PPREDICTING MOLECULAR MUTAGENICITY USING KNN FOR SPR MODELING

Presented By : Raj Aryan

Problem Statement :

Predicting mutagenicity is crucial for the development of new drugs and industrial chemicals. Traditional testing methods are expensive and time-consuming. Therefore, developing predictive models based on molecular descriptors can help identify potentially harmful compounds more efficiently and reduce the need for animal testing.

Objective :

This project aims to develop a k-Nearest Neighbors (kNN) classification model to predict whether a molecule is mutagenic based on its molecular descriptors. The goal is to optimize the model's performance by tuning its hyperparameters to balance precision and recall, ensuring accurate predictions with minimal false positives and false negatives.

DATASET OVERVIEW:

Dataset : mutagenicity.csv(file used in the analysis).

Source : The dataset contains molecular descriptors and their corresponding mutagenicity (whether a molecule is mutagenic or not)

Target Variable: The target variable is a binary label indicating whether a molecule is mutagenic (1) or non-mutagenic (0).

Preprocessing: Missing values were handled by filling them with the mean of the respective columns.

METHODOLGY

k-Nearest Neighbors (kNN) Algorithm:

- The k-Nearest Neighbors (kNN) algorithm is a simple yet powerful classification method based on distance metrics.
- For a given test data point, the algorithm identifies the 'k' closest neighbors in the training dataset and predicts the class label through majority voting.
- **Key parameters include:**

n_neighbors : The number of neighbors to consider.
Weights: The weighting scheme for neighbor contributions (e.g., uniform or distance-based).
Metric: The distance metric used (e.g., Euclidean or Manhattan).
Algorithm: The computational method for finding nearest neighbors.

MODELING WORKFLOW

- **Data Loading:** The dataset is loaded from a CSV file containing molecular descriptors and mutagenicity labels.
- **Feature Selection:** Relevant molecular descriptors (e.g., TPSA, MolWt, BalabanJ index) are identified and selected for prediction**.**
- **Preprocessing:** Features are standardized using StandardScaler to ensure all descriptors are on the same scale, which is critical for distance-based algorithms like kNN.
- **Model Training:** A kNN model is trained on the preprocessed training dataset.
- **Hyperparameter Tuning:** The hyperparameters (e.g., n_neighbors, metric) are optimized using GridSearchCV with cross-validation to ensure robust model performance**.**
- **Model Evaluation:** The model's performance is evaluated using metrics such as F1-score, accuracy, precision, recall, and a confusion matrix.

FEATURE SELECTION

Selected Features:

Features such as NumValenceElectrons, MolLogP, and other molecular properties were selected for the model.
These features capture essential characteristics of molecules that may influence their mutagenicity.

Preprocessing:

Standardization: The features were standardized using StandardScaler, which scales the data to have a mean of zero and a unit variance.
This ensures that all features contribute equally to the model, preventing variables with larger scales from dominating the distance calculations.

HYPERPARAMETER OPTIMIZATION

Hyperparameters for kNN:

n_neighbors: The number of nearest neighbors to consider.
weights: The method to weight the neighbors (e.g., 'uniform' or 'distance').
metric: The distance metric used for computing neighbors (e.g., 'euclidean', 'manhattan', 'minkowski').

Optimization Process:

- GridSearchCV: Used for hyperparameter tuning, searching through a specified grid of parameters.
- Cross-Validation: 5-fold cross-validation was employed to evaluate the model for each combination of hyperparameters.
- Scoring Metric: The F1-score was used as the scoring metric, as it balances precision and recall, which are crucial for mutagenicity prediction.

MODEL EVALUATION

Performance Metrics:

F1-Score: The primary metric used to evaluate the model. It balances precision and recall, minimizing both false positives and false negatives.

Accuracy: The proportion of correct predictions (true positives and true negatives).

Precision: The proportion of correctly predicted positive cases (mutagenic molecules).

Recall: The proportion of actual positive cases correctly predicted.

Confusion Matrix: Used to visualize the performance of the classification model in terms of true positives, true negatives, false positives, and false negatives.

Classification Report: Provides a summary of key classification metrics, including F1-score, precision, recall, and support.

RESULTS

Model Evaluation on test set :

F1-Score on Test Set: 0.7398

Accuracy: 0.7016

Precision: 0.7276

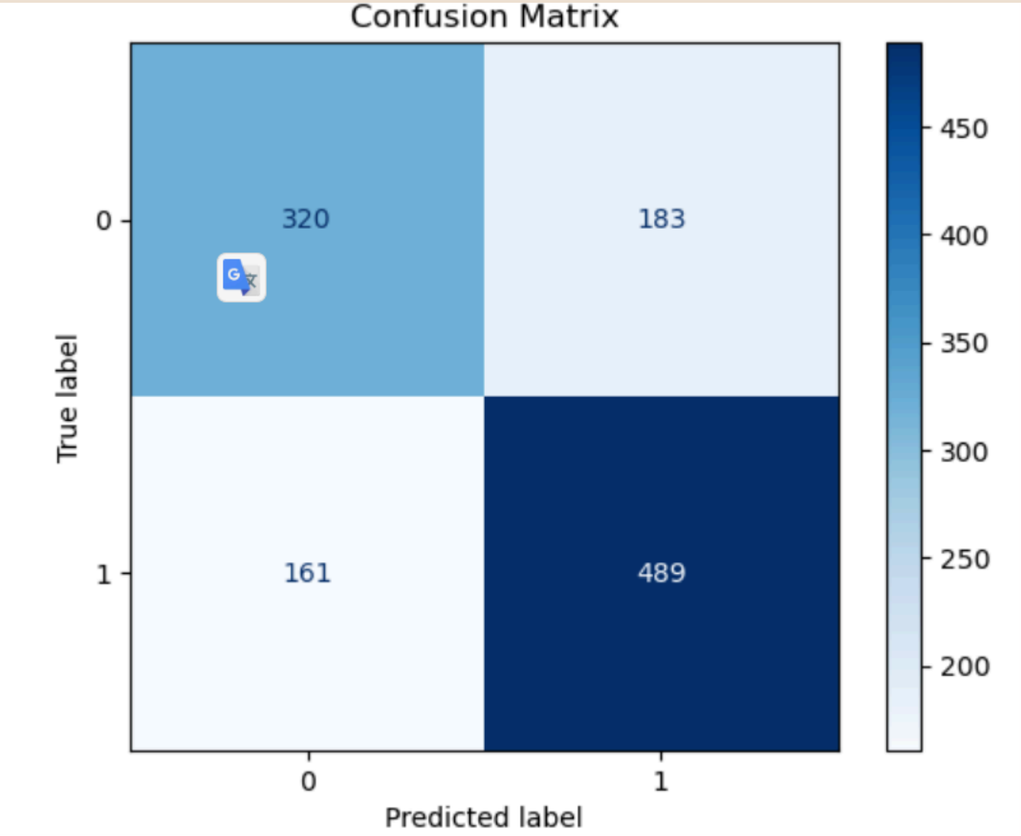
Recall: 0.7523

Confusion Matrix:

[[320 183]

[161 489]]

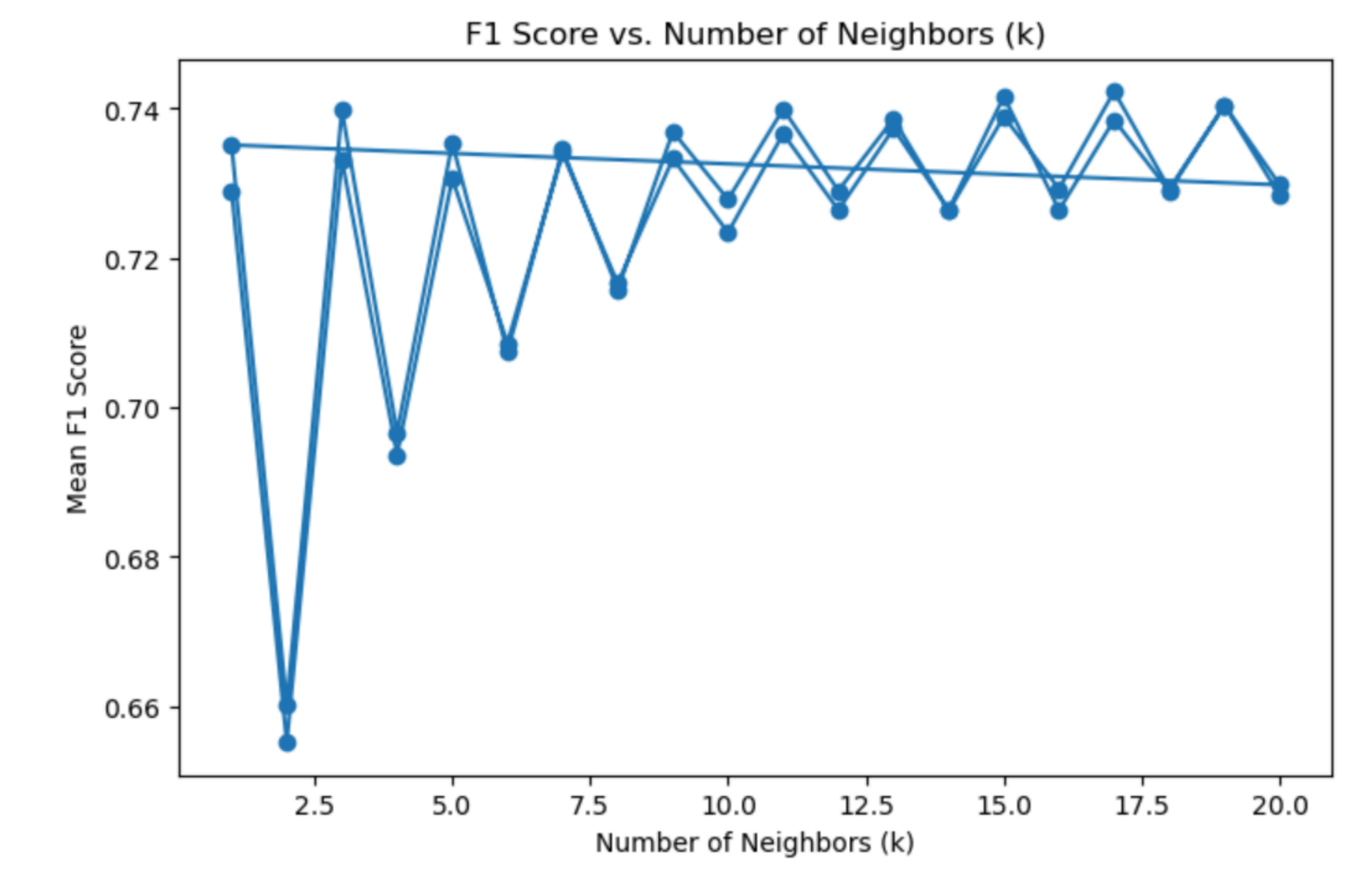
CONFUSION MATRIX AND CLASSIFICATION REPORT



Classification Report:

	precision	recall	f1-score	support
0	0.67	0.64	0.65	503
1	0.73	0.75	0.74	650
accuracy			0.70	1153
macro avg	0.70	0.69	0.70	1153
weighted avg	0.70	0.70	0.70	1153

PLOT OF F1 SCORE VS NUMBER OF NEIGHBORS(K)



CONCLUSION

The kNN model demonstrated strong predictive capability for classifying experimental chemical properties, achieving an **accuracy of 70.16% and an F1-score of 73.97%.**

Future Work:

Explore alternative classifiers (e.g., Random Forest, SVM) to compare performance.

Investigate advanced feature extraction techniques to improve model interpretability and predictive power.

Conduct further hyperparameter tuning to enhance the model's performance and generalize its predictive capabilities.

THANK YOU