

K Nearest Neighbors (KNN) Documentation & Log

Data Analytics & Algorithms Log

- o **Algorithm Used:** K Nearest Neighbors (KNN)
- o **Dataset Used:** Breast Cancer Wisconsin dataset
- o **Framework:** CRISP-DM

Original Workbook Link: [Notebook](#)

Phases	Changes Made	Reason for the change	Duration	Difficulty level (1-10)
Dataset selection	Changed the dataset from “Classified dataset” to “Breast Cancer Wisconsin dataset”	In the original notebook, classified dataset worked out really well, just wanted to explore a new dataset in my area of interest i.e., Breast Cancer	30 min	6
Algorithm Version	Used two types i.e., KNN with uniform weights and KNN with distance weights	I know KNN with uniform weights will work because of the size of the dataset, but wanted to explore different versions of KNN to select the best one	40 min	7
Further Enhancements	GridSearchCV	Instead of randomly selecting the n and p values, I selected GridSearchCV to select optimal number of neighbors (n_neighbors) and minkowski parameter ‘p’ to choose manhattan or euclidean distance	3 hours	8
	Error Rate plot	Got n =3, p=2 from the above method, just wanted to verify the conclusion we made from the above method		

	ROC Curve Implementation	Through error rate plot, successfully cross validated that $n = 3$, $p = 2$ are the excellent choices, I wanted to check the false positives, so I implemented ROC Curve		
Conclusions		<p>Different versions of algorithms are performed exceptionally, this strengthens my analytical skills to select the best one.</p> <p>Very low misclassifications, because of the data balance. By this dataset and prediction results I understood how important the balance in a dataset is.</p> <p>Finally, I can conclude that the model performed exceptionally better in all cases (recall, F1 Score. etc.) than the model in the original notebook.</p>	40 min	6

1. Business Understanding

- **Original Notebook Context**

The original notebook showed how the K-Nearest Neighbors (KNN) algorithm worked on a sample dataset called the “Classified Data.”

- **New Objective**

I decided to go with the **Breast Cancer Wisconsin dataset** available from scikit-learn to test KNN on a **real-life medical classification issue** (benign vs. malignant tumors).

- **Rationale**

- **Useful Context:** The breast cancer dataset is a benchmark in medical diagnostics and provides a useful context.
- **Increased Relevance:** Using a clinically relevant dataset, helps me explain better how KNN deals with real world data variability.

2. Data Understanding

- **Data Source**

- **Breast Cancer Wisconsin dataset -- scikit-learn.**

- **569 samples** and **30 numeric features**.
- Target variable: **0** (malignant) and **1** (benign).
- **Adjustments**
 - Instead of importing “Classified Data” file, I used *load_breast_cancer()* from scikit-learn.
 - I checked the dataset by printing the shape of the dataset, distribution of target, and showing the first five rows.
- **Practical Impact**
 - The **feature distributions** change from the original in the new dataset.
 - This means KNN will require **meticulous scaling** and parameter tuning to perform effectively.

3. Data Preparation

- **Scaling**
 - I applied *StandardScaler* to all 30 features to so that one feature does not dominate the distance metric.
- **Train-Test Split**
 - I divided the data into **70%-30% training** and **testing sets**, using **stratification** to keep the ratio of benign vs malignant proportion the same.
- **Adjustments**
 - I used a breast cancer dataset appropriate pipeline to replace original reading and scaling steps.
 - Stratified split keeps class balance intact.
- **Practical Impact**
 - **Stratifying** makes the model evaluation more robust.
 - **Scaling** is important for distance-based algorithms like KNN, avoiding the results being distorted by large-scale features.

4. Modeling

- **Algorithm Selection**
 - I continued to use **KNN** but added **two weighting schemes**:
 - **Uniform Weights** (what each neighbor contributes is equal).
 - **Distance Weights** (more influence on closer neighbors).
- **Hyperparameter Tuning**
 - I used **GridSearchCV** to perform a systematic search on:
 - *n_neighbors* between **1 to 30**
 - *p* = {1, 2}, where **Manhattan** (p=1) or **Euclidean** (p=2) distance
- **Adjustments**

- I used **GridSearchCV** to find optimal parameters for both weighting schemes, instead of a simple for-loop over k-values,
- I documented the which parameters were the best for each scheme (***n_neighbors=3 and p=2***)
- **Practical Impact**
 - This method enabled me to **discern the optimal KNN configuration** more rigorously.
 - Both the Uniform and Distance weighting performed best with same hyperparameters, suggesting consistent performance for these variants.

5. Evaluation

- **Metrics Used**
 - **Classification Report** (Precision, Recall, F1-score)
 - **Accuracy Score**
 - **Confusion Matrix**
 - **Error Rate vs. K Value Plot**
 - **ROC Curve & AUC**
- **Results**
 - **Accuracy** is 94.7% for both Uniform and Distance weighting.
 - **ROC AUC**: 0.98 for the best uniform model which demonstrates excellent class separation.
 - **Confusion Matrix**: Less false positives/negatives, indicates robust performance.
- **Analysis**
 - Once the hyperparameters were tuned, both weighting schemes performed to an **equal extent**.
 - Plotting of the **error rate** showed lowest misclassification for **k=3**.
 - The model demonstrated strong discriminative power as confirmed by **ROC curve** and **AUC** metric (0.98)

6. Deployment (Optional)

- **Practical Considerations**
 - In a live healthcare diagnostic setting, a **high recall** (low false negatives) is very important to make sure malignant cases are identified early.
 - As its performance metrics are so strong, the model may be implemented into a clinical decision-support pipeline, after further validations in real-world settings.
- **Next Steps**
 - Perhaps I can compare KNN's outcomes with other classifiers (e.g., **Logistic Regression, Random Forest**) for a more in-depth analysis.

- Additional interpretability methods (e.g., **LIME**, **SHAP**) could be useful in a clinical setting.

Summary of Changes & Their Impact

1. Dataset Replacement

- **From:** “Classified Data”
- **To:** Breast Cancer Wisconsin (real-life medical dataset)
- **Effect:** More clinically relevant scenario (new data distributions), extensive feature scaling and parameters tuning.

2. GridSearchCV: Tuning Hyperparameter

- **From:** Manual iteration over k-values
- **To:** In-depth grid search across **n_neighbors** (1–30) and **p** (1, 2)
- **Impact:** Found optimal **n_neighbors=3** and **p=2** for both uniform and distance weighting thus increasing overall accuracy.

3. Weighting Scheme Comparison

- **From:** Only uniform weighting
- **To:** Additional distance weighting method
- **Impact:** Demonstrated that **both methods** can achieve high accuracy (~94.7%) with the correct hyperparameters.

4. Expanded Evaluation Metrics

- **Added:** ROC curve, AUC, error rate plot, confusion matrix
- **Impact:** Gave a more complete perspective on the performance of the model, such as the sensitivity of it to different thresholds and the discriminate ability between classes.

Overall Reflection

By replacing the dataset, improving the hyperparameter tuning, and expanding the evaluation, I developed a deeper theoretical and practical understanding of the KNN algorithm. These models all performed well with the final models yielding **high accuracy** and **strong ROC AUC** scores, demonstrating that the changes were effective. At every step, I documented my why I was doing, what I was doing, and the resulting improvements, satisfying the requirement to keep a detailed log of all changes made and how they practically affected.