МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



Факультет прикладной математики и информатики

Кафедра ТПИ

Дисциплина: «Основы теории машинного обучения»

Лабораторная работа №5

**КЛАССИФИКАЦИЯ ПО МЕТОДУ SVM**

Факультет: ФПМИ

Группа: ПМИМ-31

Выполнили: Монгуш Н. С., Тарулин М. А., Филипенко Ю. Д.

Преподаватель: Попов А. А.

Дата выполнения:

Отметка о защите:

1. **Цель работы**

Получить практические навыки по решению задачи классификации с использованием метода опорных векторов.

1. **Задание**

1. Ознакомиться с теоретическими основами метода опорных векторов (SVM), а также метода LS SVM для решения задачи классификации.

2. Осуществить выбор прикладного набора данных двухклассовой классификации из репозиториев данных.

3. Сформировать обучающую и тестовую части выборки.

4. Провести классификацию данных с настройкой внутренних параметров алгоритма SVM по ошибке на тестовой части выборки.

1. **Исходные данные**

В качестве набора данных будут использованы результаты газового анализа двух видов топлив: дизельное топливо и мазут.

Таблица 1. Первые 5 записей газового анализа дизельного топлива.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| F\_fuel | F\_air | O2 | CO | NO | NO2 | NOx | CO2 | SO2 |
| 0.200 | 1.784 | 11.828453 | 679.846033 | 0.0 | 8.741990 | 8.176684 | 7.377569 | 67.435602 |
| 0.216 | 1.784 | 11.828453 | 679.846033 | 0.0 | 8.741990 | 8.176684 | 7.377569 | 67.435602 |
| 0.232 | 1.784 | 11.828453 | 679.846033 | 0.0 | 8.741990 | 8.176684 | 7.377569 | 67.435602 |
| 0.248 | 1.784 | 11.828453 | 679.846033 | 0.0 | 8.741990 | 8.176684 | 7.377569 | 67.435602 |
| 0.264 | 1.784 | 11.807047 | 678.252766 | 0.0 | 8.731427 | 8.176684 | 7.379899 | 67.510727 |

Таблица 2. Первые 5 записей газового анализа мазута.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| F\_fuel | F\_air | O2 | CO | NO | NO2 | NOx | CO2 | SO2 |
| 0.200 | 1.784 | 11.192942 | 958.172981 | 187.324359 | 188.523741 | 7.241927 | 656.002634 | 0.200 |
| 0.216 | 1.784 | 11.192942 | 958.172981 | 187.324359 | 188.523741 | 7.241927 | 656.002634 | 0.216 |
| 0.232 | 1.784 | 11.192942 | 958.172981 | 187.324359 | 188.523741 | 7.241927 | 656.002634 | 0.232 |
| 0.248 | 1.784 | 11.192942 | 958.172981 | 187.331832 | 188.524245 | 7.241927 | 656.002634 | 0.248 |
| 0.264 | 1.784 | 11.177439 | 957.951340 | 187.345810 | 188.531915 | 7.241967 | 656.526937 | 0.264 |

Для выполнения задачи двухклассовой классификации сформируем набор данных для обучения модели из 4,000 случайных записей с процентным соотношением 50.075 и 49.925 соответственно из набора данных для дизельного топлива и мазута, выбрав два компонента газового состава, которые наилучшим образом описывают химический состав топлива: NOx и SO2­ и в третьем столбце укажем тип топлива, где 0 – это мазут, а 1 – дизельное топливо.

Таблица 3. Первые 5 записей обучающей выборки.

|  |  |  |
| --- | --- | --- |
| NOx | SO2 | Fuel\_type |
| 81.439163 | 160.220501 | 1 |
| 222.057444 | 853.923404 | 0 |
| 46.420804 | 121.307775 | 1 |
| 99.925896 | 160.441733 | 1 |
| 89.012879 | 148.947651 | 1 |

Тестовая выборка будет состоять из 20,000 записей с неопределенным типом топлива:

Таблица 4. Первые 5 записей тестовой выборки.

|  |  |  |
| --- | --- | --- |
| NOx | SO2 | Fuel\_type |
| 8.176684 | 67.435602 | None |
| 8.176684 | 67.435602 | None |
| 8.176684 | 67.435602 | None |
| 8.176684 | 67.435602 | None |
| 8.176684 | 67.510727 | None |

1. **Полученные результаты**

В качестве программного обеспечения выступает библиотека python sklearn с классом svm, в котором реализован метод SVM классификации. Настройка внутренних параметров производилась в ручном режиме с помощью метода SVC класса svm с помощью обучающей выборки.

В обученную модель передадим значения NOx, SO2 из тестовой выборки и получим значение типа топлива. Оценка полученных значений с действительными величинами показала 100% точность модели. Изобразим на рисунке 1 точки характерные для дизельного топлива и мазута, разделенные гиперплоскостью, полученную с помощью класса svm.

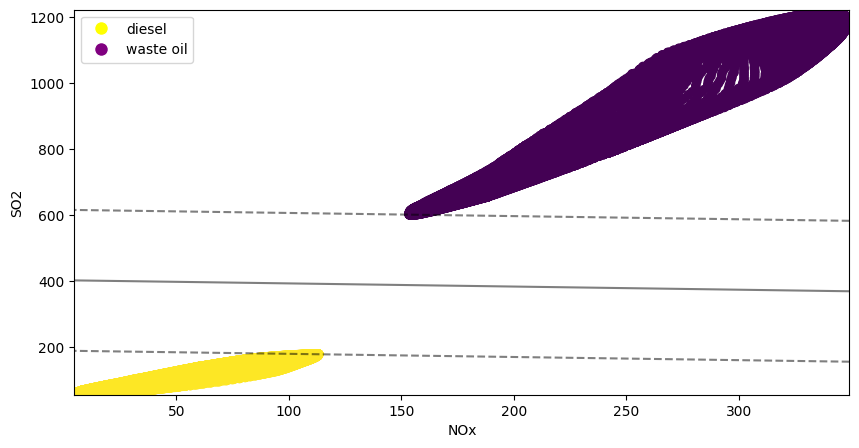


Рисунок 1. Результат SVM классификации для тестовой выборки.

1. **Вывод**

В ходе работы была получена модель SVM с линейным ядром, обученная на выборке из 4,000 записей. Результаты классификации на обучающей и тестовой выборки не имеют ошибок.

1. **Код программы**

**import** **sys**

**import** **os**

**import** **pandas** **as** **pd**

**import** **seaborn** **as** **sns**

**from** **sklearn** **import** svm

**import** **numpy** **as** **np**

**import** **matplotlib.pyplot** **as** **plt**

**from** **sklearn.inspection** **import** DecisionBoundaryDisplay

PATH\_TO\_PROJECT: str = 'C:**\\**Users**\\**Mark**\\**PycharmProjects**\\**machineLearningNstu'

sys.path.append(PATH\_TO\_PROJECT)

PATH\_TO\_DIESEL: str = os.path.join(PATH\_TO\_PROJECT, 'data', 'raw', 'diesel\_air.csv')

PATH\_TO\_HEAVY\_OIL: str = os.path.join(PATH\_TO\_PROJECT, 'data', 'raw', 'heavy\_oil\_air.csv')

diesel\_df: pd.DataFrame = pd.read\_csv(PATH\_TO\_DIESEL)

heavy\_oil\_df: pd.DataFrame = pd.read\_csv(PATH\_TO\_HEAVY\_OIL)

diesel\_df: pd.DataFrame = pd.concat([diesel\_df['NOx'], diesel\_df['SO2']], axis=**1**)

diesel\_df['Type'] = **1**

heavy\_oil\_df: pd.DataFrame = pd.concat([heavy\_oil\_df['NOx'], heavy\_oil\_df['SO2']], axis=**1**)

heavy\_oil\_df['Type'] = **0**

df: pd.DataFrame = pd.concat([diesel\_df, heavy\_oil\_df], axis=**0**)

training\_sample\_size: int = int(len(df) / **5**)

training\_sample: pd.DataFrame = df.sample(training\_sample\_size)

sns.scatterplot(data=training\_sample, x='NOx', y='SO2', hue='Type')

x: list[float, float] = [list(sample) **for** sample **in** zip(training\_sample['NOx'], training\_sample['SO2'])]

y: list[float] = list(training\_sample['Type'])

clf = svm.SVC(C=**1.0**, kernel='linear').fit(x, y)

x: list[float, float] = [list(sample) **for** sample **in** zip(df['NOx'], df['SO2'])]

y: list[float] = list(df['Type'])

x = np.array(x)

**print**('True') **if** np.all(clf.predict(x) == y) **else** **print**('False')

plt.figure(figsize=(**10**, **5**))

plt.scatter(x[:, **0**], x[:, **1**], c=y)

ax = plt.gca()

DecisionBoundaryDisplay.from\_estimator(

clf,

x,

ax=ax,

grid\_resolution=**50**,

plot\_method="contour",

colors="k",

levels=[-**1**, **0**, **1**],

alpha=**0.5**,

linestyles=["--", "-", "--"],

)

plt.xlabel('NOx')

plt.ylabel('SO2')

colors = ['yellow', 'purple']

labels = ['diesel', 'waste oil']

legend\_elements = [Line2D([**0**], [**0**], marker='o', color='w',

label=label, markerfacecolor=color, markersize=**10**)

**for** color, label **in** zip(colors, labels)]

plt.legend(handles=legend\_elements)