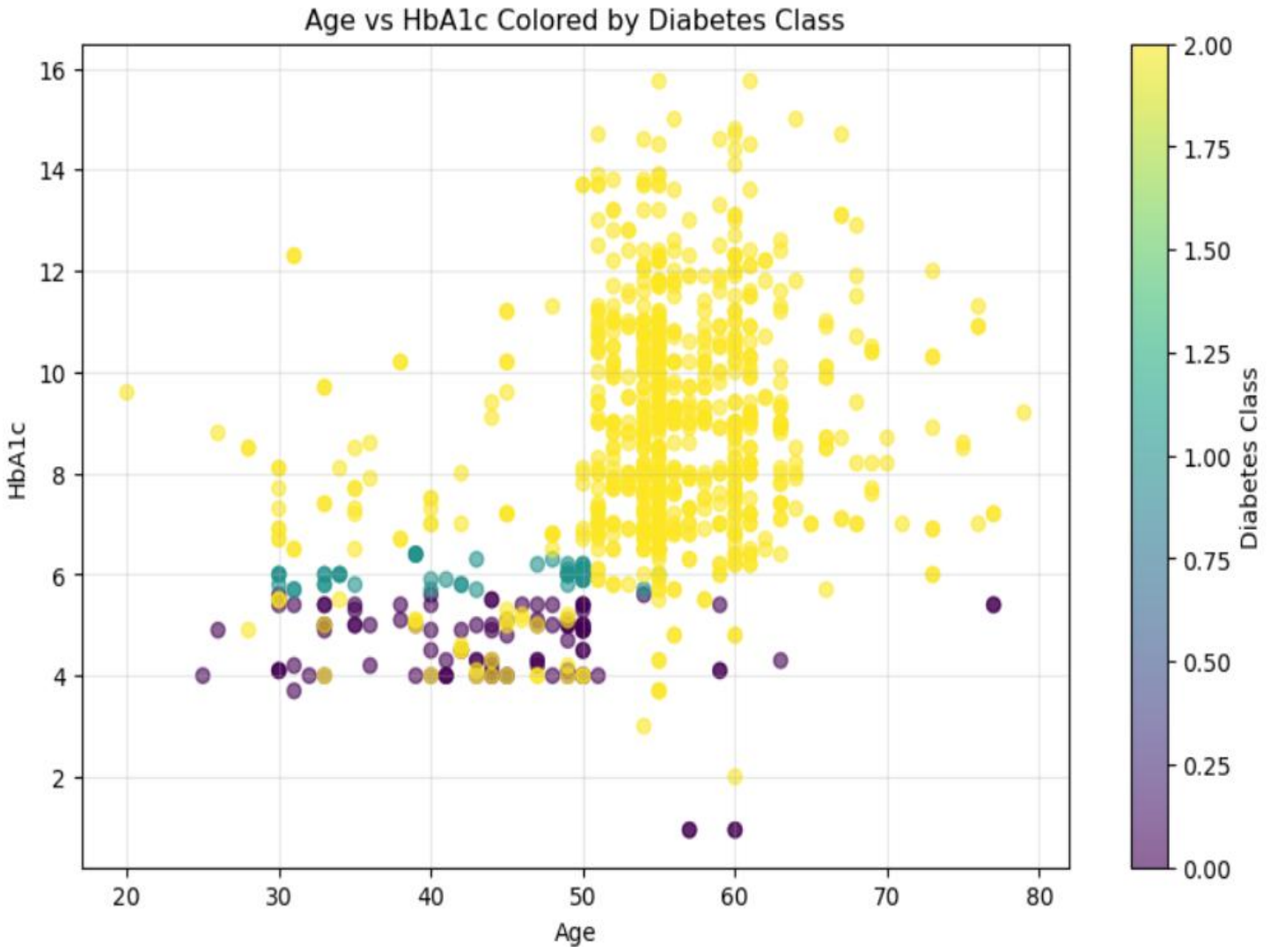


Age vs HbA1c Relationship:

Code:

```
plt.figure(figsize=(10, 6))
scatter = plt.scatter(db['AGE'], db['HbA1c'], c=db['CLASS'],
                      cmap='viridis', alpha=0.6)
plt.colorbar(scatter, label='Diabetes Class')
plt.xlabel('Age')
plt.ylabel('HbA1c')
plt.title('Age vs HbA1c Colored by Diabetes Class')
plt.grid(True, alpha=0.3)
plt.show()
```

Output:

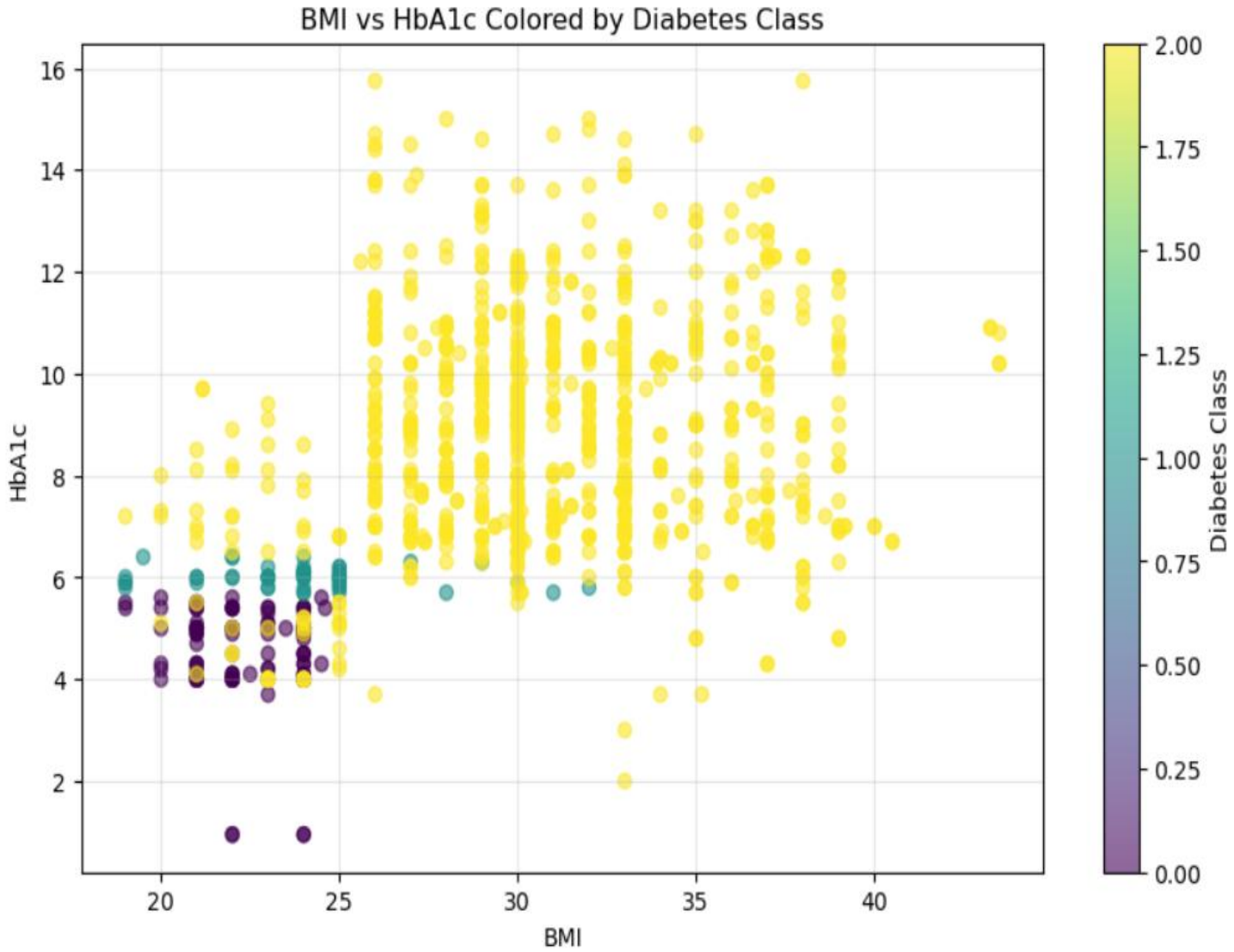


BMI vs HbA1c Relationship:

Code:

```
plt.figure(figsize=(10, 6))
scatter = plt.scatter(db['BMI'], db['HbA1c'], c=db['CLASS'], cmap='viridis', alpha=0.6)
plt.colorbar(scatter, label='Diabetes Class')
plt.xlabel('BMI')
plt.ylabel('HbA1c')
plt.title('BMI vs HbA1c Colored by Diabetes Class')
plt.grid(True, alpha=0.3)
plt.show()
```

Output:

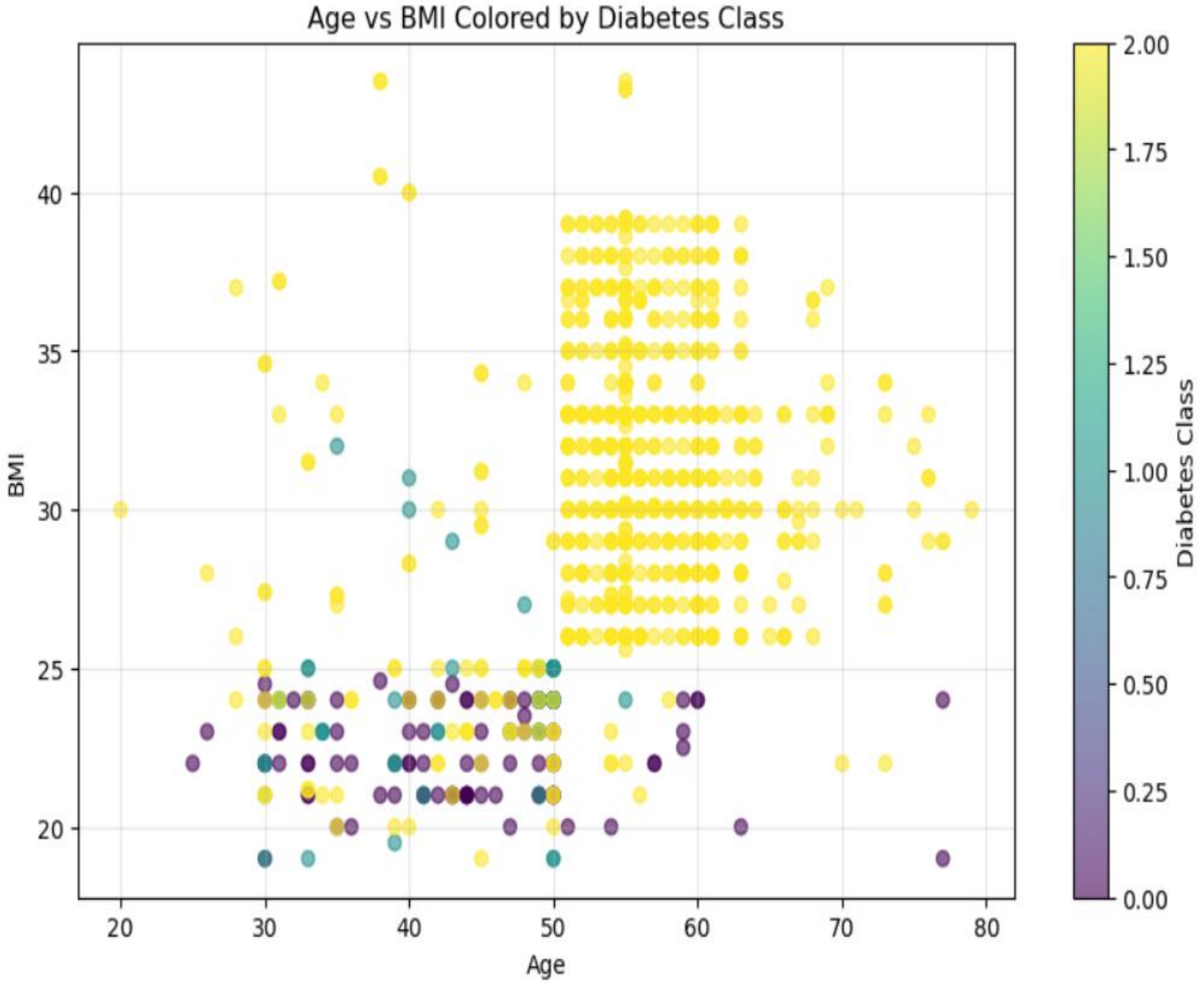


Age and BMI Relationship:

Code:

```
plt.figure(figsize=(10, 6))
scatter = plt.scatter(db['AGE'], db['BMI'], c=db['CLASS'], cmap='viridis', alpha=0.6)
plt.colorbar(scatter, label='Diabetes Class')
plt.xlabel('Age')
plt.ylabel('BMI')
plt.title('Age vs BMI Colored by Diabetes Class')
plt.grid(True, alpha=0.3)
plt.show()
```

Output:



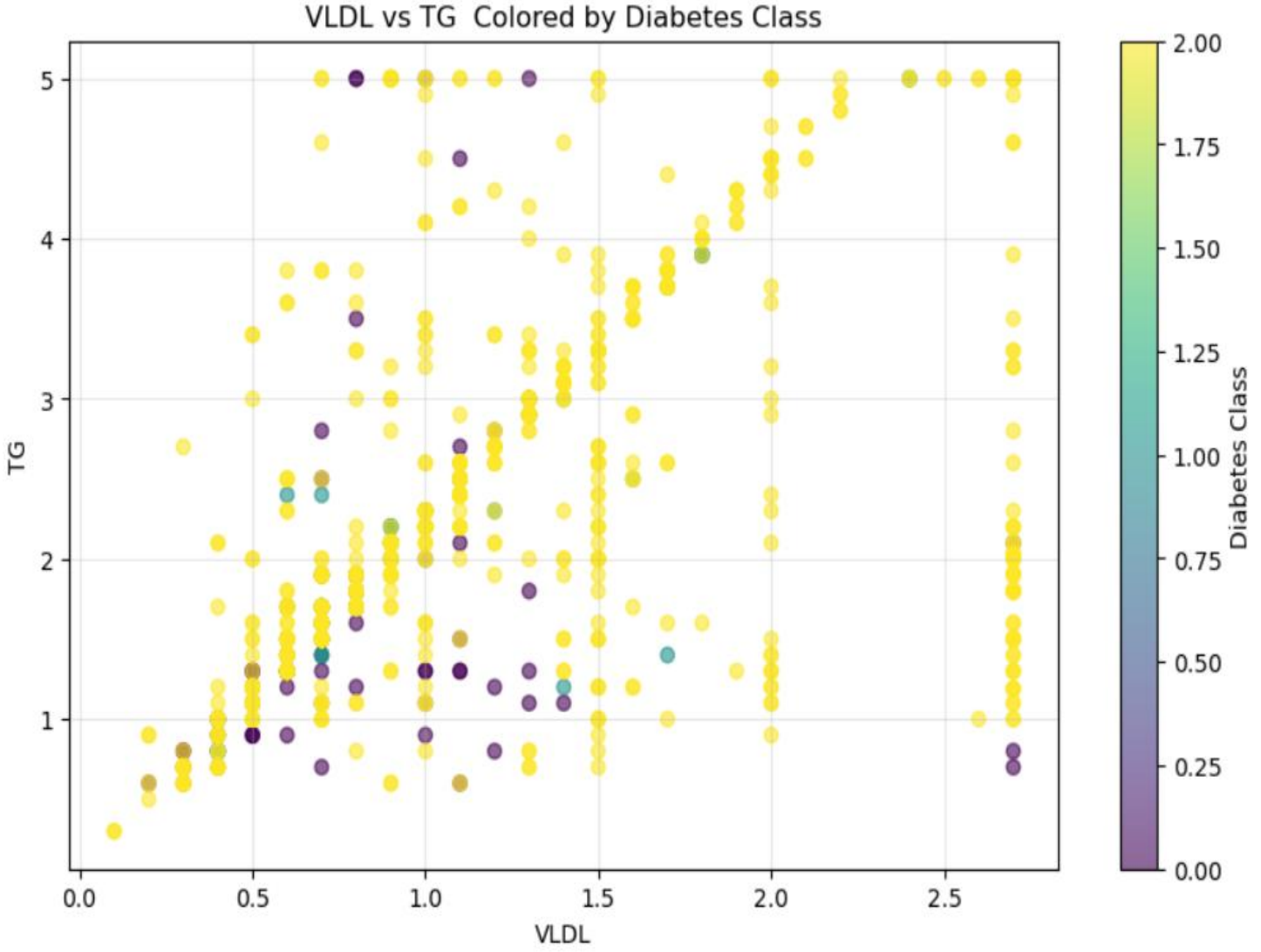
5.3.2 Lipid Profile Relationships

VLDL vs TG:

Code:

```
plt.figure(figsize=(10, 6))
scatter = plt.scatter(db['VLDL'], db['TG'], c=db['CLASS'], cmap='viridis', alpha=0.6)
plt.colorbar(scatter, label='Diabetes Class')
plt.xlabel('VLDL')
plt.ylabel('TG')
plt.title('VLDL vs TG  Colored by Diabetes Class')
plt.grid(True, alpha=0.3)
plt.show()
```

Output:

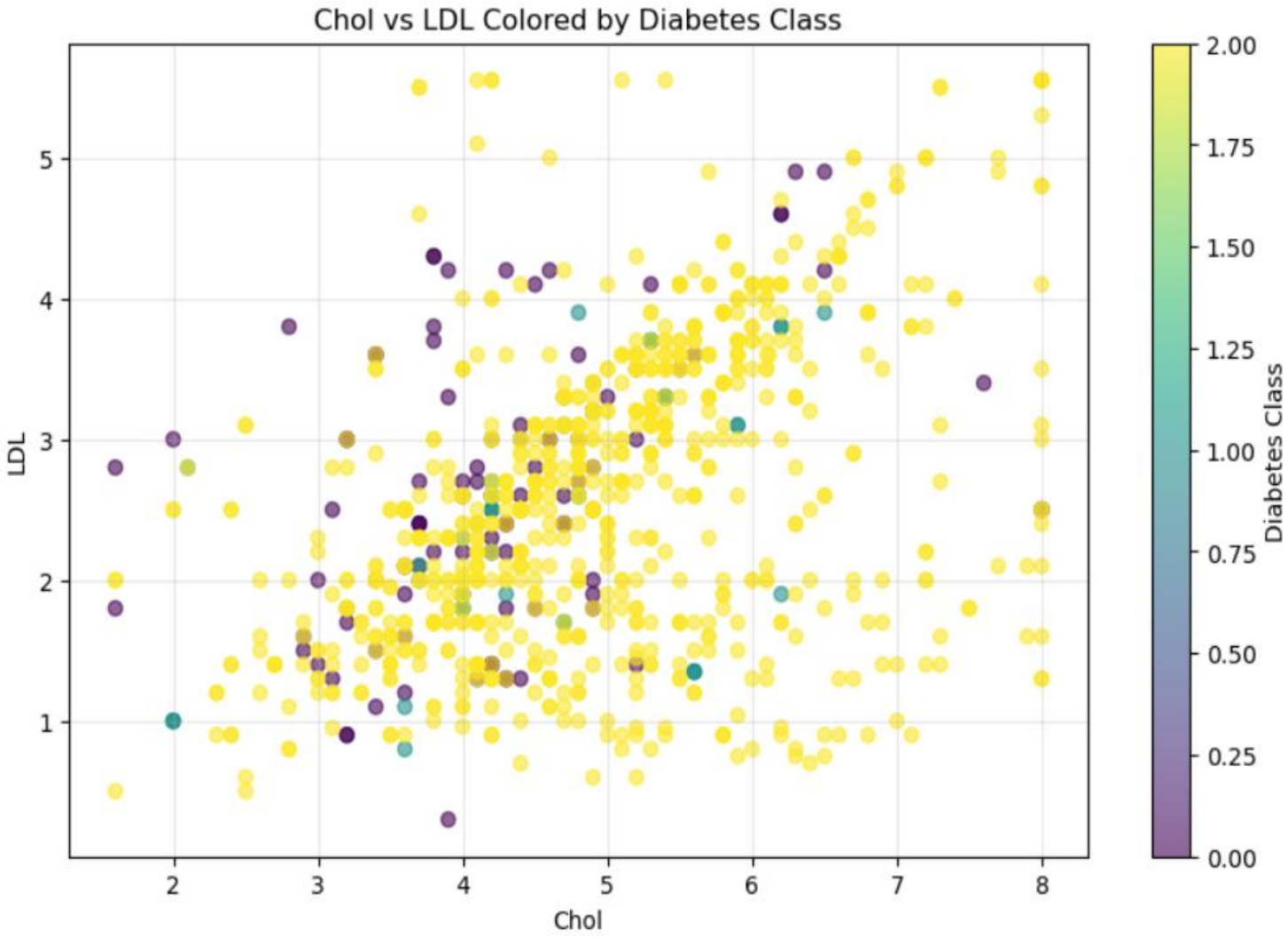


Chol vs LDL:

Code:

```
plt.figure(figsize=(10, 6))
scatter = plt.scatter(db['Chol'], db['LDL'], c=db['CLASS'], cmap='viridis', alpha=0.6)
plt.colorbar(scatter, label='Diabetes Class')
plt.xlabel('Chol')
plt.ylabel('LDL')
plt.title('Chol vs LDL Colored by Diabetes Class')
plt.grid(True, alpha=0.3)
plt.show()
```

Output:



5.4 Distribution by Demographic Factors

5.4.1 BMI Distribution by Gender and Class

Code:

```
plt.figure(figsize=(12, 6))
sns.violinplot(x='CLASS', y='BMI', hue='Gender', data=db, split=True,palette='pastel')
plt.title('BMI Distribution by Diabetes Class and Gender')
plt.xticks([0, 1, 2], ['Non-Diabetic', 'Predict-Diabetic', 'Diabetic'])
plt.legend(title='Gender', labels=['Male', 'Female'])
plt.show()
```

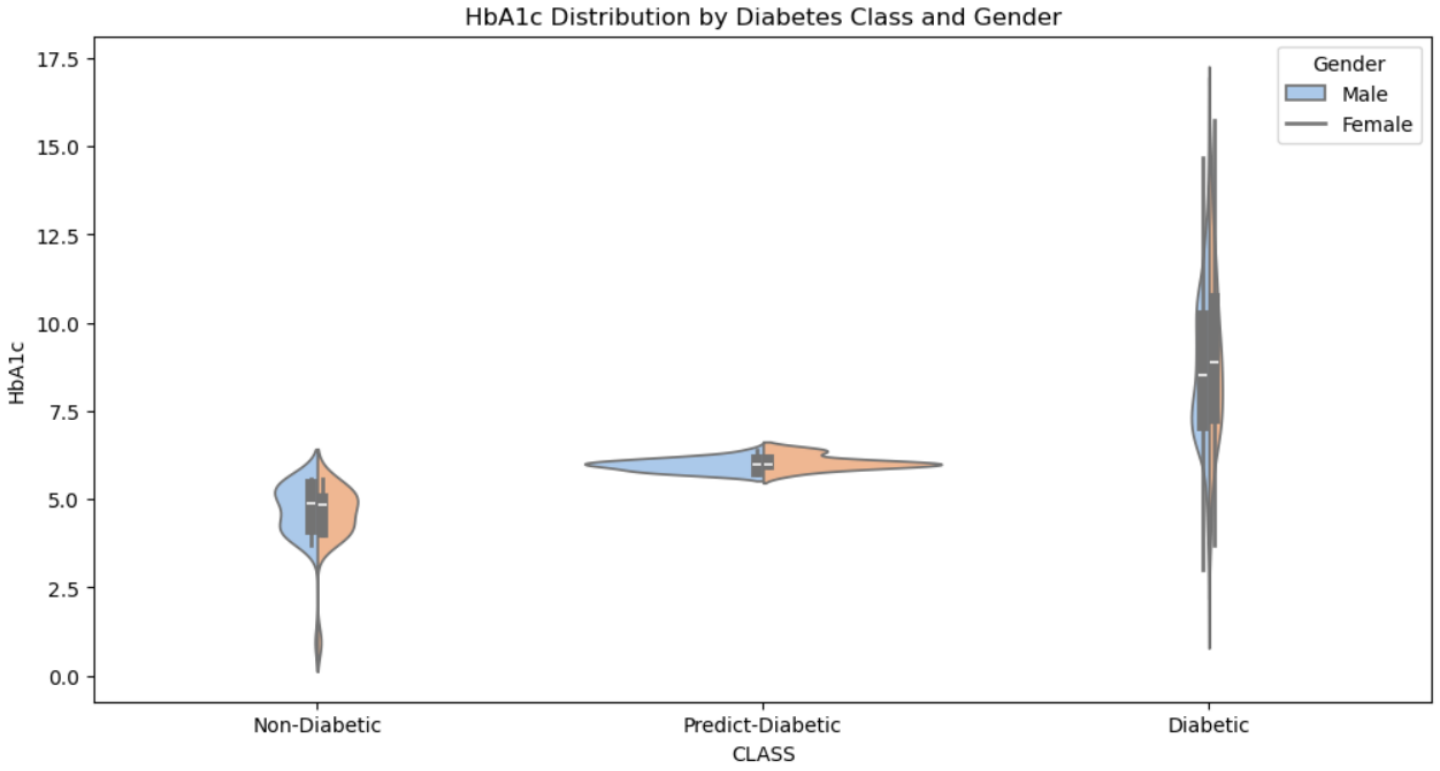
Output:

5.4.2 HbA1c Distribution by Gender and Class

Code:

```
plt.figure(figsize=(12, 6))
sns.violinplot(x='CLASS', y='HbA1c', hue='Gender', data=db, split=True,palette='pastel')
plt.title('HbA1c Distribution by Diabetes Class and Gender')
plt.xticks([0, 1, 2], ['Non-Diabetic', 'Predict-Diabetic', 'Diabetic'])
plt.legend(title='Gender', labels=['Male', 'Female'])
plt.show()
```

Output:



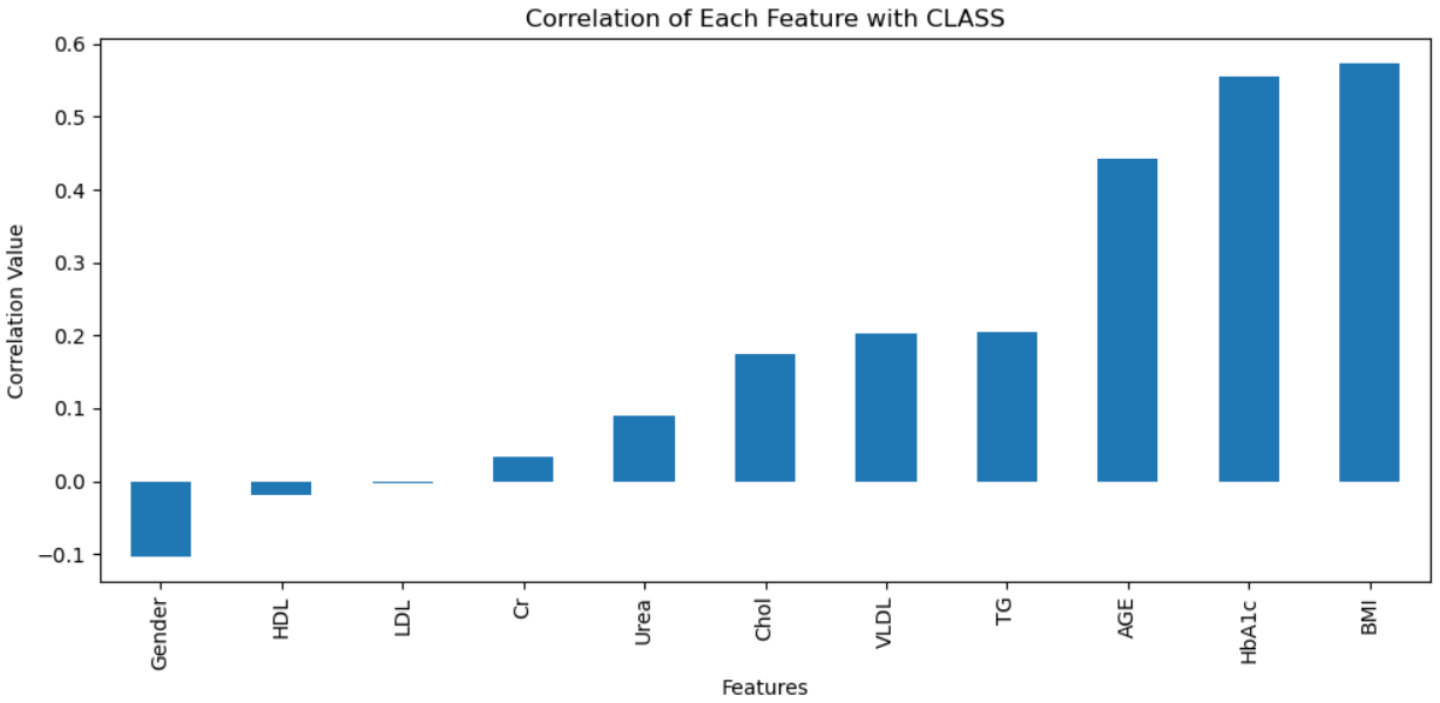
5.5 Feature Importance Visualization

5.5.1 Correlation with Target Variable

Code:

```
corr_with_class = db.corr()['CLASS'].drop('CLASS').sort_values()

plt.figure(figsize=(10, 5))
corr_with_class.plot(kind='bar')
plt.title('Correlation of Each Feature with CLASS')
plt.ylabel('Correlation Value')
plt.xlabel('Features')
plt.tight_layout()
plt.show()
```

Output:



6. Key Findings and Insights

6.1 Feature Impact on Diabetes Classification

6.1.1 Primary Influential Parameters

HbA1c (0.56)

- Direct relationship with diabetes classification status
- Values increase progressively across disease categories
- Serves as clinical benchmark for glycemic monitoring

BMI (0.57)

- Highest individual parameter relationship with diabetes status
- Distribution patterns show clear separation between classes
- Progressive increase from non-diabetic to diabetic categories

Age (0.44)

- Demonstrates expected pattern of increased prevalence with advancing years
- Younger populations are predominantly in non-diabetic category
- Elderly populations show higher representation in diabetic class

6.1.2 Secondary Influential Parameters

Lipid Profile Components

- VLDL and Triglycerides both show 0.20 relationship values
- Reflect known diabetic dyslipidemia patterns
- Interdependent relationship confirmed through correlation analysis

Renal Function Markers

- Urea and Creatinine demonstrate emerging patterns in diabetic class
- Individual values show limited direct impact
- Class-specific analysis reveals progressive changes

6.2 Population Distribution Patterns

6.2.1 Age-Based Stratification

Younger Cohort (≤35 years)

- 5.3% representation in predict-diabetic category
- 10.3% presence in diabetic classification
- Lower overall prevalence but notable early-onset cases

Older Cohort (≥50 years)

- 84.4% representation in diabetic classification
- Dominant demographic in advanced disease categories
- Supports age as significant demographic factor

6.2.2 Gender Distribution Analysis

Overall Impact (0.10)

- Limited direct relationship with diabetes classification
- Challenges assumptions about gender-based predisposition
- Requires deeper analysis of manifestation patterns

Parameter-Specific Variations

- BMI distributions show subtle gender differences within classes
- HbA1c levels demonstrate gender-based variability patterns
- Suggests different physiological manifestations rather than prevalence differences

6.3 Metabolic Parameter Interrelationships

6.3.1 Lipid Metabolism Connections

VLDL and Triglycerides (0.48)

- Expected physiological relationship confirmed
- Reflects known metabolic pathway connections
- Consistent across patient classifications

Cholesterol and LDL (0.43)

- Demonstrates expected compositional relationship
- LDL represents primary cholesterol carrier
- Pattern consistency validates data quality

HDL Isolation Pattern

- Limited correlations with other lipid parameters
- Supports independent metabolic role
- Consistent with reverse cholesterol transport function

6.3.2 Metabolic Syndrome Indicators

Parameter Clustering Evidence

- Central adiposity (BMI) with dyslipidemia (TG, VLDL)
- Glycemic control (HbA1c) with age progression
- Supports clinical metabolic syndrome concept
- Multiple abnormalities often present simultaneously

6.4 Disease Progression Patterns

6.4.1 Non-Diabetic Class Characteristics

Parameter Stability

- Narrow value distributions within normal ranges
- Limited internal parameter correlations
- Suggests independent metabolic regulation

Demographic Profile

- Younger age distribution dominance
- Stable metabolic parameter relationships
- Represents baseline metabolic state

6.4.2 Predict-Diabetic Transition Phase

Intermediate Patterns

- Parameter values between non-diabetic and diabetic ranges
- Emerging correlation structures
- Represents metabolic transition state

Clinical Implications

- Critical window for intervention
- Early metabolic changes detectable
- Opportunity for preventive measures

6.4.3 Diabetic Class Characteristics

Metabolic Dysregulation Evidence

- Multiple significant parameter correlations
- Broader value distributions across parameters
- Indicates systemic metabolic changes

Advanced Disease Patterns

- Clear demographic clustering in older populations
- Established metabolic relationship patterns
- Represents chronic disease state

6.5 Data Quality Assessment

6.5.1 Distribution Characteristics

Right-Skewed Parameters

- Urea (4.30), Cr (8.47), HDL (6.28)
- Majority values within normal ranges
- Subset with significantly elevated values
- Suggests for complication development in advanced cases

Near-Normal Distributions

- HbA1c (0.22) and BMI (0.13)
- Affect broader patient populations
- Reliable screening and monitoring parameters

6.5.2 Analytical Methodology Impact

Outlier Treatment Outcomes

- Maintained clinical relevance of parameter values
- Reduced extreme value influence on analysis
- Preserved majority of original data integrity
- Enhanced statistical analysis robustness

7. Conclusion

7.1 Analytical Achievements

7.1.1 Data Processing Framework

Quality Assurance Implementation

- Comprehensive data completeness evaluation
- Systematic consistency verification
- Standardized processing methodology

Outlier Management Approach

- Clinically informed detection strategies
- Appropriate treatment methodologies
- Reproducible analytical workflows

7.1.2 Clinical Insight Generation

Risk Parameter Identification

- Clear hierarchy of influential factors
- Confirmation of established clinical knowledge
- Quantitative support for screening priorities

Metabolic Pattern Recognition

- Complex parameter interrelationships revealed
- Disease progression patterns identified
- Population-specific characteristics documented

7.2 Methodological Contributions

7.2.1 Visualization Framework

Clinical Interpretation Support

- Intuitive graphical representations
- Multidimensional analysis capabilities
- Pattern recognition enhancement

Statistical Validation

- Quantitative parameter relationship confirmation
- Epidemiological pattern verification
- Clinical guideline support evidence

7.3 Practical Applications

7.3.1 Healthcare Strategy Implications

Screening Program Optimization

- Age-specific approach recommendations
- Comprehensive parameter assessment protocols
- Early intervention targeting strategies

Personalized Medicine Support

- Individualized risk assessment frameworks
- Parameter-specific monitoring approaches
- Tailored intervention methodologies

7.4 Limitations and Development Opportunities

7.4.1 Current Constraints

Data Characteristics

- Single timepoint measurement limitation
- Limited clinical context information
- Sample size considerations for rare patterns

7.4.2 Future Directions

Analytical Advancements

- Predictive model development potential
- Longitudinal analysis opportunities
- External validation requirements

Clinical Integration

- Decision support tool development
- Healthcare workflow incorporation
- Educational resource creation

7.5 Implementation Recommendations

Clinical Practice Enhancement

- Diabetes risk factors understanding improvement
- Screening program effectiveness optimization
- Patient education resource development

Public Health Impact

- Data-driven prevention strategy formulation
- Resource allocation optimization
- Population health improvement planning