МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению практических работ
по дисциплине «Анализ данных в эксплуатации программных систем» для студентов
направления подготовки 09.04.04 Программная инженерия
Направленность (профиль)
«Разработка, мониторинг и управление жизненным циклом инженерных систем»

Ставрополь 2026

Лабораторная работа №1

Парсинг и нормализация полуструктурированных логов (формат JSON / Logstash)

Цель работы:
Освоить методы извлечения структурированной информации из неструктурированных и полуструктурированных логов программных систем.

Задание:
Дан файл `webserver.log` (смесь строк в формате Apache Common Log Format и JSON-логов). Необходимо:

1. Распарсить временную метку, уровень логирования (INFO/WARN/ERROR), сообщение.
2. Преобразовать результат в DataFrame Pandas.
3. Сохранить нормализованные данные в `parsed_logs.csv`.

Теоретический минимум
Логи бывают:

- **Неструктурированные** – обычный текст (Apache log).
- **Полуструктурированные** – JSON, Key-Value.
- **Структурированные** – таблицы (CSV).

Для парсинга используют:

- Регулярные выражения (`re`)
- `json.loads()` для JSON-строк
- `pandas` для работы с таблицами

Ход выполнения (пошагово)

1. **Импорт библиотек**

python
```
import re
import json
import pandas as pd
from datetime import datetime
```

2. **Регулярное выражение для Apache Combined Log Format**

python
```
apache_pattern = re.compile(
    r'(?P<ip>\S+) \S+ \S+ \[(?P<time>.*?)\] "(?P<method>\S+) (?P<url>\S+) \S+" (?P<st
atus>\d{3}) (?P<size>\d+)'
```

```
)
```

3. Функция парсинга одной строки

python

```python
def parse_line(line):
    # Попытка распарсить JSON
    if line.strip().startswith('{'):
        try:
            data = json.loads(line)
            return {
                'timestamp': data.get('@timestamp'),
                'level': data.get('level', 'INFO'),
                'message': data.get('message'),
                'source': 'json'
            }
        except:
            pass

    # Попытка распарсить Apache-формат
    match = apache_pattern.match(line)
    if match:
        return {
            'timestamp': datetime.strptime(match.group('time'), '%d/%b/%Y:%H:%M:%S %z
'),
            'level': 'INFO' if match.group('status').startswith('2') else 'WARN' if m
atch.group('status').startswith('4') else 'ERROR',
            'message': f"{match.group('method')} {match.group('url')} status {match.g
roup('status')}",
            'source': 'apache'
        }

    return None
```

4. Обработка файла и создание DataFrame

python

```python
log_entries = []
with open('webserver.log', 'r', encoding='utf-8') as f:
    for line in f:
        parsed = parse_line(line)
        if parsed:
            log_entries.append(parsed)

df = pd.DataFrame(log_entries)
```

```
df.to_csv('parsed_logs.csv', index=False)
print(df.head())
```

Контрольные вопросы

1. Чем отличается полуструктурированный лог от неструктурированного?
2. Зачем нужна нормализация временных меток?
3. Как обработать случай, когда строка лога повреждена?

Список литературы (ГОСТ)

1. Маккинни, У. Python и анализ данных / У. Маккинни ; перевод с английского А. А. Слинкина. — Москва : ДМК Пресс, 2020. — 540 с. — ISBN 978-5-97060-903-5. (Глава 7: Очистка и подготовка данных)

Лабораторная работа №2

Выявление аномалий в метриках CPU/памяти с помощью скользящего окна и Z-оценки

Цель работы:
Научиться обнаруживать аномальные выбросы во временных рядах метрик производительности.

Задание:
Дан CSV-файл `metrics.csv` (временные метки, CPU_usage, Memory_usage).
Необходимо:

1. Загрузить данные, проверить на пропуски.
2. Реализовать метод скользящего окна (rolling window) для вычисления Z-score.
3. Отметить точки, где |Z| > 3, как аномалии.
4. Построить график с выделением аномалий.

Теоретический минимум

- **Z-оценка:** $Z=x-\mu\sigma Z=\sigma x-\mu$
Аномалия при $|Z|>3|Z|>3$ (правило трёх сигм).
- **Скользящее окно:** вместо глобальных $\mu,\sigma\mu,\sigma$ используют локальные (например, за последние 60 минут), чтобы учитывать тренд и сезонность.

Ход выполнения

1. Загрузка и первичный анализ

python

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df = pd.read_csv('metrics.csv', parse_dates=['timestamp'])
df.set_index('timestamp', inplace=True)
print(df.isnull().sum())
df.interpolate(method='linear', inplace=True)  # заполняем пропуски
```

2. Функция обнаружения аномалий через rolling Z-score

python

```python
def detect_anomalies_zscore(data, window=60, threshold=3):
    rolling_mean = data.rolling(window=window).mean()
    rolling_std = data.rolling(window=window).std()
    z_score = (data - rolling_mean) / rolling_std
    anomalies = np.abs(z_score) > threshold
    return anomalies, z_score
```

3. Применение к CPU и Memory

python

```python
anomalies_cpu, z_cpu = detect_anomalies_zscore(df['CPU_usage'], window=30)
anomalies_mem, z_mem = detect_anomalies_zscore(df['Memory_usage'], window=30)

df['CPU_anomaly'] = anomalies_cpu
df['Memory_anomaly'] = anomalies_mem
```

4. Визуализация

python

```python
fig, axes = plt.subplots(2, 1, figsize=(12, 8))

axes[0].plot(df.index, df['CPU_usage'], label='CPU usage')
axes[0].scatter(df.index[df['CPU_anomaly']], df['CPU_usage'][df['CPU_anomaly']],
                color='red', label='Anomaly', s=50)
axes[0].set_ylabel('CPU %')
axes[0].legend()

axes[1].plot(df.index, df['Memory_usage'], label='Memory usage')
axes[1].scatter(df.index[df['Memory_anomaly']], df['Memory_usage'][df['Memory_anomaly
']],
                color='red', label='Anomaly', s=50)
axes[1].set_ylabel('Memory %')
axes[1].legend()
```

```
plt.tight_layout()
plt.show()
```

Контрольные вопросы

1. Почему глобальный Z-score не подходит для данных с трендом?
2. Как выбрать размер окна скользящего среднего?
3. Какие альтернативы Z-score вы знаете для обнаружения аномалий?

Литература

1. Хан, Дж. Анализ данных : учебное пособие / Дж. Хан, М. Камбер. — Москва : Вильямс, 2016. — 590 с. (Глава 12: Обнаружение аномалий)

Лабораторная работа №3

Сегментация типов ошибок с использованием алгоритма K-means

Цель работы:
Применить кластеризацию для группировки сообщений об ошибках по смыслу (без ручного разбора).

Задание:
Дан файл `errors.log` (тысячи сообщений об исключениях). Необходимо:

1. Векторизовать тексты ошибок (TF-IDF).
2. Выполнить кластеризацию методом K-means.
3. Интерпретировать полученные кластеры как разные типы багов.
4. Визуализировать кластеры в 2D (PCA или t-SNE).

Теоретический минимум

- **K-means:** минимизирует внутрикластерное расстояние. Требует выбора K.
- **TF-IDF:** Term Frequency – Inverse Document Frequency. Превращает текст в числовой вектор.
- **Silhouette Score:** метрика качества кластеризации (от -1 до 1).

Ход выполнения

1. **Подготовка текстов**

```
python
import pandas as pd
```

```python
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

# Читаем логи (каждая строка – одно сообщение об ошибке)
with open('errors.log', 'r') as f:
    error_messages = [line.strip() for line in f if line.strip()]
```

2. Векторизация

```python
vectorizer = TfidfVectorizer(max_features=500, stop_words='english')
X = vectorizer.fit_transform(error_messages)
```

3. Подбор K через метод локтя и силуэт

```python
from sklearn.metrics import silhouette_score

inertias = []
sil_scores = []
K_range = range(2, 11)

for k in K_range:
    km = KMeans(n_clusters=k, random_state=42, n_init=10)
    labels = km.fit_predict(X)
    inertias.append(km.inertia_)
    sil_scores.append(silhouette_score(X, labels))

# Графики
fig, axes = plt.subplots(1, 2, figsize=(12, 4))
axes[0].plot(K_range, inertias, marker='o')
axes[0].set_title('Метод локтя')
axes[1].plot(K_range, sil_scores, marker='o')
axes[1].set_title('Силуэт')
plt.show()
```

4. Кластеризация с оптимальным K (например, 5)

```python
k = 5
km = KMeans(n_clusters=k, random_state=42, n_init=10)
clusters = km.fit_predict(X)

# Сохраняем результат
results = pd.DataFrame({'error_text': error_messages, 'cluster': clusters})
```

5. **Интерпретация кластеров – топ-10 слов в каждом кластере**

```python
terms = vectorizer.get_feature_names_out()
for cluster_id in range(k):
    center = km.cluster_centers_[cluster_id]
    top_indices = center.argsort()[-10:][::-1]
    top_words = [terms[i] for i in top_indices]
    print(f"\nКластер {cluster_id}: {', '.join(top_words)}")
    print(f"Пример ошибки: {results[results.cluster == cluster_id]['error_text'].iloc
[0]}")
```

6. **Визуализация PCA**

```python
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X.toarray())
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=clusters, cmap='viridis')
plt.title('Кластеры ошибок (PCA)')
plt.show()
```

Контрольные вопросы

1. Почему для текстов ошибок нельзя использовать евклидово расстояние напрямую?
2. Что такое «проклятие размерности» при TF-IDF?
3. Как определить, что кластер действительно соответствует одному типу бага?

Литература

1. Агрест, А. Введение в категорийный анализ данных / А. Агрест. — Москва : Финансы и статистика, 2018. — 420 с. (Глава 14: Кластерный анализ)

Лабораторная работа №4

Прогнозирование отказа узла системы по предикторам нагрузки (бинарная классификация)

Цель работы:
Построить модель машинного обучения, которая предсказывает отказ сервера за 5 минут до события.

Задание:
Дан датасет `azure_failure.csv` (метрики CPU, memory, disk I/O, network и бинарный флаг failure). Необходимо:

1. Выполнить балансировку классов (дисбаланс: отказов ~1%).
2. Обучить Random Forest Classifier.
3. Оценить качество по Precision, Recall, F1 и ROC-AUC.
4. Построить матрицу ошибок (Confusion Matrix).

Теоретический минимум

- **Дисбаланс классов:** Accuracy бесполезна. Нужны Precision (точность) и Recall (полнота).
- **Random Forest:** ансамбль решающих деревьев, устойчив к выбросам.
- **SMOTE:** синтез новых примеров минорного класса.

Ход выполнения

1. **Загрузка и разделение данных**

python
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score, C
onfusionMatrixDisplay
from imblearn.over_sampling import SMOTE

df = pd.read_csv('azure_failure.csv')
X = df.drop('failure', axis=1)
y = df['failure']

print(f"Доля отказов: {y.mean():.4f}")  # ~0.01

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state
=42, stratify=y)
```

2. **Применение SMOTE для балансировки**

python
```
smote = SMOTE(random_state=42)
X_train_bal, y_train_bal = smote.fit_resample(X_train, y_train)
print(f"После SMOTE: отказов = {y_train_bal.mean():.2f}")
```

3. **Обучение модели**

python
```
rf = RandomForestClassifier(n_estimators=100, max_depth=10, random_state=42)
```

```
rf.fit(X_train_bal, y_train_bal)
y_pred = rf.predict(X_test)
y_proba = rf.predict_proba(X_test)[:, 1]
```

4. **Оценка качества**

python

```
print(classification_report(y_test, y_pred))
print(f"ROC-AUC: {roc_auc_score(y_test, y_proba):.3f}")

# Матрица ошибок
ConfusionMatrixDisplay.from_estimator(rf, X_test, y_test, cmap='Blues')
plt.title('Confusion Matrix')
plt.show()
```

5. **Важность признаков**

python

```
importances = pd.Series(rf.feature_importances_, index=X.columns)
importances.sort_values(ascending=False).plot(kind='bar', figsize=(10,5))
plt.title('Важность признаков для предсказания отказа')
plt.show()
```

Контрольные вопросы

1. Почему в задаче предсказания отказов Recall важнее Precision?
2. Как SMOTE создаёт синтетические примеры?
3. Что показывает ROC-AUC = 0.99?

Литература

1. Хасти, Т. Введение в статистическое обучение / Т. Хасти, Р. Тибширани, Дж. Фридман. — Москва : ДМК Пресс, 2020. — 456 с. (Глава 8: Бэггинг и случайные леса)

Лабораторная работа №5

Автоматическое обнаружение атак (Web Attack) через анализ HTTP-логов методами NLP

Цель работы:
Научиться классифицировать HTTP-запросы на нормальные и вредоносные (SQL Injection, XSS) с помощью обработки естественного языка.

Задание:
Дан файл `http_logs.csv` (IP, метод, URL, параметры запроса, метка is_attack).
Необходимо:

1. Извлечь URL и параметры, объединить в один текст.
2. Применить TF-IDF и классификатор (SVM или Naive Bayes).
3. Оценить качество обнаружения атак.
4. Проанализировать ошибочно классифицированные запросы.

Ход выполнения

1. **Подготовка текстов запросов**

python
```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import LinearSVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report

df = pd.read_csv('http_logs.csv')
# Склеиваем URL и параметры в один текст
df['query_text'] = df['url'] + "?" + df['params'].fillna('')
print(df['query_text'].head())
```

2. **Векторизация и разделение**

python
```
vectorizer = TfidfVectorizer(max_features=2000, ngram_range=(1,2), lowercase=True)
X = vectorizer.fit_transform(df['query_text'])
y = df['is_attack']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state
=42)
```

3. **Обучение классификатора (SVM)**

python
```
svm = LinearSVC(class_weight='balanced', random_state=42, max_iter=2000)
svm.fit(X_train, y_train)
y_pred = svm.predict(X_test)

print(classification_report(y_test, y_pred))
```

4. **Анализ самых «подозрительных» слов**

python
```
feature_names = vectorizer.get_feature_names_out()
```

```python
coefs = svm.coef_.toarray()[0]
important = pd.DataFrame({'word': feature_names, 'coef': coefs})
important_attack = important.sort_values('coef', ascending=False).head(20)
important_normal = important.sort_values('coef').head(20)

print("Топ-10 признаков атаки:")
print(important_attack['word'].values)
```

5. **Проверка на реальных примерах (False Negatives)**

```python
misclassified = X_test[(y_pred != y_test).values]
mis_idx = np.where((y_pred != y_test).values)[0]
for idx in mis_idx[:5]:
    print(f"Истинный: {y_test.iloc[idx]}, Предсказанный: {y_pred[idx]}")
    print(f"Текст: {df.loc[X_test.index[idx], 'query_text'][:200]}\n")
```

Контрольные вопросы

1. Почему для HTTP-логов нужно использовать n-граммы, а не отдельные слова?
2. Чем опасны ложноположительные срабатывания (False Positive) в задаче обнаружения атак?
3. Как можно улучшить модель, используя знания о структуре SQL/XSS?

Литература

1. Чоллет, Ф. Глубокое обучение на Python / Ф. Чоллет. — Санкт-Петербург : Питер, 2018. — 400 с. (Глава 6: Обработка текста)

Лабораторная работа №6

Разработка интерактивного дашборда для SLA-отчетности в Grafana

Цель работы:
Создать дашборд для мониторинга соблюдения SLA (Service Level Agreement) в реальном времени.

Задание:
Используя данные о времени ответа API (файл `api_latency.csv`) и доступности сервиса (`uptime.csv`), построить в Grafana (или в Python + Dash) дашборд, содержащий:

1. График времени ответа (p50, p95, p99) за последние 24 часа.
2. Heatmap ошибок по часам (количество HTTP 5xx).
3. Индикатор соблюдения SLA (доступность 99.9%).
4. Таблицу топ-5 самых медленных эндпоинтов.

Теоретический минимум

- **SLA:** обычно 99.9% доступности = допустимо 43 минуты простоя в месяц.
- **Перцентили:** p95 важнее среднего, так как среднее скрывает выбросы.
- **Grafana:** подключение к Prometheus/InfluxDB/CSV через плагин.

Ход выполнения (в Python + Plotly Dash, если нет доступа к серверу Grafana)

1. **Загрузка и агрегация данных**

python
```python
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
from dash import Dash, dcc, html, Input, Output
from datetime import datetime, timedelta

latency = pd.read_csv('api_latency.csv', parse_dates=['timestamp'])
uptime = pd.read_csv('uptime.csv', parse_dates=['timestamp'])

# Добавляем час для heatmap
latency['hour'] = latency['timestamp'].dt.hour
latency['date'] = latency['timestamp'].dt.date
```

2. **Расчёт перцентилей**

python
```python
percentiles = latency.groupby('endpoint')['latency_ms'].agg(
    p50=lambda x: x.quantile(0.5),
    p95=lambda x: x.quantile(0.95),
    p99=lambda x: x.quantile(0.99)
).reset_index()
```

3. **Heatmap ошибок (5xx)**

python
```python
errors = latency[latency['status_code'] >= 500]
error_heat = errors.groupby(['date', 'hour']).size().reset_index(name='error_count')
heat_fig = px.density_heatmap(error_heat, x='hour', y='date', z='error_count',
                              title='Тепловая карта ошибок 5xx')
```

4. **Расчёт доступности SLA**

python

```
total_minutes = (uptime['timestamp'].max() - uptime['timestamp'].min()).total_seconds
() / 60
downtime_minutes = uptime['is_up'].value_counts().get(0, 0)  # предполагаем, что 1 =
работает
availability = (total_minutes - downtime_minutes) / total_minutes * 100
sla_status = "SLA соблюдается" if availability >= 99.9 else "SLA нарушено"
```

5. Сборка Dash-приложения

python

```
app = Dash(__name__)

app.layout = html.Div([
    html.H1(f"Дашборд SLA: {sla_status} (Доступность: {availability:.3f}%)"),
    dcc.Graph(figure=heat_fig),
    dcc.Graph(figure=px.line(latency, x='timestamp', y='latency_ms', color='endpoint'
,
                             title='Латентность API по времени')),
    dcc.Graph(figure=px.bar(percentiles, x='endpoint', y=['p50', 'p95', 'p99'],
                             title='Перцентили времени ответа', barmode='group')),
    html.H3("Топ-5 медленных эндпоинтов (p95)"),
    html.Table(percentiles.nlargest(5, 'p95')[['endpoint', 'p95']].to_dict('records')
)
])

if __name__ == '__main__':
    app.run(debug=True)
```

Контрольные вопросы

1. Почему для SLA используют p99, а не среднее арифметическое?
2. Что означает доступность «99.9%» в пересчёте на минуты в месяц?
3. Какой показатель важнее для бизнеса: latency p95 или количество ошибок 5xx?

Литература

1. Ван дер Алст, В. Анализ процессов: обнаружение, проверка и улучшение процессов на основе данных событий / В. ван дер Алст. — Москва : ДМК Пресс, 2019. — 412 с. (Глава 10: Мониторинг соответствия)

Итоговое оформление практикума (требования к сдаче)

Каждая лабораторная работа должна быть оформлена в виде **Jupyter Notebook** с:

- Титульным листом (ФИО, группа, номер работы).
- Кратким теоретическим введением.
- Кодом с комментариями.
- Выводами по полученным результатам.
- Ответами на контрольные вопросы.

Файлы для скачивания (датасеты):

- `webserver.log` – сгенерировать с помощью `fake-log-generator`.
- `metrics.csv` – взять с Kaggle: Server Metrics.
- `errors.log` – использовать LogHub Error Logs.
- `azure_failure.csv` – датасет Microsoft Azure Failure Prediction.
- `http_logs.csv` – CSIC 2010 HTTP Dataset (атаки).
- `api_latency.csv`, `uptime.csv` – синтетические, сгенерированные скриптом.

Критерии оценки каждой ЛР:

- 0 баллов – не сдана.
- 3 балла – код работает, но выводы отсутствуют.
- 4 балла – код работает, выводы есть, но есть ошибки в интерпретации.
- 5 баллов – код оптимален, выводы аналитические, ответы на вопросы верные.

Итоговая аттестация по практикуму:
Студент защищает все 6 работ в виде **единого портфолио** (архив с ноутбуками и скриншотами дашборда). Дополнительный бонус – если код выложен на GitHub и оформлен как `README.md` с примерами графиков.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по организации и проведению самостоятельной работы по дисциплине «Анализ данных в эксплуатации программных систем» для студентов направления подготовки
09.04.04 Программная инженерия Направленность (профиль) «Разработка, мониторинг и управление жизненным циклом инженерных систем»

Ставрополь 2026

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

1. Цель и задачи освоения дисциплины (модуля)

Формирование у студентов компетенций по применению методов интеллектуального анализа данных (Data Mining) и машинного обучения для решения задач мониторинга, диагностики и прогнозирования надежности программных систем в процессе их эксплуатации.

Задачи освоения дисциплины:

– Изучить метрики качества программного обеспечения и методы сбора эксплуатационных логов.

– Освоить методы визуализации данных для анализа аномалий и «узких мест» производительности.

– Научиться применять регрессионный и кластерный анализ для прогнозирования отказов ПО.

– Изучить основы анализа текстов логов (Text Mining) для автоматической классификации ошибок.

2. Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Анализ данных в эксплуатации программных систем» относится к дисциплинам обязательной части.

3. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесённых с планируемыми результатами освоения образовательной программы

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
УК-1. Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий	ИД-1$_{УК-1}$ осуществляет разработку стратегии действий для выявления и решения проблемной ситуации.	осуществляет разработку стратегии действий для выявления и решения проблемной ситуации.
	ИД-2 $_{УК-1}$ вырабатывает стратегию действий, принимает конкретные решения для ее реализации.	вырабатывает стратегию действий, принимает конкретные решения для ее реализации.
	ИД-3 $_{УК-1}$ ставит цели, определяет способы ее достижения, разрабатывает стратегий действий.	ставит цели, определяет способы ее достижения, разрабатывает стратегий действий.
ОПК-7. Способен применять при решении профессиональных задач методы и средства получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе,	ИД-1 $_{ОПК-7}$. использует принципы построения математических моделей процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений;	использует принципы построения математических моделей процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений;
	ИД-2 $_{ОПК-7.}$ строит математические модели для	строит математические модели для реализации

в глобальных компьютерных сетях	реализации успешного функционирования распределенных информационных систем и систем поддержки принятия решений	успешного функционирования распределенных информационных систем и систем поддержки принятия решений

4. Объем учебной дисциплины и формы контроля *

Объем занятий: всего: 5 з.е. 180 ак.ч.	ОФО, в астр. часах
Контактная работа:	54
Лекции/из них практическая подготовка	18
Лабораторных работ/из них практическая подготовка	36
Практических занятий/из них практическая подготовка	
Самостоятельная работа	126
Формы контроля	
Экзамен	
Зачет	
Зачет с оценкой	
Расчетно-графические работы	
Курсовые работа	
Контрольные работы	

* Дисциплина (модуль) предусматривает применение электронного обучения, дистанционных образовательных технологий

5. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием количества часов и видов занятий

№	Раздел (тема) дисциплины и краткое содержание	Формируемые компетенции, индикаторы	очная форма			
			Контактная работа обучающихся с преподавателем /из них в форме практической подготовки, часов			Самостоятельная работа, часов
			Лекции	Практические занятия	Лабораторные работы	

1	Раздел 1. Введение в DataOps Понятие телеметрии ПО. Разница между логированием, метриками и трассировкой. Обзор стека ELK.	УК-1, ОПК-7	2		6	21
2	Раздел 2. Анализ временных рядов Стационарность, тренды, сезонность. Методы сглаживания. Корреляционный анализ (Пирсон, Спирмен).	УК-1, ОПК-7	4		6	21
3	Раздел 3. Кластеризация отказов Метрики расстояний. K-means. DBSCAN для шумных данных. Оценка качества кластеризации (Silhouette).	УК-1, ОПК-7	2		6	21
4	Раздел 4. Предиктивная аналитика Задачи классификации: Decision Trees, Random Forest. Оценка Precision/Recall. Дисбаланс классов.	УК-1, ОПК-7	4		6	21
5	Раздел 5. Анализ текстов логов Методы векторизации: Bag of Words, TF-IDF. Поиск шаблонов логов (Log parsing).	УК-1, ОПК-7	4		6	21
6	Раздел 6. Визуализация и BI Принципы построения дашбордов. OLAP-кубы. Инструменты: Grafana, Tableau, matplotlib.	УК-1, ОПК-7	2		6	21
	ИТОГО за 3 семестр		18		36	126
	ИТОГО		18		36	126

1. ОБЩАЯ ХАРАКТЕРИСТИКА САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Все виды самостоятельной работы по дисциплине " Анализ данных в эксплуатации программных систем " приведены в таблице «Технологическая карта самостоятельной работы». В основу самостоятельной работы студентов по дисциплине положено конспектирование лекций и рекомендуемых источников, подготовка к компьютерному тестированию знаний, оформление и подготовка к защите отчетов по лабораторным работам, выполнению разноуровневых индивидуальных заданий в ходе лабораторных работ. По каждому виду самостоятельной работы предусмотрены определённые формы отчетности, которые студенты предъявляют преподавателю в ходе лабораторных занятий или в часы индивидуальных консультаций.

Таблица 1 - Технологическая карта самостоятельной работы.

Код оцениваемой компетенции, индикаторов	Этап формирования компетенции (№ темы) (в соответствии с рабочей программой дисциплины)	Средства и технологии оценки	Вид контроля, аттестация (текущий/промежуточный)	Тип контроля (устный, письменный или с использованием технических средств)	Наименование оценочного средства
1 семестр					

УК-1	Темы № 1–6	Собеседование, оценка ответов по вопросам заданных тем	текущий	устный	вопросы для собеседования
ОПК-7	Темы № 1–6	Зачетное задание-выполнение индивидуальных(групповых) заданий из практических работ	текущий	письменный	Комплект заданий

Для успешного освоения дисциплины, необходимо выполнить указанные в таблице 1 виды самостоятельной работы, используя рекомендуемые источники информации.

8.1.1. Перечень основной литературы:

1. Хасти, Т. Введение в статистическое обучение с примерами на языке R / Т. Хасти, Р. Тибширани, Дж. Фридман ; перевод с английского А. А. Слинкина. — 2-е изд. — Москва : ДМК Пресс, 2020. — 456 с. — ISBN 978-5-97060-861-8.

2. Ван дер Алст, В. Анализ процессов: обнаружение, проверка и улучшение процессов на основе данных событий / В. ван дер Алст ; перевод с английского Д. А. Беликова. — Москва : ДМК Пресс, 2019. — 412 с. — ISBN 978-5-97060-816-8.

3. Маккинни, У. Python и анализ данных / У. Маккинни ; перевод с английского А. А. Слинкина. — Москва : ДМК Пресс, 2020. — 540 с. — ISBN 978-5-97060-903-5.

8.1.2. Перечень дополнительной литературы:

4. Бринк, Х. Машинное обучение / Х. Бринк, Дж. Ричардс, М. Феверолф ; перевод с английского В. С. Яценкова. — Санкт-Петербург : Питер, 2017. — 336 с. — ISBN 978-5-496-02992-0.

5. Лурье, К. М. Анализ данных в программной инженерии: методы и инструменты / К. М. Лурье. — Новосибирск : НГТУ, 2022. — 148 с. — ISBN 978-5-7782-4689-1..

Особенности освоения дисциплины (модуля) лицами с ограниченными возможностями здоровья

Обучающимся с ограниченными возможностями здоровья предоставляются специальные учебники, учебные пособия и дидактические материалы, специальные технические средства обучения коллективного и индивидуального пользования, услуги ассистента (помощника), оказывающего
обучающимся необходимую техническую помощь, а также услуги сурдопереводчиков и тифлосурдопереводчиков.

Освоение дисциплины (модуля) обучающимися с ограниченными возможностями здоровья может быть организовано совместно с другими обучающимися, а также в отдельных группах.

Освоение дисциплины (модуля) обучающимися с ограниченными возможностями здоровья осуществляется с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья.

В целях доступности получения высшего образования по образовательной программе лицами с ограниченными возможностями здоровья при освоении дисциплины (модуля) обеспечивается:

1) для лиц с ограниченными возможностями здоровья по зрению:

- присутствие ассистента, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе, записывая под диктовку),
- письменные задания, а также инструкции о порядке их выполнения оформляются увеличенным шрифтом,
- специальные учебники, учебные пособия и дидактические материалы (имеющие крупный шрифт или аудиофайлы),
- индивидуальное равномерное освещение не менее 300 люкс,
- при необходимости студенту для выполнения задания предоставляется увеличивающее устройство;

2) для лиц с ограниченными возможностями здоровья по слуху:

- присутствие ассистента, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе, записывая под диктовку),
- обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости обучающемуся предоставляется звукоусиливающая аппаратура индивидуального пользования;
- обеспечивается надлежащими звуковыми средствами воспроизведения информации;

3) для лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата (в том числе с тяжелыми нарушениями двигательных функций верхних конечностей или отсутствием верхних конечностей):

- письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются ассистенту;
- по желанию студента задания могут выполняться в устной форме.

Особенности реализации дисциплины с применением дистанционных образовательных технологий и электронного обучения

Согласно части 1 статьи 16 Федерального закона от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации» под *электронным обучением* понимается организация образовательной деятельности с применением содержащейся в базах данных и используемой при реализации образовательных программ информации и обеспечивающих ее обработку информационных технологий, технических средств, а также информационно-телекоммуникационных сетей, обеспечивающих передачу по линиям связи указанной информации, взаимодействие обучающихся и педагогических работников. Под *дистанционными образовательными технологиями* понимаются образовательные технологии, реализуемые в основном с применением информационно- телекоммуникационных сетей при опосредованном (на расстоянии) взаимодействии обучающихся и педагогических работников.

Реализация дисциплины может быть осуществлена с применением дистанционных образовательных технологий и электронного обучения полностью или частично. Компоненты УМК дисциплины (рабочая программа дисциплины, оценочные и методические материалы, формы аттестации), реализуемой с применением дистанционных образовательных технологий и электронного обучения, содержат указание на их использование.

При организации образовательной деятельности с применением дистанционных образовательных технологий и электронного обучения могут предусматриваться асинхронный и синхронный способы осуществления взаимодействия участников образовательных отношений посредством информационно-телекоммуникационной сети «Интернет».

При применении дистанционных образовательных технологий и электронного обучения в расписании по дисциплине указываются: способы осуществления взаимодействия участников образовательных отношений посредством информационно-телекоммуникационной сети «Интернет» (ВКС-видеоконференцсвязь, ЭТ – электронное тестирование); ссылки на электронную информационно- образовательную среду СКФУ, на образовательные платформы и ресурсы иных организаций, к которым предоставляется открытый доступ через информационно-телекоммуникационную сеть
«Интернет»; для синхронного обучения - время проведения онлайн-занятий и преподаватели; для асинхронного обучения - авторы онлайн-курсов.

При организации промежуточной аттестации с применением дистанционных образовательных технологий и электронного обучения используются Методические рекомендации по применению технических средств, обеспечивающих объективность результатов при проведении промежуточной и государственной итоговой аттестации по образовательным программам высшего образования - программам бакалавриата, программам специалитета и программам магистратуры с применением дистанционных образовательных технологий (Письмо Минобрнаки России от 07.12.2020 г. № МН- 19/1573-АН "О направлении методических рекомендаций").

Реализация дисциплины с применением электронного обучения и дистанционных образовательных технологий осуществляется с использованием электронной информационно-образовательной среды СКФУ, к которой обеспечен доступ обучающихся через информационно-телекоммуникационную сеть «Интернет», или с использованием ресурсов иных организаций, в

том числе платформ, предоставляющих сервисы для проведения видеоконференций, онлайн-встреч и дистанционного обучения (Bigbluebutton, Microsoft Teams, а также с использованием возможностей социальных сетей для осуществления коммуникации обучающихся и преподавателей.

Учебно-методическое обеспечение дисциплины, реализуемой с применением электронного обучения и дистанционных образовательных технологий, включает представленные в электронном виде рабочую программу, учебно-методические пособия или курс лекций, методические указания к выполнению различных видов учебной деятельности обучающихся, предусмотренных дисциплиной, и прочие учебно-методические материалы, размещенные в информационно-образовательной среде СКФУ.

2. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ИЗУЧЕНИЮ ТЕОРЕТИЧЕСКОГО МАТЕРИАЛА

При самостоятельном изучении тем лекционных занятий и заданной литературы студентам рекомендуется:

- в день проведения занятий проработать, изученный на занятии учебный материал по конспекту. При этом необходимо выделить вопросы, на которые следует обратить большее внимание при дальнейшем изучении текущей темы. После проработки учебного материала необходимо ответить на, рекомендуемые контрольные вопросы;

-с целью качественного закрепления изученного материала в последующем следует более детально проработать учебный материал с использованием рекомендованной литературы. Студентам рекомендуется выделить вопросы, требующие более детальной проработки с помощью преподавателя;

- накануне проведения лекции студенту рекомендуется закрепить ранее изученный теоретический материал, подготовить вопросы, требующие уточнения у преподавателя.

Рекомендации по организации работы с литературой

Для овладения, закрепления и систематизации знаний студентам рекомендуется самостоятельная работа с рекомендованной литературой и конспектом лекций. При самостоятельной работе с рекомендованной литературой студентам рекомендуется проводить конспектирование учебного материала, изложенного в рекомендованных источниках.

Конспектирование источников проводится при детальном изучении темы, при этом студентам рекомендуется:

- дополнять конспект лекционного занятия путем его расширения по наиболее важным вопросам, рассмотренным на занятии. Конспектирование источников заключается в дополнении конспекта следующими положениями:

-основные определения и их физический смысл;

- основные математические соотношения с раскрытием их физического смысла;
- выводы по учебным вопросам, изучаемой темы занятий.

Контроль этого вида самостоятельной работы осуществляется на учебных занятиях путем проверки преподавателем конспекта лекций и собеседования.

При конспектировании источников студенты должны исходить из рационального объема, конспектируемого материала. Конспект должен быть кратким и понятным при его использовании после изучения текущей темы.

При конспектировании источников рекомендуется отдельные рисунки, наиболее важные их фрагменты, наиболее важные формулы выделять цветом. При ведении конспекта на занятии рекомендуется заполнять две трети страницы листа тетради по вертикали, а свободную часть использовать при дополнении конспекта, по изучаемому вопросу.

Для успешного освоения дисциплины, необходимо самостоятельно детально изучить

представленные темы по рекомендуемым источникам информации.
Конспектирование источников оценивается удовлетворительно и неудовлетворительно. Работа по конспектированию источников оценивается удовлетворительно если:
- конспект имеет достаточный объем детализации по, изучаемой теме, обеспечивающий заданный уровень освоения теоретического материала дисциплины;
- конспект оформлен аккуратно, изучаемый материал изложен последовательно в соответствии с порядком изучения дисциплины, содержит обобщающие выводы по каждой теме.

Работа по конспектированию источников оценивается неудовлетворительно, если не выполнены требования на оценку удовлетворительно.

3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

При выполнении практикума по дисциплине для достижения базового уровня сформированности компетенций студент должен уметь применять базовые знания по разделам дисциплины, владеть навыками самостоятельного решения учебных задач базового уровня.
При выполнении практикума по дисциплине для достижения повышенного уровня сформированности компетенций студент должен уметь применять повышенные знания по разделам дисциплины, владеть навыками самостоятельного решения учебных задач повышенного уровня.

Защита практической работы осуществляется путем собеседования студента с преподавателем. При собеседовании студент представляет на проверку отчет по практической работе. Собеседование со студентом, претендующим на оценку «отлично» проводится по вопросам и практическим заданиям повышенного уровня обучения. Собеседование со студентом, не претендующим на оценку «отлично» проводится по вопросам и практическим заданиям базового уровня обучения.

4. ВОПРОСЫ ДЛЯ СОБЕСЕДОВАНИЯ

ПО ДИСЦИПЛИНЕ Анализ данных в эксплуатации программных систем

Базовый уровень

1. Понятие телеметрии ПО.
2. Разница между логированием, метриками и трассировкой.
3. Обзор стека ELK.
4. Стационарность, тренды, сезонность.
5. Методы сглаживания.
6. Корреляционный анализ (Пирсон, Спирмен).
7. Метрики расстояний. K-means.
8. DBSCAN для шумных данных.
9. Оценка качества кластеризации (Silhouette).
10. Задачи классификации: Decision Trees, Random Forest.
11. Оценка Precision/Recall.
12. Дисбаланс классов.
13. Методы векторизации: Bag of Words, TF-IDF.
14. Поиск шаблонов логов (Log parsing).
15. Принципы построения дашбордов.
16. OLAP-кубы.
17. Инструменты: Grafana, Tableau, matplotlib.

Повышенный уровень

1. Трехстолпная модель наблюдаемости (логи, метрики, трассировка) против непрерывного профилирования (eBPF). Какие классы проблем не диагностируются тремя столпами? Как профилирование их решает?
2. Проблема высокой кардинальности метрик. Почему добавление user_id в метрику типа

гистограммы вызывает проблемы в Prometheus, но допустимо в VictoriaMetrics? Как дизайн хранилища влияет на это ограничение?

3. Стек ELK (Elasticsearch, Logstash, Kibana): при каких объемах (ТВ/сутки) модель «индексируй всё» становится экономически неэффективной? Сравните с архитектурой Loki (Grafana Labs).

4. Сэмплирование в распределенной трассировке: head-based vs. tail-based. Приведите сценарий, где head-based (1% запросов) пропустит критическую аномалию, а tail-based — нет. Как это влияет на архитектуру OpenTelemetry Collector?

5. Структурированное логирование (JSON) vs. неструктурированное. Почему для корреляции с метриками и трассировками предпочтителен подход «лог как набор полей»?

6. Ложная корреляция в нестационарных рядах. Коэффициент Пирсона между продажами и числом дождливых дней равен 0.7. Почему это может быть артефактом? Как удаление тренда (разностное преобразование) меняет интерпретацию?

7. Сравнение фильтра Ходрика–Прескотта (HP filter) и экспоненциального сглаживания (ETS) для выделения тренда в данных с выбросами. Почему простое скользящее среднее не подходит, и как его улучшает медианное сглаживание с динамическим окном?

8. Стационарность в ML-прогнозировании. Вы строите LSTM для прогноза CPU на 24 часа вперед. Ряд имеет недельную сезонность и тренд. Почему подача сырого ряда может привести к провалу? Какие методы приведения к стационарности (разность, Box–Cox, STL) предпочтительны для нейросетей и почему?

9. Проклятие размерности для k-means. У вас 1000 логов, векторизованных TF IDF, размерность — 5000. Почему евклидово расстояние становится неинформативным? Какие метрики (косинусное, Манхэттен) менее чувствительны к высокой размерности?

10. DBSCAN на зашумленных логах ошибок. Кластеризация дает один огромный кластер «шум» (метка -1). Как, не меняя eps и min_samples, модифицировать входные данные (предварительная кластеризация, UMAP) чтобы DBSCAN нашел структуру?

11. Коэффициент силуэта (Silhouette). В каком случае он может быть высоким, но кластеры не имеют бизнес-смысла? Приведите пример.

12. Чувствительность DBSCAN к глобальному параметру eps в датасетах с кластерами разной плотности. Как этот недостаток исправляет алгоритм OPTICS?

13. Decision Tree vs. Random Forest на временных рядах. Вы определяете: всплеск RPS — аномалия или плановая нагрузка? Почему отдельное дерево может переобучиться при наличии автокорреляции? Как Random Forest с бутстрапом по блокам (а не по строкам) решает эту проблему?

14. Компромисс Precision и Recall для редких событий. Дисбаланс 1000:1 (норма/отказ диска). Почему Accuracy бесполезна? В каком сценарии высокая Precision (>0.99) вредна, а Recall должен быть максимальным даже ценой ложных тревог? Примеры из банкинга и авиационной телеметрии.

15. Методы борьбы с дисбалансом классов. Сравните SMOTE (генерация синтетических точек) и Focal Loss. Почему простое взвешивание классов (class_weight) может ухудшить качество на шумных данных? Какая проблема возникает при применении SMOTE к категориальным признакам?

16. TF IDF и Bag of Words для шаблонизации логов. Лог: User 12345 logged in from 192.168.1.1. Почему оба метода плохи для поиска шаблона? Как алгоритм Drain использует длину токена и тип данных (число/хекстет) для замены переменных частей на *?

17. Семантическая векторизация логов. Предложите гибрид: Log Parser (Drain) → шаблон → sentence BERT. Когда это оправдано (распределенные системы, микросервисы), а когда — избыточно?

18. Grafana + Prometheus vs. Tableau + ClickHouse (OLAP). Сравните архитектурные компромиссы. В каком сценарии вы выберете первую связку, а в каком — вторую? Почему в реальном времени Tableau проигрывает, а в многомерном анализе по 50 измерениям Grafana становится непригодной?

19. Предрасчет агрегатов в OLAP-кубе. Какие агрегаты (SUM, COUNT, AVG, STDDEV) можно предрассчитать, а какие нет? Объясните проблему «раздувания куба» (combinatorial explosion) при наличии иерархий (год→месяц→день) и как её решают растровые индексы (bitmap index).
20. Манипуляции с matplotlib в production. Почему plt.plot() с 1000 временных рядов непригоден? Как библиотеки с WebGL (plotly, bokeh + datashader) решают проблему производительности? Когда «пустой» дашборд с высоким разрешением вводит в заблуждение, и как LOESS (регрессионное сглаживание) это исправляет?

5. КРИТЕРИИ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с вопросами не только базового, но и повышенного уровня, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение.
Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопросы базового уровня, правильно применяет теоретические положения при решении практических вопросов базового уровня, владеет необходимыми навыками и приемами их выполнения.
Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала базового уровня, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала.
Оценка «неудовлетворительно» выставляется студенту, не знающему значительной части программного материала, допускающему грубые ошибки.

Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Процедура проведения данного оценочного мероприятия включает в себя собеседование.
Предлагаемые студенту задания позволяют проверить компетенции УК-1, ОПК-7. При проверке задания оценивается:
– объем и глубина усвоенного программного материала;
–последовательность, четкость и логическая стройность его изложения.