**Московский государственный технический университет им. Н. Э. Баумана**

Курс «Технологии машинного обучения»

Отчёт по лабораторной работе №3

Выполнил:
Флоринский В. А.
группа ИУ5-64Б

Проверил:
Гапанюк Ю.Е.

Дата: 07.04.25

Подпись:

Дата:

Подпись:

Москва, 2025 г.

**Цель лабораторной работы:** изучение способов подготовки выборки и подбора гиперпараметров на примере метода ближайших соседей.

**Задание:**

1. Выберите набор данных (датасет) для решения задачи классификации или регрессии.
2. В случае необходимости проведите удаление или заполнение пропусков и кодирование категориальных признаков.
3. С использованием метода train_test_split разделите выборку на обучающую и тестовую.
4. Обучите модель ближайших соседей для произвольно заданного гиперпараметра K. Оцените качество модели с помощью подходящих для задачи метрик.
5. Произведите подбор гиперпараметра K с использованием GridSearchCV и RandomizedSearchCV и кросс-валидации, оцените качество оптимальной модели. Используйте не менее двух стратегий кросс-валидации.
6. Сравните метрики качества исходной и оптимальной моделей.

**Ход выполнения:**

```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, GridSearchCV, RandomizedSearchCV, StratifiedKF
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report
```

Python

```python
df = pd.read_csv("student_depression_dataset.csv")
```

Python

```python
df.info(), df.head()
```

Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27901 entries, 0 to 27900
Data columns (total 18 columns):
 #   Column                                 Non-Null Count  Dtype
---  ------                                 --------------  -----
 0   id                                     27901 non-null  int64
 1   Gender                                 27901 non-null  object
 2   Age                                    27901 non-null  float64
 3   City                                   27901 non-null  object
 4   Profession                             27901 non-null  object
 5   Academic Pressure                      27901 non-null  float64
 6   Work Pressure                          27901 non-null  float64
 7   CGPA                                   27901 non-null  float64
 8   Study Satisfaction                     27901 non-null  float64
 9   Job Satisfaction                       27901 non-null  float64
 10  Sleep Duration                         27901 non-null  object
 11  Dietary Habits                         27901 non-null  object
 12  Degree                                 27901 non-null  object
 13  Have you ever had suicidal thoughts ?  27901 non-null  object
```

```
RangeIndex: 27901 entries, 0 to 27900
Data columns (total 18 columns):
 #   Column                                 Non-Null Count  Dtype  
---  ------                                 --------------  -----  
 0   id                                     27901 non-null  int64  
 1   Gender                                 27901 non-null  object 
 2   Age                                    27901 non-null  float64
 3   City                                   27901 non-null  object 
 4   Profession                             27901 non-null  object 
 5   Academic Pressure                      27901 non-null  float64
 6   Work Pressure                          27901 non-null  float64
 7   CGPA                                   27901 non-null  float64
 8   Study Satisfaction                     27901 non-null  float64
 9   Job Satisfaction                       27901 non-null  float64
 10  Sleep Duration                         27901 non-null  object 
 11  Dietary Habits                         27901 non-null  object 
 12  Degree                                 27901 non-null  object 
 13  Have you ever had suicidal thoughts ?  27901 non-null  object 
 14  Work/Study Hours                       27901 non-null  float64
 15  Financial Stress                       27901 non-null  object 
 16  Family History of Mental Illness       27901 non-null  object 
 17  Depression                             27901 non-null  int64  
dtypes: float64(7), int64(2), object(9)
memory usage: 3.8+ MB

(None,
    id  Gender   Age           City Profession  Academic Pressure  \
 0   2    Male  33.0  Visakhapatnam    Student                5.0   
 1   8  Female  24.0      Bangalore    Student                2.0   
 2  26    Male  31.0       Srinagar    Student                3.0   
 3  30  Female  28.0       Varanasi    Student                3.0   
 4  32  Female  25.0         Jaipur    Student                4.0   

    Work Pressure  CGPA  Study Satisfaction  Job Satisfaction  \
 0            0.0  8.97                 2.0               0.0   
 1            0.0  5.90                 5.0               0.0   
 2            0.0  7.03                 5.0               0.0   
 3            0.0  5.59                 2.0               0.0   
```

```
   Work Pressure  CGPA  Study Satisfaction  Job Satisfaction  \
0            0.0  8.97                 2.0               0.0
1            0.0  5.90                 5.0               0.0
2            0.0  7.03                 5.0               0.0
3            0.0  5.59                 2.0               0.0
4            0.0  8.13                 3.0               0.0

        Sleep Duration Dietary Habits   Degree  \
0         '5-6 hours'        Healthy  B.Pharm
1         '5-6 hours'       Moderate      BSc
2  'Less than 5 hours'        Healthy       BA
3         '7-8 hours'       Moderate      BCA
4         '5-6 hours'       Moderate   M.Tech

  Have you ever had suicidal thoughts ?  Work/Study Hours Financial Stress  \
0                                   Yes               3.0              1.0
1                                    No               3.0              2.0
...
0                                  No           1
1                                 Yes           0
2                                 Yes           0
3                                 Yes           1
4                                  No           0  )
```

*Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...*

```python
# Предобработка
df.drop(columns=['id'], inplace=True)  # Удаление лишнего
df['Sleep Duration'] = df['Sleep Duration'].str.replace("'", "").str.strip()
df['Financial Stress'] = df['Financial Stress'].replace('?', np.nan)
df['Financial Stress'] = df['Financial Stress'].astype(float)
df['Financial Stress'] = df['Financial Stress'].fillna(df['Financial Stress'].median())
```

[10] Python

```python
valid_cities = df['City'].value_counts().loc[lambda x: x > 50].index
df = df[df['City'].isin(valid_cities)]
# Удаление мусорных значений из "City"
```

[ ] Python

```python
categorical_cols = df.select_dtypes(include='object').columns
df_encoded = pd.get_dummies(df, columns=categorical_cols, drop_first=True)
# One-hot кодирование категориальных признаков
```

[ ] Python

```python
X = df_encoded.drop("Depression", axis=1)
y = df_encoded["Depression"]
```

[13] Python

```python
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

print(X_train.shape, X_test.shape)
```

[22] Python

```
(22300, 87) (5575, 87)
```

```python
knn_initial = KNeighborsClassifier(n_neighbors=5)
knn_initial.fit(X_train, y_train)
y_pred_initial = knn_initial.predict(X_test)
```

[15] Python

```python
# Оценка базовой модели
print("Базовая модель (K=5)")
print("Accuracy:", accuracy_score(y_test, y_pred_initial))
print("Classification report:\n", classification_report(y_test, y_pred_initial))
```

[16] Python

```
Базовая модель (K=5)
Accuracy: 0.8052017937219731
Classification report:
              precision    recall  f1-score   support

           0       0.79      0.73      0.76      2311
           1       0.82      0.86      0.84      3264

    accuracy                           0.81      5575
   macro avg       0.80      0.79      0.80      5575
weighted avg       0.80      0.81      0.80      5575
```

```python
param_grid = {'n_neighbors': list(range(1, 21))}
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
```

[17] Python

```python
grid_search = GridSearchCV(KNeighborsClassifier(), param_grid, cv=cv, scoring='accuracy')
grid_search.fit(X_train, y_train)

random_search = RandomizedSearchCV(KNeighborsClassifier(), param_grid, n_iter=10, cv=cv, scoring='ac
random_search.fit(X_train, y_train)
```

[18] Python

```
    print("GridSearchCV: Лучшее K =", grid_search.best_params_['n_neighbors'],
          "Средняя точность:", grid_search.best_score_)
    print("RandomizedSearchCV: Лучшее K =", random_search.best_params_['n_neighbors'],
          "Средняя точность:", random_search.best_score_)
```
[19]

```
GridSearchCV: Лучшее K = 20 Средняя точность: 0.8140807174887893
RandomizedSearchCV: Лучшее K = 18 Средняя точность: 0.8125112107623318
```

```
    best_k = grid_search.best_params_['n_neighbors']
    knn_optimized = KNeighborsClassifier(n_neighbors=best_k)
    knn_optimized.fit(X_train, y_train)
    y_pred_optimized = knn_optimized.predict(X_test)
```
[20]

```
    print(f"\nОптимизированная модель (K={best_k})")
    print("Accuracy:", accuracy_score(y_test, y_pred_optimized))
    print("Classification report:\n", classification_report(y_test, y_pred_optimized))
```
[21]

```
Оптимизированная модель (K=20)
Accuracy: 0.8224215246636771
Classification report:
              precision    recall  f1-score   support

           0       0.80      0.76      0.78      2311
           1       0.84      0.87      0.85      3264

    accuracy                           0.82      5575
   macro avg       0.82      0.81      0.82      5575
weighted avg       0.82      0.82      0.82      5575
```