

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**»

Выполнила:
Студентка 3 курса
Группы АС-65
Степанова Д. 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 seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, mean_absolute_error

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

X = df[["horsepower", "citympg", "enginesize", "carwidth", "carlength",
"carheight"]]
y = df["price"]

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

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

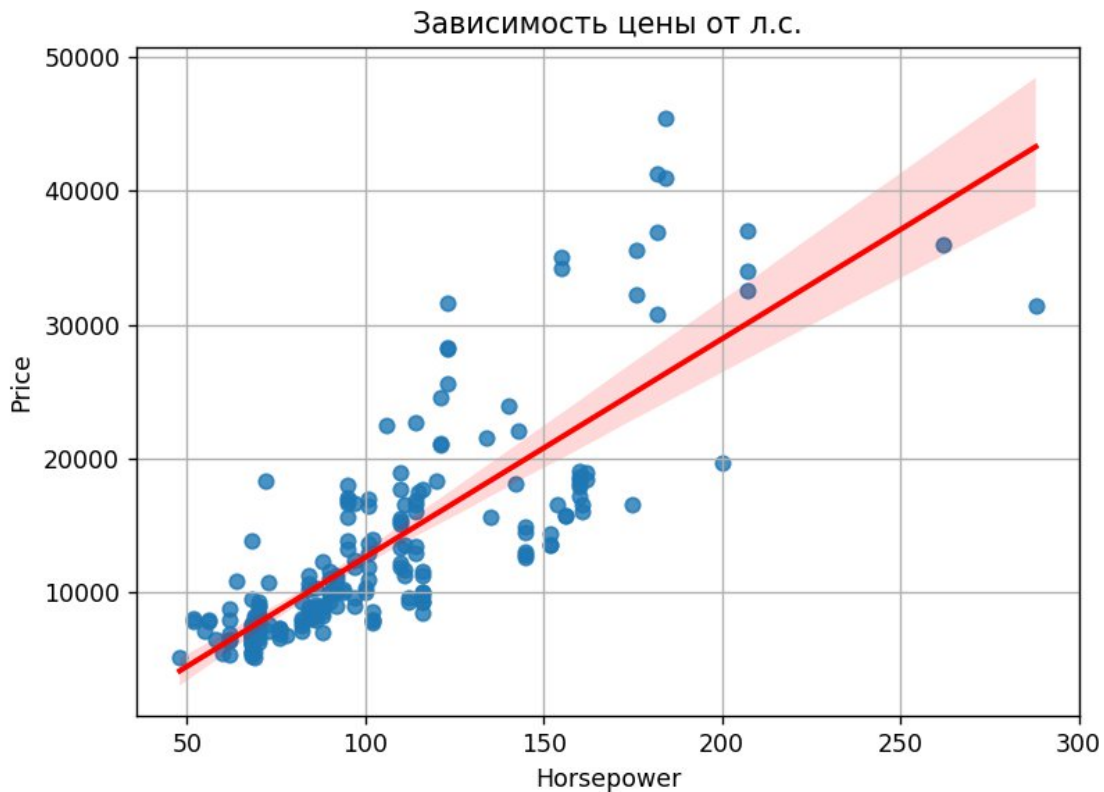
y_pred = model.predict(X_test)

r2 = r2_score(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)

print(f"R²: {r2:.3f}")
print(f"MAE: {mae:.2f}")

plt.figure(figsize=(7, 5))
sns.regplot(x=df["horsepower"], y=df["price"], line_kws={"color": "red"})
plt.title("Зависимость цены от л.с.")
plt.xlabel("Horsepower")
plt.ylabel("Price")
plt.grid(True)
plt.show()
```

$R^2: 0.813$
MAE: 2796.94



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
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
import seaborn as sns
import matplotlib.pyplot as plt

# 1
df = pd.read_csv("adult.csv")

df = df.replace('?', pd.NA)
df = df.dropna()

cat_cols = df.select_dtypes(include="object").columns
for col in cat_cols:
    le = LabelEncoder()
```

```

df[col] = le.fit_transform(df[col])

# 2
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)

# 3
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 (>50K): {prec:.3f}")
print(f"Recall (>50K): {rec:.3f}")

# 4
cm = confusion_matrix(y_test, y_pred)

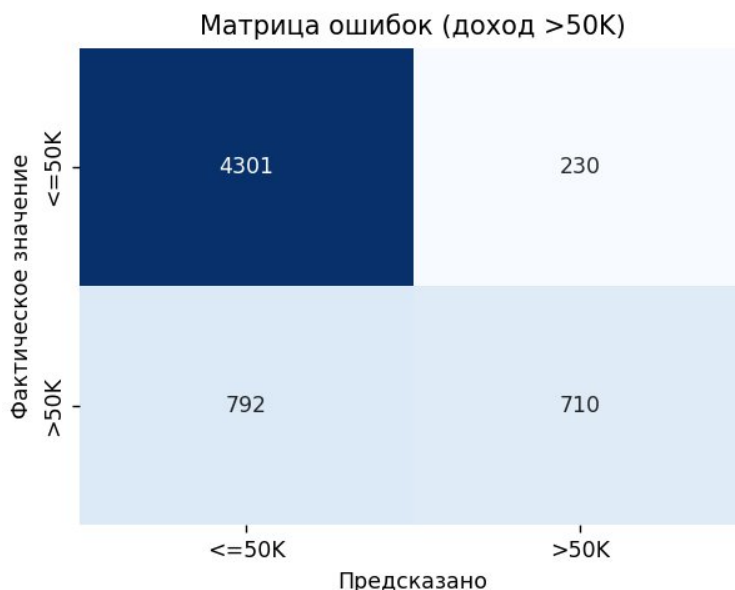
plt.figure(figsize=(5, 4))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", cbar=False,
            xticklabels=["<=50K", ">50K"], yticklabels=["<=50K", ">50K"])
plt.title("Матрица ошибок (доход >50K)")
plt.xlabel("Предсказано")
plt.ylabel("Фактическое значение")
plt.tight_layout()
plt.show()

```

```

Accuracy: 0.831
Precision (>50K): 0.755
Recall (>50K): 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.