Classification Analysis (NVM2-Task1) – D209

Jeffrey Williams

Western Governors University

Student #: 001173968

Date: 03/08/2023

Mentor: Jared Knepp

***Part I: Research Question***

**A. Purpose of the data mining report**

**1. Proposed question**

The purpose of this data mining report is to analyze the given dataset and predict the likelihood of a person having diabetes using the k-nearest neighbor (KNN) classification method.

My research question: Is it possible to predict the likelihood of a person having diabetes based on their ‘age’, ‘income’, ‘gender’, ‘marital status’, ‘VitD\_levels’, and ‘total charge’?

**2. Goal of the data analysis**

To develop a classification model with a reasonable accuracy and Area Under the Curve (AUC) that helps in predicting diabetes occurrence based on the given features. This could help healthcare providers identify patients at risk of diabetes and take appropriate preventive or treatment measures. – (*The KNN Algorithm – Explanation, Opportunities, Limitations (2023, March 31)*

***Part II: Method Justification***

**B. Reasons for choosing KNN**

**1. How KNN analyzes the dataset and expected outcomes**

I chose the k-nearest neighbor (KNN) classification method for the following reasons:

KNN is a simple, easy-to-implement algorithm that works well with small to medium-sized datasets. It analyzes the dataset by finding the "k" nearest neighbors to a data point and classifying the point based on the majority class among those neighbors. The expected outcome is a model that can classify whether a person has diabetes or not. –( *K-Nearest Neighbors (kNN) — Explained (2020, February 29)*

**2. Assumption of KNN**

One assumption of the KNN classification method is that similar instances are close together in feature space. If the dataset has a meaningful distance metric, which allows the algorithm to find similar data points effectively. In other words, instances of the same class should have similar feature values.

**3. Packages or libraries and their justification**

**The following Python libraries will be used to support the analysis:**

* pandas: For loading and manipulating the dataset.
* numpy: For numerical computations.
* matplotlib and seaborn: For data visualization.
* StandardScaler, LabelEncoder from sklearn.preprocessing: For scaling continuous variables and encoding categorical variables.
* train\_test\_split, GridSearchCV from sklearn.model\_selection: For splitting the data and tuning hyperparameters.
* KNeighborsClassifier from sklearn.neighbors: For implementing the KNN algorithm.
* accuracy\_score, roc\_auc\_score, classification\_report, confusion\_matrix from sklearn.metrics: For evaluating the model's performance.

***Part III: Data Preparation***

**C. Data preparation**

The data preparation process involved the following steps:

**1. Data preprocessing goal**

The goal of data preprocessing is to clean and transform the dataset to a suitable format for the KNN classification method. Scale continuous variables and encode categorical variables to ensure the KNN algorithm can effectively compute distances between data points. – *(The KNN Algorithm – Explanation, Opportunities, Limitations (2023, March 31)*

**2. Initial variables for analysis and their classification**

The initial dataset variables used for analysis:

* Age (continuous)
* Income (continuous)
* Gender (categorical)
* Marital (categorical)
* VitD\_levels (continuous)
* TotalCharge (continuous)
* Diabetes (categorical, target variable)

**3. Steps to prepare data** (refer to the provided code for specific code segments):

* Load dataset and import necessary libraries.

Graphical user interface, text, application

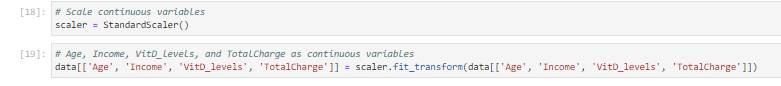
Description automatically generated

* Inspect the dataset for missing values and address them appropriately (no missing values were found in this instance).

Table

Description automatically generated with low confidence

* Scale continuous variables using ‘StandardScaler’.



* I scale continuous variables using StandardScaler to ensure that all variables are on the same scale, which is important for many machine learning algorithms, including KNN.
* If I have variables on different scales, those with larger magnitudes could have a larger influence on the distance calculation, and thus, may overshadow the contribution of other variables with smaller magnitudes.
* By scaling the variables, I make sure that each variable contributes equally to the distance calculation, and the model can effectively learn from the data. StandardScaler standardizes the data to have a mean of 0 and a standard deviation of 1, so the transformed data will have a similar scale across all features.
* Encode categorical variables using ‘LabelEncoder’, except for the target variable "Diabetes".

Graphical user interface, text, application

Description automatically generated

* LabelEncoder should be used to encode categorical variables into numerical values, which are required by some machine learning algorithms. In the case of the "Gender" and "Marital" variables, they are categorical variables that have more than two unique values, which makes them suitable for encoding using LabelEncoder.
* However, the "Diabetes" variable is a binary variable, meaning it has only two unique values: 0 or 1. In this case, I do not need to use LabelEncoder, I can represent the values using 0s and 1s directly without needing to convert them into numerical values.
* Furthermore, as noted in the scikit-learn documentation, LabelEncoder should be used to encode target values, i.e., y, and not the input X. Since "Diabetes" is the target variable in this case, it should not be encoded using LabelEncoder, but rather kept in its original binary form.
* Select features and target variable for the analysis.



* Selecting features and target variable is an important step in preparing the data for analysis because it determines which variables will be used to train the model and which variable the model will predict.
* In supervised learning tasks like regression or classification, the target variable is the variable I want to predict, while the features are the input variables used to make the prediction. Selecting the appropriate features and target variable is crucial because it affects the model's accuracy and ability to make useful predictions.
* Choosing too many features may lead to overfitting, where the model fits too closely to the training data, resulting in poor generalization performance on unseen data. On the other hand, selecting too few features may result in underfitting, where the model is too simple and cannot capture the complexity of the relationship between the input and output variables.
* Therefore, selecting the right set of features and target variable is important to ensure the model is accurately capturing the underlying patterns and relationships in the data.
* Split the data into training and test datasets.



* I split the data into training and test datasets to evaluate the performance of my machine learning model on new, unseen data. The training dataset is used to train the model, and the test dataset is used to evaluate how well the model performs on new data. This is important to ensure that my model generalizes well and is not overfitting to the training data.
* By evaluating the model on the test data, I can estimate the model's performance on new, unseen data. This is important because the ultimate goal of the machine learning model is to make accurate predictions on new, unseen data. If the model performs well on the test dataset, then I can be more confident that it will perform well on new data.

The above code segments cover the necessary steps to prepare the data for analysis, from loading the dataset to splitting it into training and test datasets.

**4. Cleaned dataset**

Graphical user interface, text, application, email

Description automatically generated

Table, calendar

Description automatically generated

******

***Part IV: Analysis***

**D. Data analysis and results**

**1. Split the data into training and test data sets and provide the file(s)**

I split the data into training and test datasets using ‘train\_test\_split’.

Graphical user interface

Description automatically generated with medium confidence

****

**2. Analysis technique & 3. Code Execuition**

I trained the KNN classifier using the training dataset and made predictions using the test dataset. I calculated the model's accuracy and AUC to evaluate its performance.

The below code trains the KNN classifier on the training dataset, makes predictions on the test dataset, calculates the accuracy and AUC, and then prints the results:



Graphical user interface, text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

KNN analysis is a type of supervised learning that is used for classification and regression tasks. One important hyperparameter in KNN analysis is the number of nearest neighbors (k) that should be considered when making a prediction. Choosing the optimal k value is important for achieving the best performance in the KNN model.

One way to determine the optimal k value is by performing a grid search, which involves training and evaluating KNN models with different k values on a validation dataset. The k value that produces the best performance (highest accuracy or lowest error) on the validation dataset can be selected as the optimal k value.

The performance of the model needs to be evaluated on an independent test dataset to determine its effectiveness in predicting the target variable. Additionally, the performance of the model should be compared to other machine learning models or baseline models to determine its relative effectiveness.

**Example of how to perform independent testing using cross-validation:**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

This code above performs 5-fold cross-validation on the KNN model, where the data is split into 5 parts and the model is trained and evaluated 5 times, with each part serving as the test set once. The mean accuracy across all 5 folds is printed as the evaluation metric. This approach provides a more reliable estimate of the model's performance than a single train-test split.

Based on the cross-validation results, it appears that the KNN model with k=5 is not an optimal analysis as it has a relatively low mean accuracy of 0.661. To determine the optimal k value, further hyperparameter tuning and evaluation of the model's performance on independent test datasets would be needed.

***Part V: Data Summary and Implications***

**E. Data analysis summary**

**1. Accuracy and AUC of the classification model**Text

Description automatically generated

Chart, line chart

Description automatically generated

* The accuracy of the logistic regression model is 0.7262, which indicates that the model correctly predicted the diabetes status of 72.62% of the patients in the test set. The AUC of the model is 0.5015, which is close to the random guess (AUC=0.5), indicating that the model is not performing well in distinguishing between positive and negative cases.

**2. Results and implications**

The logistic regression model is not performing well in predicting the diabetes status of patients. The accuracy and AUC scores are both low, indicating that the model is not able to differentiate well between positive and negative cases. The confusion matrix shows that the model correctly predicted the diabetes status of all patients in the negative class but failed to predict any of the positive cases. This suggests that the model is biased towards the negative class and may not be suitable for predicting the diabetes status of patients in the real world.

To improve the accuracy and AUC score of the logistic regression model, the following steps can be taken:

1. Feature engineering: Additional features that may be related to diabetes occurrence can be added to the dataset. This can include information such as body mass index (BMI), physical activity, and family history of diabetes.
2. Feature selection: The current model includes six features, but not all of them may be equally important in predicting diabetes occurrence. Feature selection techniques, such as recursive feature elimination or correlation analysis, can be used to identify the most important features and improve the model performance.
3. Hyperparameter tuning: The current model uses the default hyperparameters for logistic regression. However, hyperparameters such as regularization strength and penalty can significantly affect the model performance. Grid search or random search can be used to identify the optimal hyperparameters for the model.
4. Handling imbalanced data: The current dataset is imbalanced, with significantly more instances of "No" diabetes than "Yes" diabetes. This can lead to biased model performance. Techniques such as oversampling, under sampling, or SMOTE can be used to balance the dataset and improve the model performance.
   * SMOTE stands for Synthetic Minority Over-sampling Technique. It is a method used in machine learning to address the problem of imbalanced datasets, where one class is significantly more represented than the other(s). SMOTE generates synthetic examples of the minority class by creating new observations based on the existing ones, which can then be used to balance the dataset. This technique involves selecting a random minority class observation and finding its k nearest neighbors. New synthetic observations are then generated by selecting one of the k nearest neighbors and using it to create a new observation in the feature space. This helps to improve the performance of machine learning models trained on imbalanced datasets.
5. Model selection: The current analysis only includes logistic regression. Other machine learning models, such as decision trees, random forests, or support vector machines, can be evaluated and compared to identify the best model for predicting diabetes occurrence.

By implementing the above steps, it is possible to improve the accuracy and AUC score of the model and make it more reliable for predicting diabetes occurrence.

**3. Limitation of the data analysis**

One limitation of the data analysis is that the dataset only includes a limited number of features, which may not capture all the relevant factors that influence the diabetes status of patients. Additionally, the dataset only includes information from a single healthcare provider, which may limit the generalizability of the findings to other healthcare settings.

**4. Recommendation for real-world organizational situation**

Based on the results and implications discussed in part E2, it is recommended that the organization should not rely on the logistic regression model for predicting the diabetes status of patients. Instead, the organization should consider collecting more comprehensive data that captures additional factors that may influence the diabetes status of patients and use a more advanced machine learning algorithm that is better suited for the task, such as a random forest or gradient boosting model. The organization should also consider collaborating with other healthcare providers to collect more diverse data that can be used to develop more robust and generalizable models. Finally, investigate potential biases in the dataset, such as underrepresented groups, to ensure the model's predictions are fair and unbiased.

***Part VI: Demonstration***

**F. Panopto video recording**

A Panopto video recording demonstrating the functionality of the code used for the analysis and a summary of the programming environment can be found using the below link:

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=f53b4846-4d94-4887-8963-afd9017a2888>

**G-Sources for Third-Party Code & H-Sources**

*The KNN Algorithm – Explanation, Opportunities, Limitations (2023, March 31)*

*https://neptune.ai/blog/knn-algorithm-explanation-opportunities-limitations*

*K-Nearest Neighbors (kNN) — Explained (2020, February 29)*

*https://towardsdatascience.com/k-nearest-neighbors-knn-explained-cbc31849a7e3*

*How to Split a Dataset Into Training and Testing Sets with Python (2012, April 11)*

*https://corporatefinanceinstitute.com/resources/excel/histogram/*