Hetavi Gheewala - KNN-multi classification

1. **In-depth analysis of the results obtained.**

```
Results by Distance Metric:
Metric: manhattan, Optimal K: 20, Accuracy: 79.84%
Metric: euclidean, Optimal K: 20, Accuracy: 79.84%
Metric: minkowski, Optimal K: 20, Accuracy: 79.84%

Optimal Metric: manhattan, Optimal K: 20, Accuracy: 79.84%
Model Accuracy with Best K (20): 79.84%

Confusion Matrix:
                  P-No Risk  P-Low Risk  P-Moderate Risk  P-High Risk
A-No Risk              7306           0                0          613
A-Low Risk              279        2080                0          785
A-Moderate Risk           0           0                0           91
A-High Risk            2886         898                0        12608

Classification Report:
                precision    recall  f1-score   support

      No Risk       0.70      0.92      0.79      7919
     Low Risk       0.70      0.66      0.68      3144
Moderate Risk       0.00      0.00      0.00        91
    High Risk       0.89      0.77      0.83     16392

     accuracy                           0.80     27546
    macro avg       0.57      0.59      0.58     27546
 weighted avg       0.81      0.80      0.80     27546
```

- **Analysis by Risk Categories**

a. **No Risk Category**

- The KNN model performed strongly in predicting the No Risk category, achieving a high recall of 92%. Indicating that the majority of samples classified as No Risk were correctly identified. However, the precision of 70% suggests some misclassifications, where samples from other categories, especially High Risk (309 cases), were wrongly classified as No Risk.

b. Low Risk Category

- Predictions for the *Low Risk* category were consistent, with balanced precision and recall at 69%. However, 366 samples were misclassified as *High Risk*, which reduced the overall reliability of predictions for this

category. The balanced F1-score highlights the model's moderate effectiveness in handling this class.

c. **Moderate Risk Category**
    - The model completely failed to predict the *Moderate Risk* category, as evidenced by zero precision, recall, and F1-score. This is likely due to the very small number of samples (46), which rendered the category underrepresented and challenging for KNN to detect. This highlights the model's limitation in handling imbalanced datasets, particularly for underrepresented classes.

d. **High Risk Category**
    - The model excelled in predicting the *High Risk* category, achieving a precision of 90% and a high F1-score of 0.82. However, the recall of 76% suggests that 24% of *High Risk* samples were misclassified, primarily as *No Risk* (1457 cases) or *Low Risk* (480 cases). The high precision shows that most samples predicted as *High Risk* were correct, but the model's sensitivity for this category could be improved.

- **Insights on Confusion Matrix**
    - A large number of High Risk instances were incorrectly classified as No Risk or Low Risk, which could be important in real-world scenarios.
    - The Moderate Risk category was completely overlooked, highlighting the need for improved handling of unusual classes.
    - Cross-category misclassification was the most common across neighboring risk levels (Low Risk and High Risk), indicating that feature distributions could overlap.

- **Macro and Weighted Averages**
    - The low macro-average scores reflect the model's difficulty in handling underrepresented classes like Moderate Risk.
    - The weighted averages are higher because they give more importance to well-represented classes (No Risk and High Risk), where the model performed better.

- **Strengths of the Model**
    - High precision for well-represented classes (No Risk and High Risk), indicating that the model can properly identify data in these categories.

- K=16 achieved near-optimal performance by balancing the trade-off between underfitting and overfitting.

2. **Evaluation Criteria**

Well-articulated computational challenges

1. **Determining the Optimal Value of KKK:**
   Selecting the best KKK was critical to balancing model bias and variance. A grid search was implemented, after testing values of different K to identify K=16 as the optimal choice based on validation accuracy. Smaller K values caused overfitting by being overly sensitive to noise, while larger K values led to underfitting due to overly smoothed decision boundaries.
2. **Multitasking Classification Design:**
   To adapt KNN for multitasking classification, we had to handle different risk levels at the same time. We created a clear structure to include all categories and ensure predictions stayed accurate. This meant carefully preparing the data so the algorithm could properly understand the category boundaries.
3. **Computational Complexity:**
   For a dataset with more than 50,000 samples, KNN was slow since it had to compute distances for every training point. To address this, we enhanced the data structures and used a quicker method for calculating distances. This made the program run quicker while keeping the accuracy the same.

4. **Hyperparameter Optimization:**

   Along with KNN, we carefully selected the distance metric (Euclidean distance) and weighting strategies. These choices played a key role in helping the model accurately differentiate between risk categories while maintaining its ability to work well on new data.

Application of realistic approaches to practical datasets

1. **Feature Engineering and Scaling:**
   The preprocessing included normalizing features, which is important for KNN because it uses distance calculations. By scaling all features to the same range, no single feature had too much influence on the distance metric.
2. **Robust Validation Strategy:**
   We used K-fold cross-validation to test the model, giving a realistic measure of how well it would perform on new data. This method helped ensure that

performance metrics like accuracy, precision, and recall were not overly optimistic due to quirks in the dataset.

3. **Model Applicability to Multitasking Scenarios:**
   The KNN multitasking model was designed with real-world challenges in mind, such as managing multiple outcomes while staying easy to understand. Predictions for each risk category were simple to trace back to the nearest neighbors, making the model practical and reliable for decision-making.

Comprehensive discussion of experimental results, including interpretation.

1. **Optimal K Selection:**
   After testing different K values, we found that K=16 worked best. This choice reduced classification errors and created smooth decision boundaries. It also showed the model's ability to handle new data well, as seen in consistently high validation scores.
2. **Confusion Matrix Insights:**
   The confusion matrix gave a clear picture of how predictions were distributed across categories. KNN performed especially well in recognizing distinct categories, proving that the distance metric effectively captured feature relationships.
3. **Metric Interpretation:**
   - **Precision and Recall:** High precision in key categories showed that the model made accurate predictions with few false positives. Strong recall indicated its ability to detect most relevant samples for each class.
   - **F1-Scores:** Balanced F1-scores across major categories highlighted the model's ability to manage the trade-off between precision and recall effectively.

3. **Additional Consideration:**

- Obstacles overcome during the project.

a. **Handling High Dimensionality:**
   KNN can struggle in high-dimensional spaces due to the "curse of dimensionality." To address this, we used feature selection and normalization to ensure the distance metric stayed effective and meaningful features weren't overshadowed.

b. **Parameter Tuning:**
   Finding the best K value through grid search was computationally demanding. This was managed by narrowing the search space with informed initial guesses based on theory and preliminary results.

- Any noteworthy issues or challenges to share.

a. **Scalability Issues:**
   Even with optimizations, KNN's O(n) complexity for predictions remained a limitation for large datasets. Larger applications might require approximate nearest neighbor search methods.
b. **Sensitivity to Distance Metrics:**
   While Euclidean distance worked well for this dataset, experiments with metrics like Manhattan or Minkowski distances produced mixed results. This underlined how important the choice of distance metric is for specific datasets.
c. **Adapting KNN for Multitasking:**
   Extending KNN from single-task to multitask classification was a challenge. It required designing a robust system to handle interactions between different risk levels in the feature space while ensuring consistent predictions.

- Future research directions or plans.

a. **Dynamic K Selection**
   Future work could explore dynamic methods for selecting K that adjust the number of neighbors based on local data density. Sparse regions could use a larger K for more reliable predictions, while dense areas might benefit from a smaller K to better capture fine-grained patterns.
b. **Hybrid Models**
   Combining KNN with other machine learning techniques like decision trees, ensemble methods, or neural networks could enhance overall performance. For example, KNN could preprocess or refine features before a more sophisticated model generates final predictions.