

Министерство образования Республики Беларусь
Учреждение образования
«Брестский Государственный технический университет»
Кафедра ИИТ

Лабораторная работа №2
По дисциплине «Основы машинного обучения»
Тема: «Линейные модели
для задач регрессии и классификации»

Выполнил:
Студент 3 курса
Группы АС-65
Касьяник К. А.
Проверил:
Крощенко А. А.

Цель: изучить применение линейной и логистической регрессии для решения практических задач. Научиться обучать модели, оценивать их качество с помощью соответствующих метрик и интерпретировать результаты.

Вариант 7

Регрессия (Прогнозирование цены автомобиля)

1. Car Price Prediction

2. Предсказать цену автомобиля (price)

3. Задания:

- загрузите данные. Выберите 5-6 числовых признаков (например, horsepower, citympg, enginesize);
- обучите модель линейной регрессии;
- рассчитайте R^2 и MAE;
- визуализируйте зависимость price от horsepower с линией регрессии.

Код программы:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_absolute_error
from sklearn.model_selection import train_test_split
import numpy as np

cars = pd.read_csv("CarPrice_Assignment.csv")

features = [
    "horsepower",
    "enginesize",
    "curbweight",
    "citympg",
    "highwaympg"
]

X = cars[features]
y = cars["price"]

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

reg = LinearRegression()
reg.fit(X_train, y_train)

predictions = reg.predict(X_test)

r2_value = r2_score(y_test, predictions)
mae_value = mean_absolute_error(y_test, predictions)

print(f"R² = {r2_value:.4f}")
print(f"MAE = {mae_value:.2f}")

hp = cars["horsepower"].values.reshape(-1, 1)
price = cars["price"].values
```

```

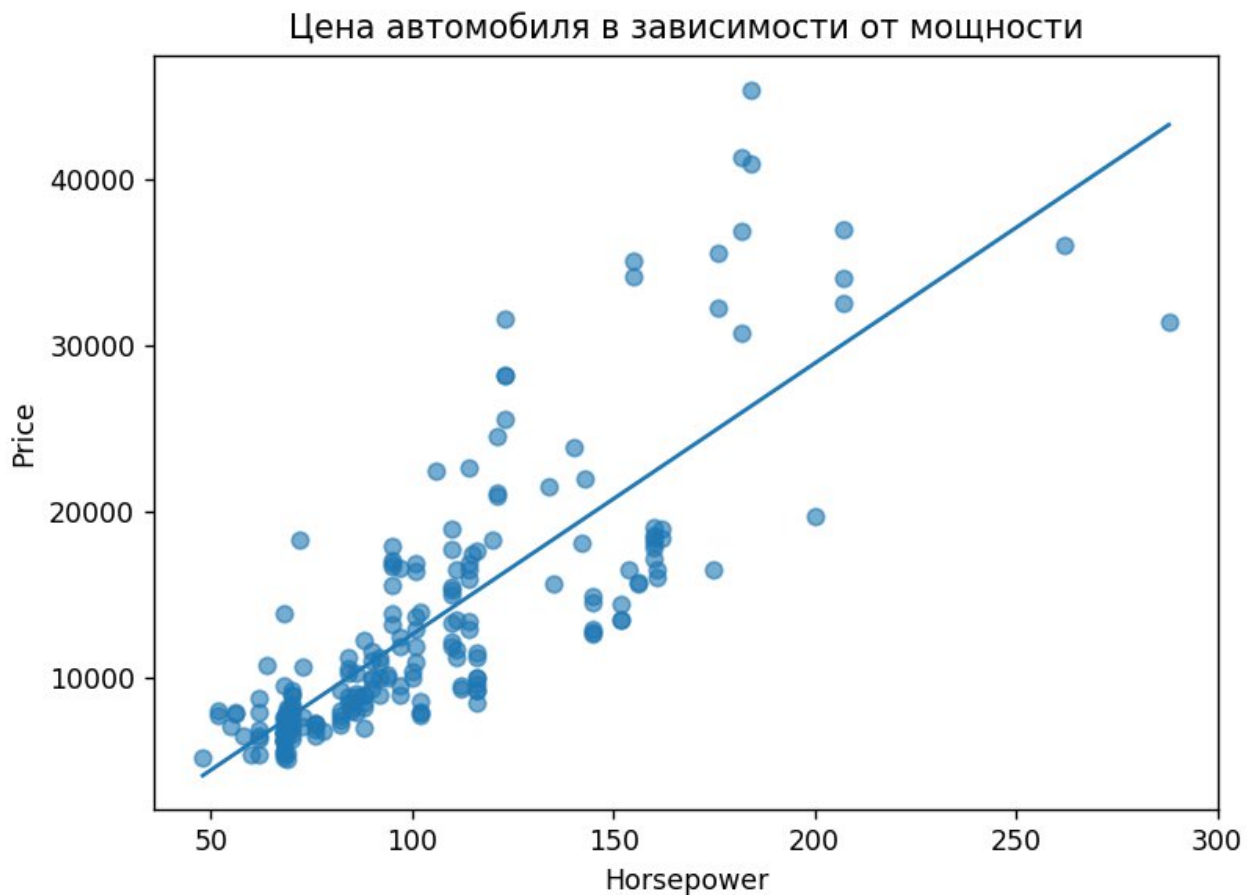
simple_model = LinearRegression()
simple_model.fit(hp, price)

x_line = np.linspace(hp.min(), hp.max(), 100).reshape(-1, 1)
y_line = simple_model.predict(x_line)

plt.figure(figsize=(7, 5))
plt.scatter(hp, price, alpha=0.6)
plt.plot(x_line, y_line)
plt.xlabel("Horsepower")
plt.ylabel("Price")
plt.title("Цена автомобиля в зависимости от мощности")
plt.show()

```

$R^2 = 0.8160$
 $MAE = 2766.60$



Классификация (Прогнозирование уровня дохода)

1. Adult Census Income

2. Предсказать, превышает ли доход \$50 тыс. в год ($income > 50K$)

3. Задания:

- загрузите данные, обработайте пропуски и категориальные признаки;
- обучите модель логистической регрессии;
- рассчитайте Accuracy, Precision и Recall;
- постройте матрицу ошибок.

Код программы:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score,
confusion_matrix

df = pd.read_csv("adult.csv", na_values="?")
df.dropna(inplace=True)

cat_columns = df.select_dtypes(include="object").columns

for col in cat_columns:
    encoder = LabelEncoder()
    df[col] = encoder.fit_transform(df[col])

X = df.drop("income", axis=1)
y = df["income"]

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(
    X_scaled,
    y,
    test_size=0.2,
    random_state=42,
    stratify=y
)

model = LogisticRegression(max_iter=10000)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

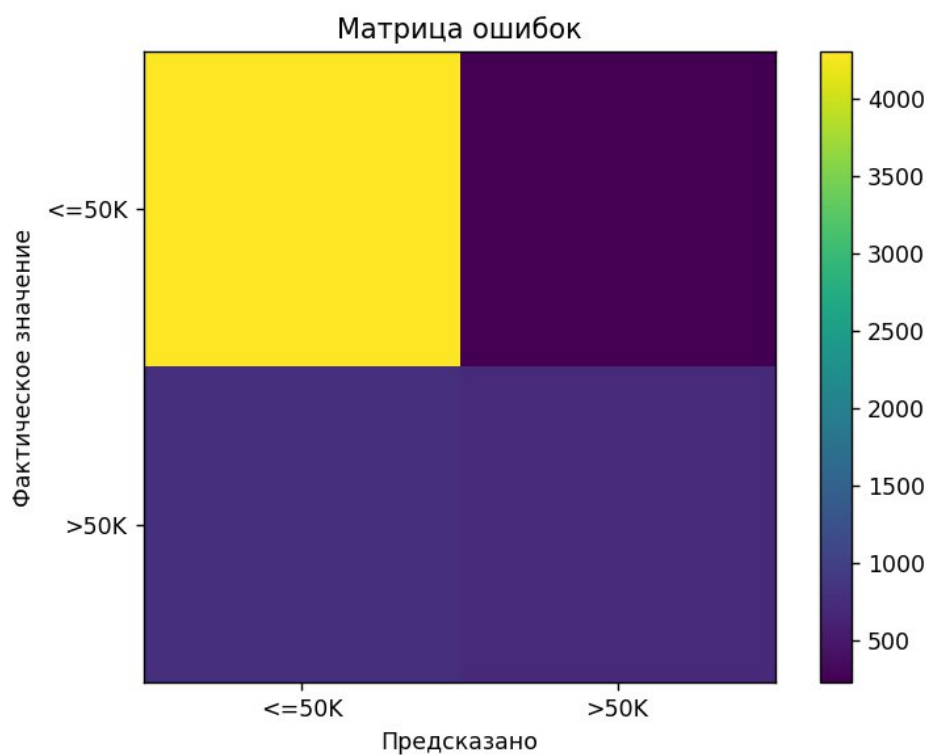
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)

print(f"Accuracy: {acc:.3f}")
print(f"Precision: {prec:.3f}")
print(f"Recall: {rec:.3f}")

cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(7, 5))
plt.imshow(cm)
plt.colorbar()
plt.xticks([0, 1], ["<=50K", ">50K"])
plt.yticks([0, 1], ["<=50K", ">50K"])
plt.xlabel("Предсказано")
plt.ylabel("Фактическое значение")
plt.title("Матрица ошибок")
plt.show()
```

```
Accuracy: 0.831
Precision: 0.755
Recall: 0.473
```



Вывод: изучили применение линейной и логистической регрессии для решения практических задач. Научились обучать модели, оценивать их качество с помощью соответствующих метрик и интерпретировать результаты.