MACHINE LEARNNING

Author: Mahmoud M. Shoieb

Email: mahmoudshoieb12@gmail.com

Project Description:
This project focuses on predicting the likelihood of diabetes based on health indicators and lifestyle-related attributes. Using machine learning classification models, the project explores data preprocessing, feature selection, and model evaluation to identify key predictors that influence diabetes risk.

Technologies & Tools:
Python, Scikit-learn, Pandas, NumPy, Matplotlib, Seaborn

Dataset Diabetes Health Indicators

Data description:

The Behavioral Risk Factor Surveillance System (BRFSS) is a health-related telephone survey that is collected annually by the CDC. Each year, the survey collects responses from over 400,000 Americans on health-related risk behaviors, chronic health conditions, and the use of preventative services. It has been conducted every year since 1984. For this project, a csv of the dataset available on Kaggle for the year 2015 was used. This original dataset contains responses from 441,455 individuals and has 330 features. These features are either questions directly asked of participants, or calculated variables based on individual participant responses.
diabetes _ 012 _ health _ indicators _ BRFSS2015.csv is a clean dataset of 253,680 survey responses to the CDC's BRFSS2015. The target variable Diabetes_012 has 3 classes. 0 is for no diabetes or only during pregnancy, 1 is for prediabetes, and 2 is for diabetes. This dataset has 21 feature variables

Data preprocessing:

1. **Removing outliers and duplicates:**

```
✓ 0s  [4]  data.dropna(inplace=True)
           data.drop_duplicates(inplace=True)
```

2. **Handling outliers:**

Outliers in Income: 0

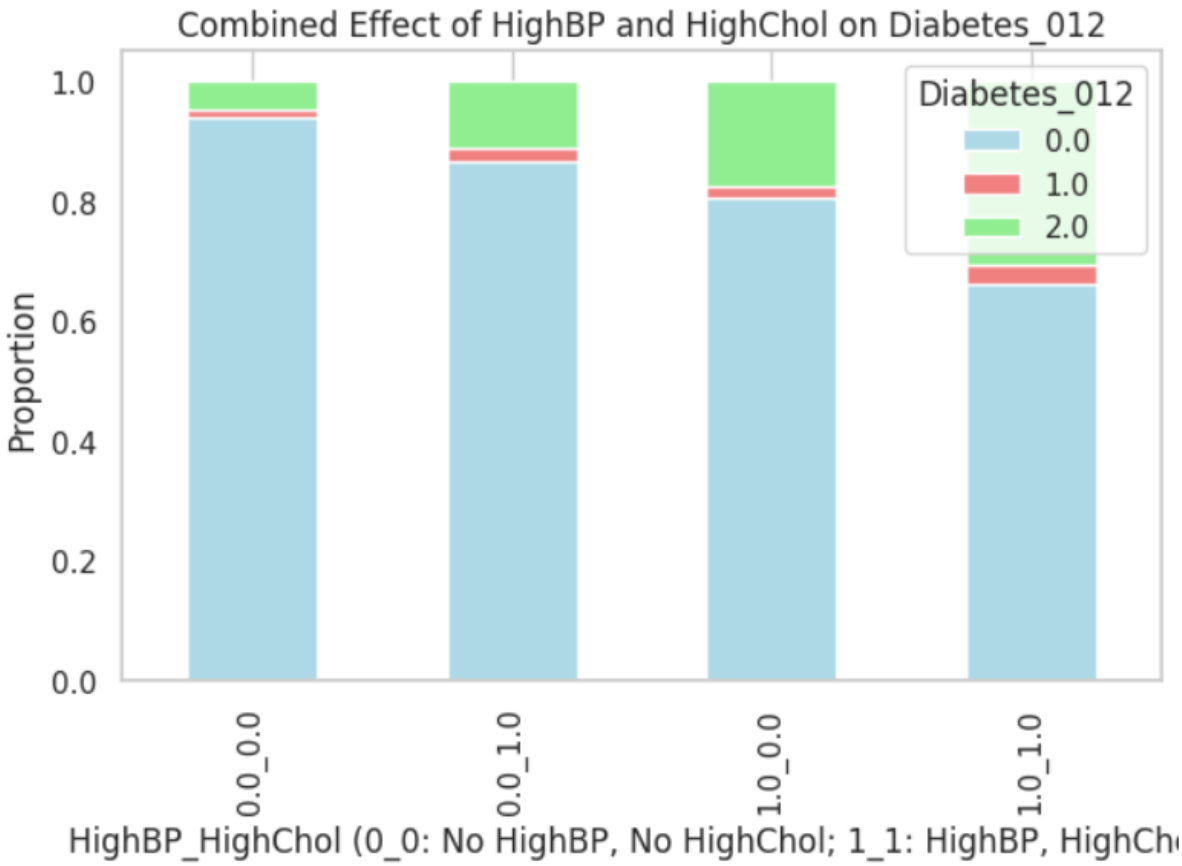
	Column	Number of Outliers
0	Diabetes_012	39726
1	HighBP	0
2	HighChol	0
3	CholCheck	9298
4	BMI	5638
5	Smoker	0
6	Stroke	10284
7	HeartDiseaseorAttack	23717
8	PhysActivity	0
9	Fruits	0
10	Veggies	47148
11	HvyAlcoholConsump	13950
12	AnyHealthcare	12391
13	NoDocbcCost	21326
14	GenHlth	12078
15	MentHlth	36163
16	PhysHlth	34347
17	DiffWalk	42626
18	Sex	0
19	Age	0
20	Education	0
21	Income	0

As we can see the only value that is continuous and has outliers is BMI since the other discrete values has no typing errors in them, then the only column that needs to be handled is BMI :

```
# handle outliers
#choosing bmi is due the fact that Extremely high or low BMI values are uncommon in healthy populations
#and among the columns that has outliers it is the only continuous one
Q1 = data['BMI'].quantile(0.25)
Q3 = data['BMI'].quantile(0.75)
IQR = Q3 - Q1
upper_limit = Q3 + 1.5 * IQR
data['BMI'] = data['BMI'].apply(lambda x: upper_limit if x > upper_limit else x)
```

Data visualization:

1. **How high Blood pressure and high cholesterol affects diabetes:**



The graph shows that having both high blood pressure and high cholesterol significantly increases the risk of developing diabetes.

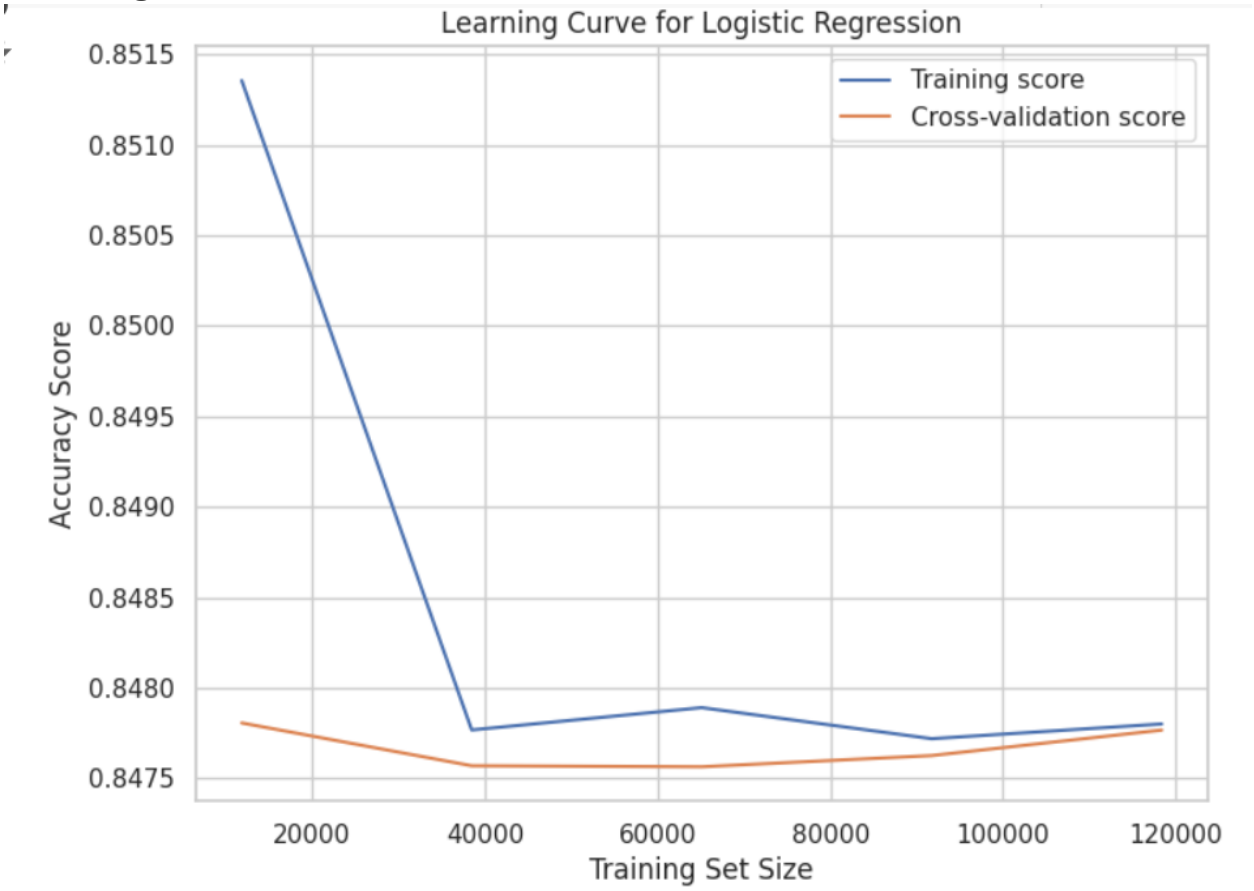
Classification models

1) Logistic regression model :

Hyperparameter tuning:

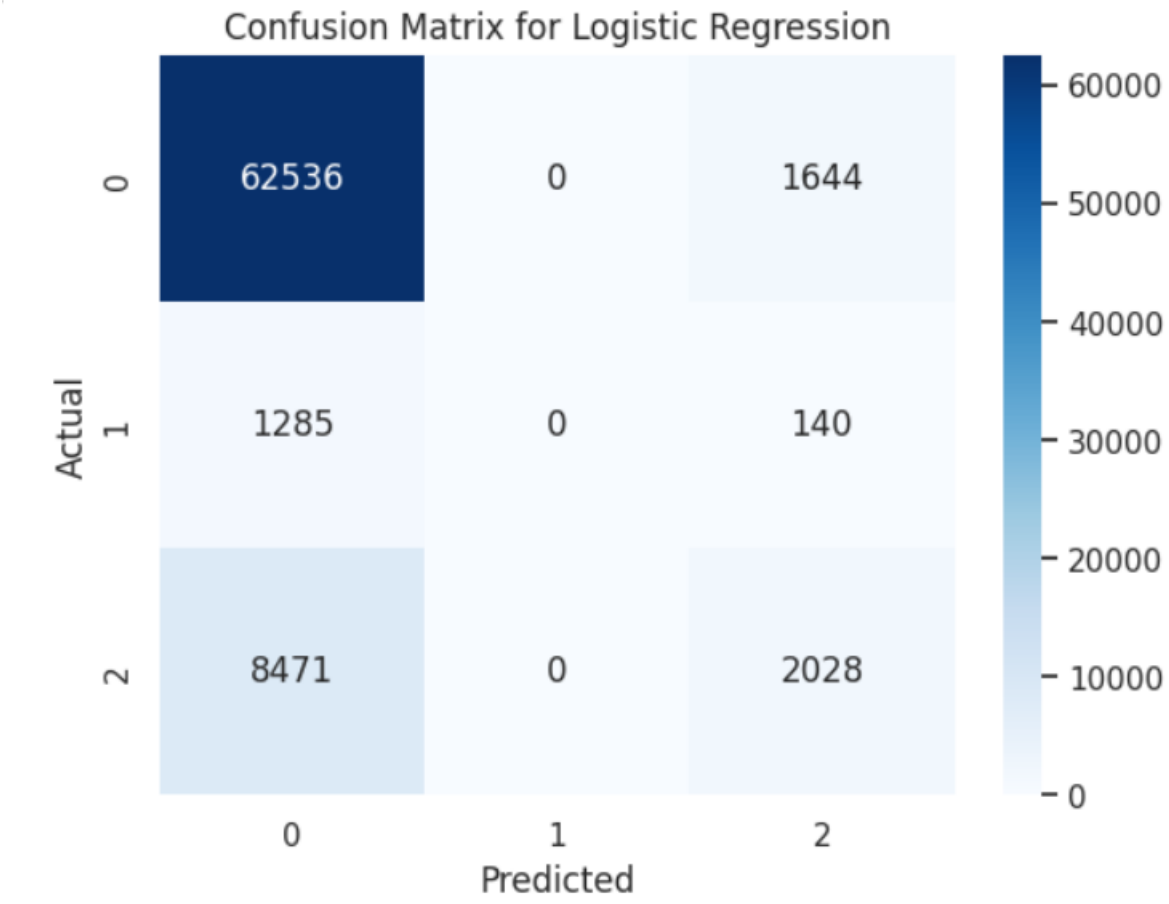
```
Hyperparameter Tuning for Logistic Regression...
Best Logistic Regression Model: LogisticRegression(C=1, max_iter=1000)
```

Learning curve:



this learning curve suggests that the logistic regression model is performing well and is not overfitting. The model's performance is likely to improve further with a larger training set.

Confusion matrix:



In this confusion matrix, the model has the highest accuracy for class 0, with 62536 correct predictions.

Overall, the model has a relatively high accuracy.

Classification report:

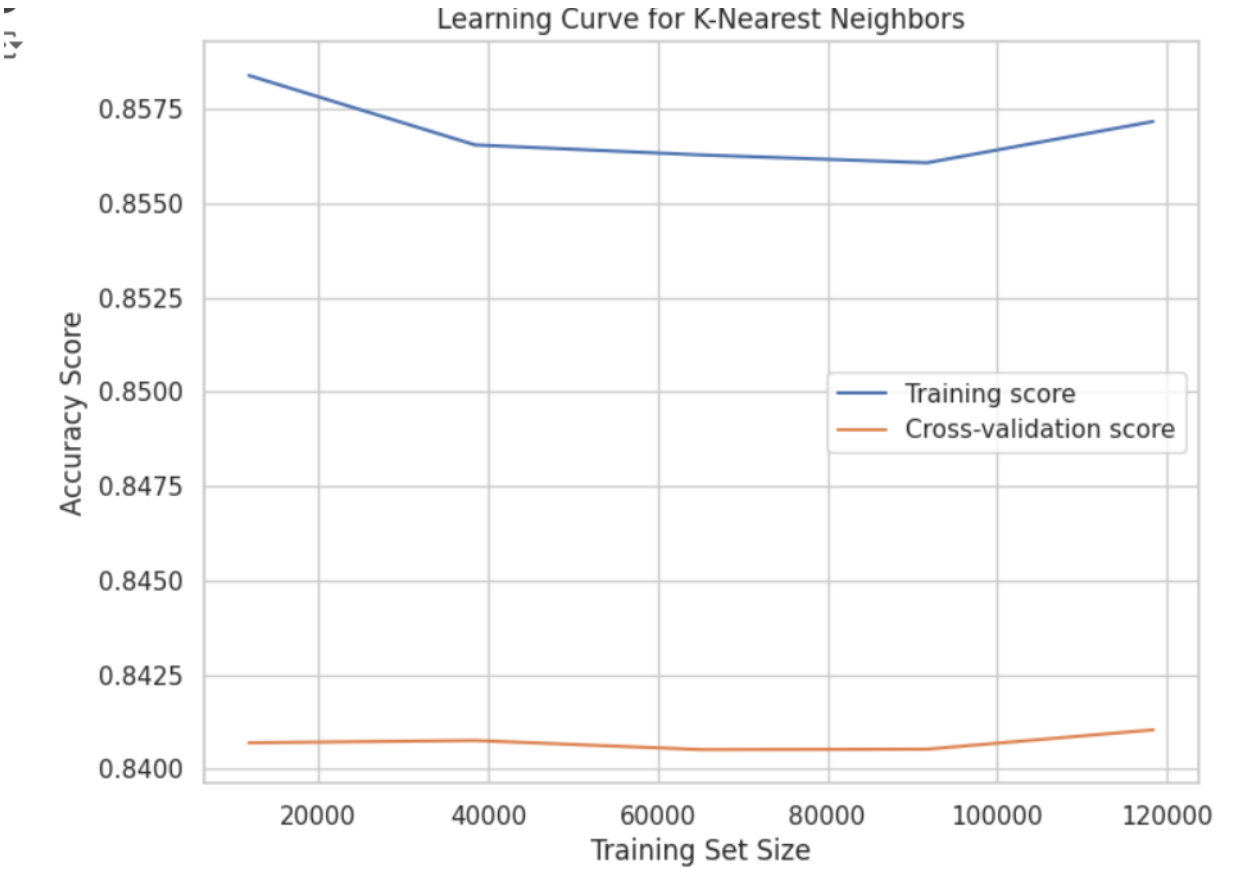
```
Classification Report:
              precision    recall  f1-score   support

         0.0       0.87      0.97      0.92     64180
         1.0       0.00      0.00      0.00      1425
         2.0       0.53      0.19      0.28     10499

    accuracy                           0.85     76104
   macro avg       0.47      0.39      0.40     76104
weighted avg       0.80      0.85      0.81     76104
```

2) K-nearest neighbors model:

Hyperparameter tunning:

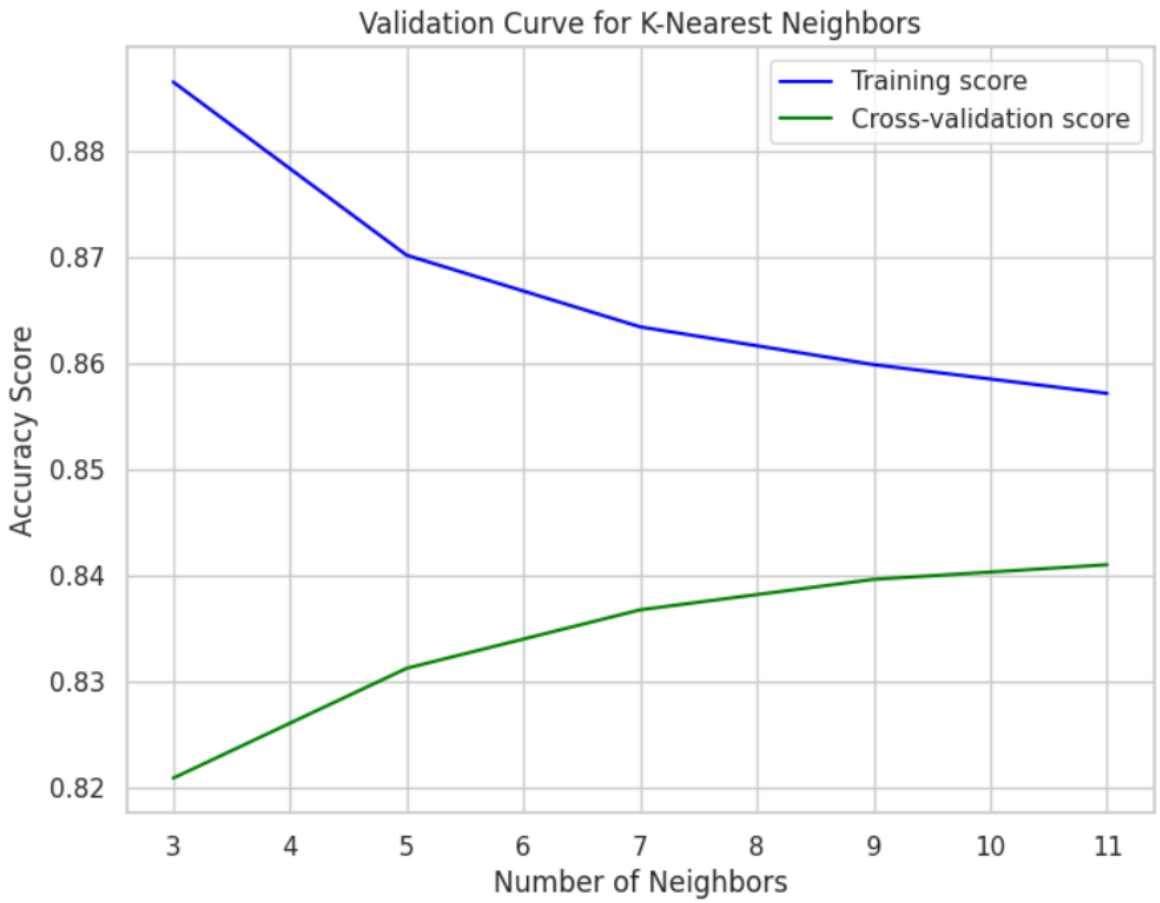
```
Hyperparameter Tuning for K-Nearest Neighbors...
Best K-Nearest Neighbors Model: KNeighborsClassifier(n_neighbors=11)
```

Learning curve:



Overall, this learning curve suggests that the KNN model is overfitting the training data. And that's why parameter curve was used

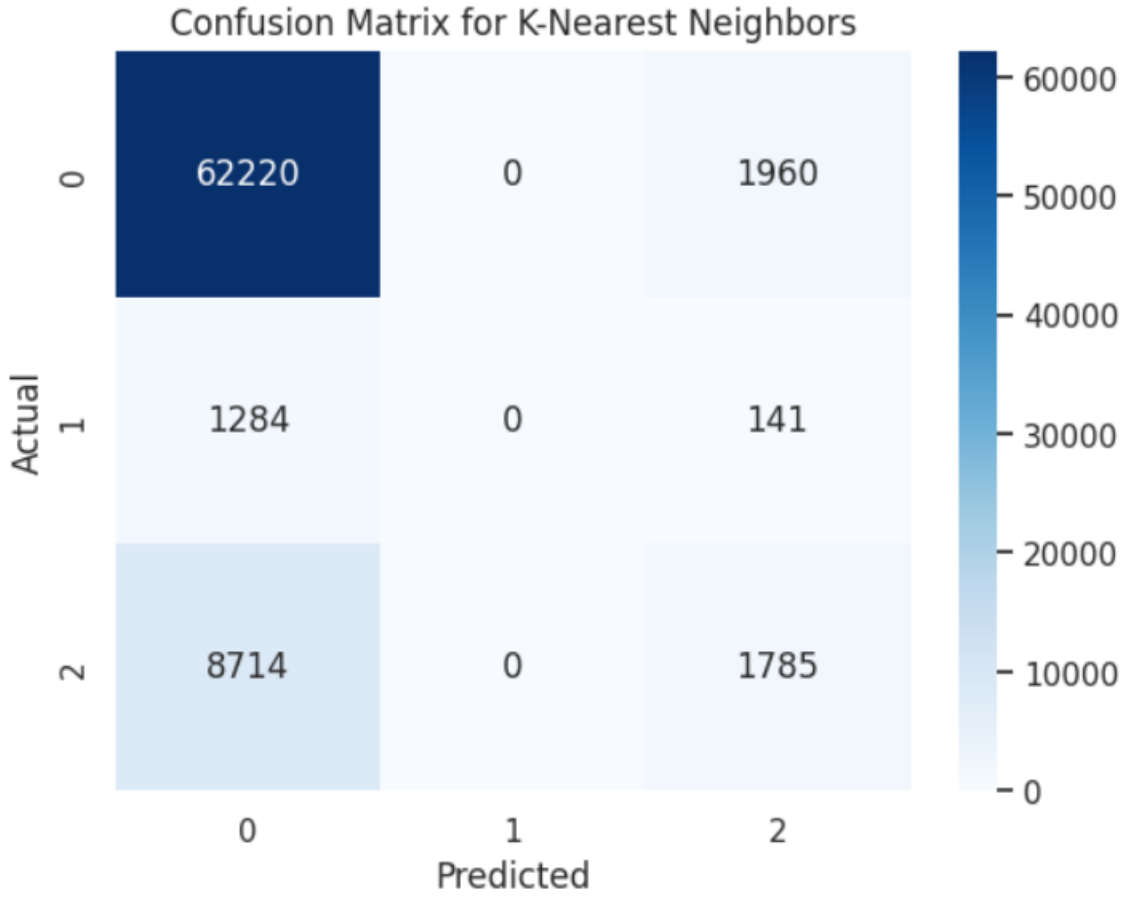
Parameter curve:



Overall, this validation curve suggests that increasing the number of neighbors can help to reduce overfitting in the KNN model. By increasing the number of neighbors, the model becomes less sensitive to noise in the data and makes more robust predictions.

Based on this validation curve, a value of around 7-8 neighbors seems to be a good choice, as it balances the trade-off between overfitting and underfitting.

Confusion matrix:



Classification report:

Classification Report:

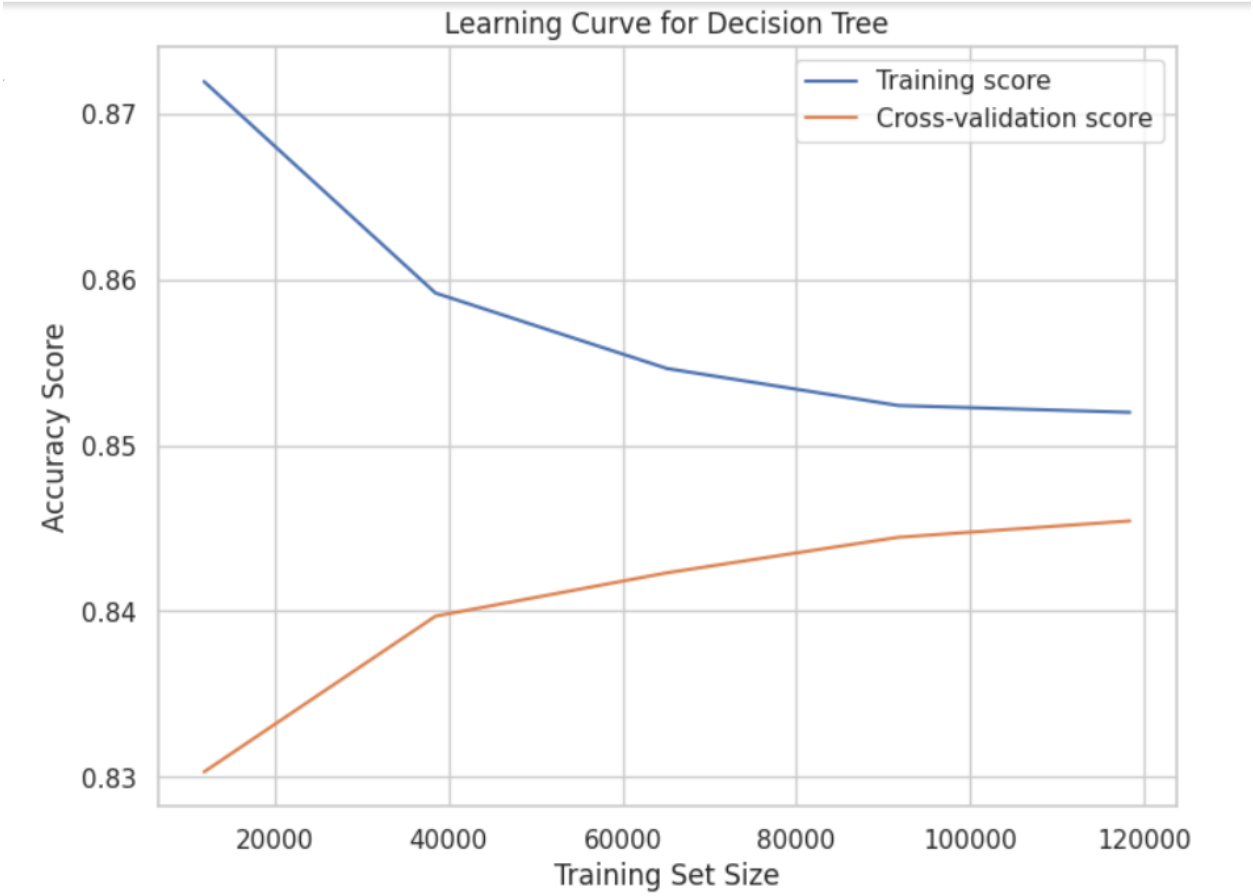
	precision	recall	f1-score	support
0.0	0.86	0.97	0.91	64180
1.0	0.00	0.00	0.00	1425
2.0	0.46	0.17	0.25	10499
accuracy			0.84	76104
macro avg	0.44	0.38	0.39	76104
weighted avg	0.79	0.84	0.80	76104

3) Decision tree model:

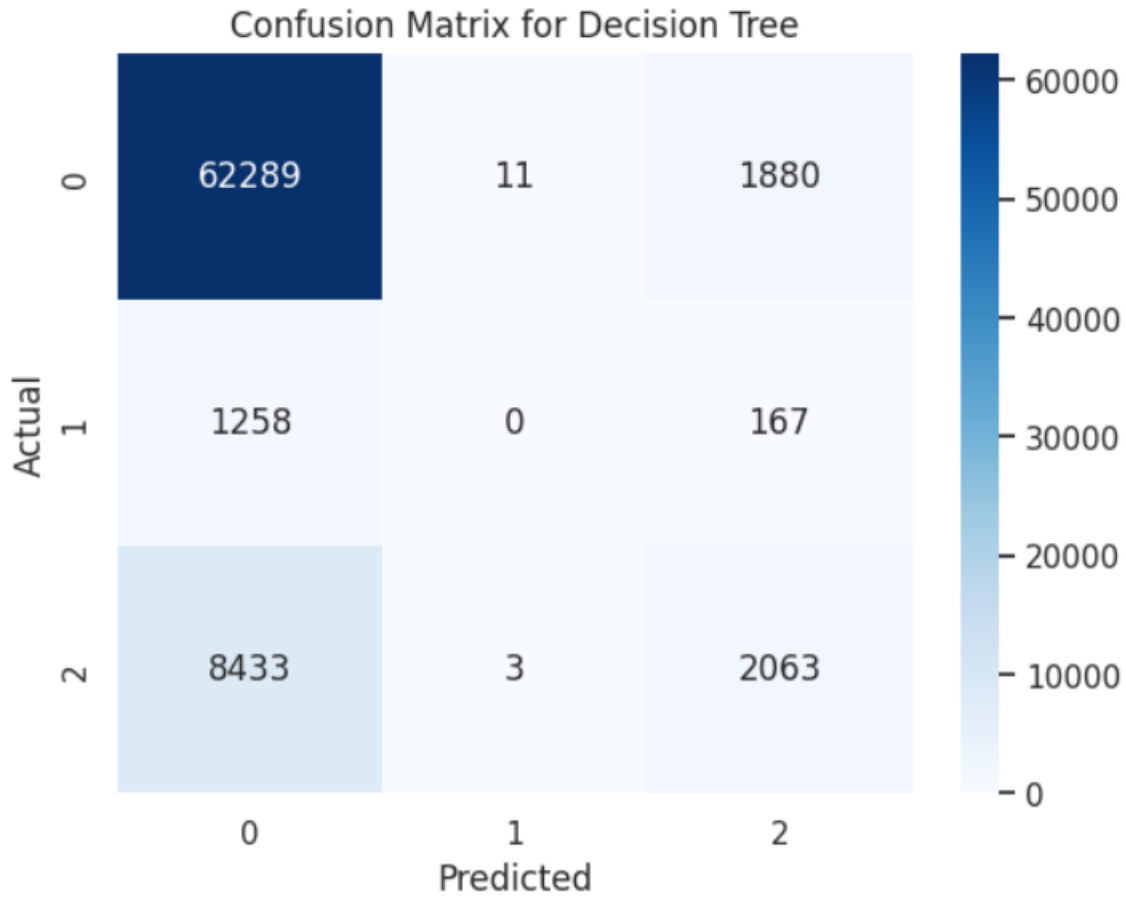
Hyperparameter tunning:

```
Hyperparameter Tuning for Decision Tree...
Best Decision Tree Model: DecisionTreeClassifier(criterion='entropy', max_depth=10, min_samples_leaf=4,
                       min_samples_split=10)
```

Learning curve:



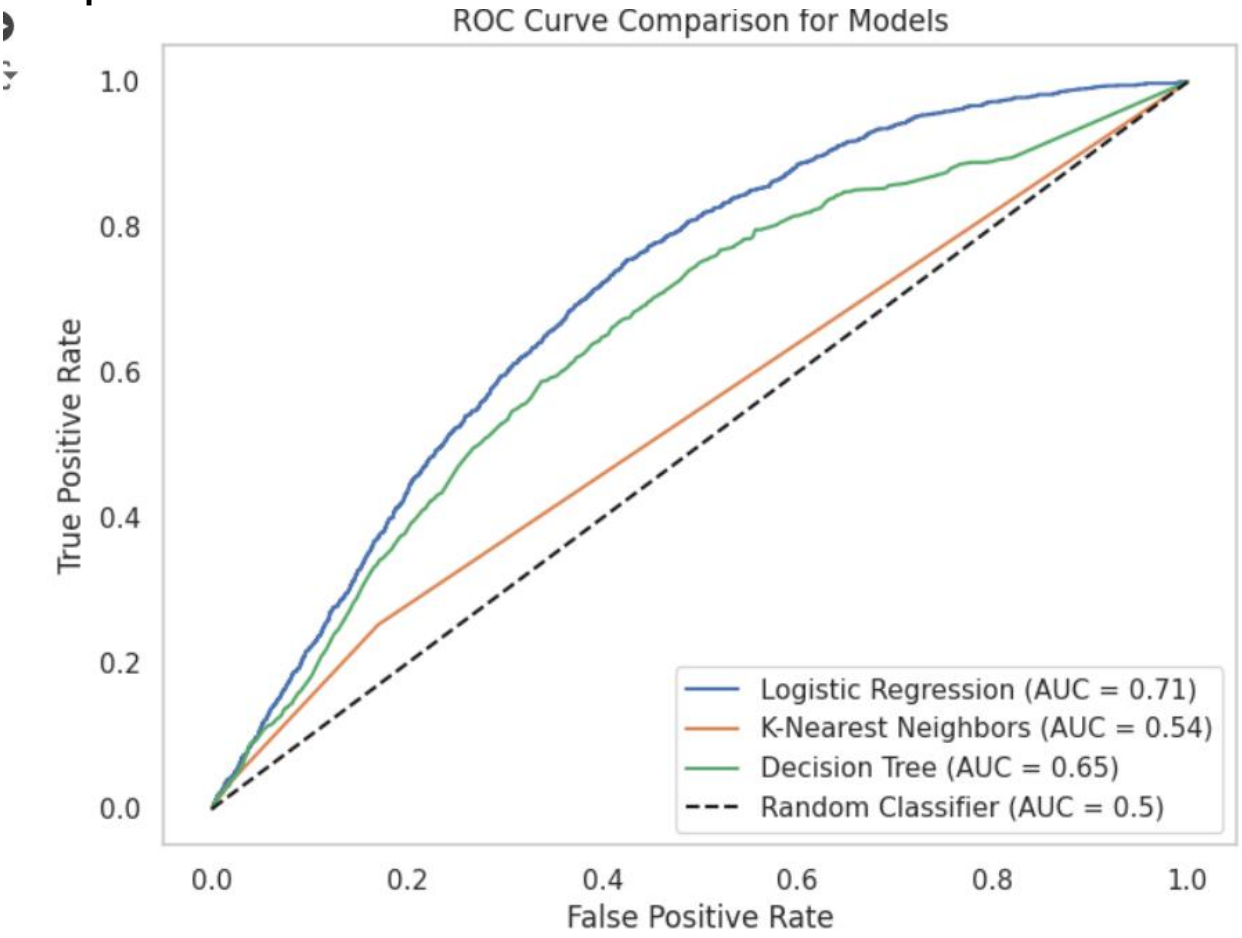
Confusion matrix:



Classification report:

```
Training Decision Tree...
Classification Report:
              precision    recall  f1-score   support

         0.0       0.87      0.97      0.91     64180
         1.0       0.00      0.00      0.00      1425
         2.0       0.50      0.20      0.28     10499

    accuracy                           0.85     76104
   macro avg       0.46      0.39      0.40     76104
weighted avg       0.80      0.85      0.81     76104
```

Comparison between models:



Model Comparison:

	Accuracy	F1 Score	Precision	Recall
Logistic Regression	0.848365	0.811973	0.802905	0.848365
K-Nearest Neighbors	0.841020	0.803623	0.789938	0.841020
Decision Tree	0.845580	0.810548	0.799026	0.845580