

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»

**УТВЕРЖДАЮ**

Зав. кафедрой ИСТ

\_\_\_\_\_ Данилов О.Ф.

« \_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

дисциплины:

направление подготовки:

направленность (профиль):

форма обучения:

**Машинное обучение**

**09.04.04 Программная инженерия**

**Программная инженерия систем искусственного интеллекта  
очная**

Фонд оценочных средств рассмотрен на заседании кафедры Интеллектуальных систем и технологий для направления 09.04.04 Программная инженерия направленность (профиль) Программная инженерия систем искусственного интеллекта

## 1. Формы аттестации по дисциплине

1.1. Форма промежуточной аттестации: зачет.

Способ проведения промежуточной аттестации: тестирование.

1.2. Формы текущей аттестации:

Таблица 1.2.1

№ п/п	Форма обучения
	ОФО
1	Устный опрос
2	Защита индивидуальной домашней работы.

## 2. Результаты обучения по дисциплине, подлежащие проверке при проведении текущей и промежуточной аттестации

Таблица 2.1

№ п/п	Структурные элементы дисциплины		Код результата обучения по дисциплине	Оценочные средства	
	Номер раздела	Дидактические единицы (предметные темы)		Текущая аттестация	Промежуточная аттестация
1	2	3	4	5	6
1	3	Подготовка данных.	31, 32, У1, В1	Устный опрос. Защита индивидуальной домашней работы.	Тест.
2	4	Информационное обучение.	31, 32, У1, У2, У3, В1, В2	Устный опрос. Защита индивидуальной домашней работы.	Тест.
3	5	Обучение на основе сходства.	31, 32, У1, У2, В1, В2	Устный опрос. Защита индивидуальной домашней работы.	Тест.
4	6	Вероятностное обучение.	31, 32, У1, У2, В1, В2	Устный опрос. Защита индивидуальной домашней работы.	Тест.
5	7	Обучение на основе ошибок.	31, 32, У1, У2, В1, В2	Устный опрос. Защита индивидуальной домашней работы.	Тест.
6	8	Оценка моделей.	31, 32, У1, У2, В1, В2	Устный опрос. Защита индивидуальной домашней работы.	Тест.

## 3. Фонд оценочных средств

3.1. Фонд оценочных средств, позволяющие оценить результаты обучения по дисциплине, включает в себя оценочные средства для текущей аттестации и промежуточной аттестации.

3.2. Фонд оценочных средств для текущей аттестации включает:

- Вопросы к устному опросу – 10 шт. (Приложение 1);
- комплект заданий для индивидуальной домашней работы – 10 шт. (Приложение 2);

3.3. Фонд оценочных средств для промежуточной аттестации включает:

- комплект тестовых заданий к экзамену по дисциплине «Машинное обучение» – 30 шт. (Приложение 3).

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»

**Вопросы к устному опросу  
по дисциплине «Машинное обучение»**

1. Жизненный цикл проекта по аналитическому прогнозированию: CRISP-DM.
2. Инструменты для аналитического прогнозирования.
3. Типы данных признаков.
4. Классификация задач машинного обучения.
5. Обработка пропущенных значений.
6. Обучение на основе сходства.
7. Обучение на основе ошибок. Линейная регрессия.
8. Кластерный анализ.
9. Перекрестная проверка данных.
10. Работа с конвейерами в библиотеке python scikit-learn.

**Критерии оценки за аттестацию**

**17-20 баллов** выставляется обучающемуся, если обучающийся в полном объеме раскрыл вопрос как на теоретическом, так и на практическом уровне, с соблюдением необходимой последовательности изложения аргументов, а также ответил на все дополнительные вопросы;

**10-16 баллов** выставляется обучающемуся, если выполнены все требования, соответствующие максимальной оценке (10 баллов), но было допущено два-три недочета или одна грубая ошибка;

**4-9 баллов** выставляется обучающемуся, если вопрос раскрыт не полностью, допущены две грубые ошибки;

**0-3 баллов** выставляется обучающемуся, если обучающийся не ответил на основной вопрос и на все дополнительные вопросы.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»

**Комплект заданий для индивидуальной домашней работы по дисциплине  
«Машинное обучение»**

*Задания для индивидуальной домашней работы*

1. Используя датасет Boston (from sklearn.datasets import load\_boston) необходимо решить задачу регрессии методом наименьших квадратов.
2. Используя датасет Boston (from sklearn.datasets import load\_boston) необходимо решить задачу методом гребневой регрессии.
3. Используя датасет Boston (from sklearn.datasets import load\_boston) необходимо решить задачу регрессии методом LASSO.
4. Используя датасет Boston (from sklearn.datasets import load\_boston) необходимо решить задачу регрессии методом эластичной сети.
5. Используя датасет Boston (from sklearn.datasets import load\_boston) необходимо решить задачу регрессии методом решающих деревьев.
6. Используя датасет Digits (from sklearn.datasets import load\_digits) необходимо решить задачу регрессии методом решающих деревьев.
7. Используя датасет Digits (from sklearn.datasets import load\_digits) необходимо решить задачу классификации методом опорных векторов.
8. Используя датасет Digits (from sklearn.datasets import load\_digits) необходимо решить задачу классификации методом случайных деревьев.
9. Используя датасет Wine (from sklearn.datasets import load\_wine) необходимо решить задачу классификации методом решающих деревьев.
10. Используя датасет Wine (from sklearn.datasets import load\_digits) необходимо решить задачу классификации методом случайных деревьев.

**Критерий оценки за аттестацию**

**30 баллов** выставляется, если обучающийся выполнил домашнюю индивидуальную работу в полном объеме;

**11-29 баллов** выставляется, если обучающийся выполнил 2/3 заданий в индивидуальной работе;

**1-10 баллов** выставляется, если обучающийся выполнил 1/3 заданий в домашней работе;

**0 баллов** выставляется, если обучающийся не выполнил индивидуальную работу.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»

**Комплект тестовых заданий по дисциплине «Машинное обучение»**

1. Распознавание рукописных символов алфавита для решения задачи мультиномиальной классификации относится к типу:

- a) обучение с учителем
- b) обучение без учителя
- c) обучение с подкреплением

2. Метод разведочного анализа, позволяющий организовать информацию в содержательные подгруппы не имея никаких предварительных сведений о принадлежности группе относится к следующему типу МО:

- a) обучение с учителем
- b) обучение без учителя
- c) обучение с подкреплением

3. В ходе взаимодействия со средой агент на основе разведочного подхода путем проб и ошибок или совещательного планирования производит вычленения серии действий, которые максимизируют это вознаграждение. Такая задача относится к следующему типу МО:

- a) обучение с учителем
- b) обучение без учителя
- c) обучение с подкреплением

4. Единичная ступенчатая функция, ступенчатая функция Хевисайда, функция единичного скачка имеет вид:

- a)  $\begin{cases} \alpha(\exp z - 1), & \text{если } z \leq 0 \\ z, & \text{если } > 0 \end{cases}$
- b)  $\begin{cases} \alpha z, & \text{если } z < 0 \\ z, & \text{если } \geq 0 \end{cases}$
- c)  $\begin{cases} -1, & \text{если } z \leq 0 \\ 1, & \text{если } > 0 \end{cases}$
- d)  $\begin{cases} -1, & \text{если } z < 0 \\ 1, & \text{если } \geq 0 \end{cases}$

5. ADALINE (ADaptive Llinear NEuron, адаптивный линейный нейрон) представлен ниже. Для правильной работы алгоритма необходимо использовать одно из следующих выражений.

```
def fit(self, X, y):
    """ Fit training data.
    Parameters
    X : {array-like}, shape = [n_samples, n_features]
        Training vectors, where n_samples is the number of samples and
        n_features is the number of features.
    y : array-like, shape = [n_samples]
        Target values.
    Returns
    self : object
    """
    rgen = np.random.RandomState(self.random_state)
    self.w_ = rgen.normal(loc=0.0, scale=0.01, size=1 + X.shape[1])
```

```

self.cost_ = []
for i in range(self.n_iter):
    net_input = self.net_input(X)
    output = self.activation(net_input)
    errors = (y - output)
    self.w_[1:] += self.eta * _____
    self.w_[0] += self.eta * errors.sum()
    cost = (errors**2).sum() / 2.0
    self.cost_.append(cost)
return self

```

- a) X.T
- b) X.dot(errors)
- c) X.T.dot(errors)

6) Частная производная функции стоимости относительно j-го веса

$$\frac{\partial}{\partial w_j} \frac{1}{2} \sum_i (y^{(i)} - \phi(z^{(i)}))^2 =$$

- a)  $\sum_i (y^{(i)} - \phi(z^{(i)}))^2 x_j^{(i)}$
- b)  $-\sum_i (y^{(i)} - \phi(z^{(i)})) x_j^{(i)}$
- c)  $-\sum_i (y^{(i)} - \phi(z^{(i)}))^2 x_j^{(i)}$

7) Логистическая функция или сигмоида имеет вид:

- a)  $\phi(z) = \frac{1}{1+e^{-z^3}}$
- b)  $\phi(z) = \frac{1}{1+e^z}$
- c)  $\phi(z) = \frac{1}{1+e^{-z}}$

8) В регулированной функции стоимости логистической

$$J(w) = C \left[ \sum_{i=1}^n \left( -y^{(i)} \log(\phi(z^{(i)})) - (1 - y^{(i)}) \log(1 - \phi(z^{(i)})) \right) \right] + \frac{1}{2} \|w\|^2$$

регрессии параметр C является:

- a) регуляризация параметра
- b) инверсия параметра
- c) смещение

9) Для использования линейного метода SVM для решения задачи классификации необходимо выполнить следующий код:

- a) `from sklearn . linear_model import LogisticRegression`  
`lr = LogisticRegression(C=1000 . 0, random_state=0 )`  
`lr.fit(X_train_std, y_train )`
- b) `from sklearn . svm import SVC`  
`svm = SVC(kernel= ' linear' , C=1 . 0, random_state=0)`  
`svm.fit(X_train_std, y_train)`
- c) `from sklearn . svm import SVC`  
`svm = SVC(kernel= ' rbf ' , random_state=0 , gamma=0 . 10, C=10.0)`  
`svm.fit(X_train_std, y_train)`

10) Для использования линейного метода SVM для решения задачи регрессии необходимо выполнить следующий код:

- a) `from sklearn . linear_model import LogisticRegression`  
`lr = LogisticRegression(C=1000 . 0, random_state=0 )`  
`lr.fit(X_train_std, y_train )`
- b) `from sklearn . svm import SVC`  
`svm = SVC(kernel= ' linear' , C=1 . 0, random_state=0)`  
`svm.fit(X_train_std, y_train)`

- c) `from sklearn . svm import SVC`  
`svm = SVC(kernel= ' rbf ', random_state=0 , gamma=0 . 10, C=10.0)`  
`svm.fit(X_train_std, y_train)`

11) Для случая, представленного на рисунке

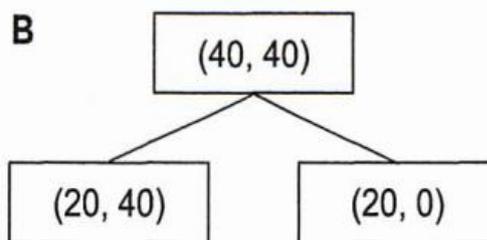


Рисунок. Разделение образцов

Рассчитайте меру неоднородности Джини и энтропийный критерий:

- a) 0,16 и 0,31  
 b) 0,25 и 0,16  
 c) 0,31 и 0,125

12) Расстояние Минковского для образцов с вещественными значениями определяется следующим образом.

- a)  $d(x^{(i)}, x^{(j)}) = |x_k^{(i)} - x_k^{(j)}|$   
 b)  $d(x^{(i)}, x^{(j)}) = \sqrt{\sum_k |x_k^{(i)} - x_k^{(j)}|^2}$   
 c)  $d(x^{(i)}, x^{(j)}) = \sqrt[p]{\sum_k |x_k^{(i)} - x_k^{(j)}|^p}$

13)

Что делает следующий код?

```
onehot_encoder = OneHotEncoder(sparse=False)
encoded_categorical_columns =
pd.DataFrame(onehot_encoder.fit_transform(df[categorical_columns]))
encoded_categorical_columns.head()
```

- a) В столбце номинального признака создавать новый фиктивный признак для каждого уникального значения  
 b) Преобразование порядковых признаков

14) Какую операцию описывает формула:

$$x^{(i)} = \frac{x^{(i)} - x_{min}}{x_{max} - x_{min}}$$

- a) Нормализации  
 b) Стандартизации

15) Метод ближайших соседей — простейший классификатор, основанный на оценивании сходства объектов. Классифицируемый объект относится к тому классу, которому принадлежат ближайшие к нему объекты обучающей выборки. Основные недостатки данного классификатора:

- a) Выбор числа соседей  
 b) Обработка больших выборок данных  
 c) Проблема выбора метрики  
 d) Необходимость исключения зашумленных объектов

16) Следующий код выведет

```
pca = PCA(n_components=None)
X_train_pca = pca.fit_transform(X_train_std)
print(pca.explained_variance_ratio_)
```

- a) Средние значения признаков  
 b) Дисперсии главных компонент  
 c) Доли объяснённых дисперсий

17) Анализ главных компонент

- a) метод линейного преобразования

- b) обучение без учителя
  - c) применяется для снижения размерности пространства признаков
  - d) применяется для выделения признаков
  - e) метод оптимизации разделимость классов
- 18) Выстройте правильную после реализации ядерного анализа главных компонент
- a) Использовать верхние собственные вектора центрированной матрицы ядра
  - b) Вычислить матрицу ядра
  - c) Центрировать матрицу ядра
- 19) Под k-блочной перекрестной проверкой подразумевается
- a) случайное разбиение на k блоков без возврата
  - b) случайное разбиение на k блоков с возвратом
  - c) k-1 блок используется для обучения
  - d) 1 блок используется для проверки
- 20) Матрица несоответствий имеет вид:

4	5
2	7

Истинно отрицательных предсказаний бинарного классификатора:

- a) 4
- b) 5
- c) 2
- d) 7

21) Отношение истинно положительных к сумме истинно положительных и истинно отрицательных исходов:

- a) Точность
- b) Полнота
- c) Комбинация точности и полноты, мера F1

22) Для сериализации и десериализации объектных структур (моделей МО) на языке python применяется модуль:

- a) os
- b) pickle
- c) re
- d) numpy

23) Что выполняет следующий фрагмент кода:

```
import sqlite3
import os
if os.path.exists('reviews.sqlite'):os.remove('reviews.sqlite')
conn = sqlite3.connect('reviews.sqlite')
c = conn.cursor()
c.execute('CREATE TABLE review_db (review TEXT, sentiment INTEGER, date TEXT)')
c.execute("SELECT * FROM review_db WHERE date BETWEEN '2017-01-01 10:10:10'
AND DATETIME('now')")
results = c.fetchall()
conn.close()
```

- a) Закрывает соединение с таблицей
- b) Создает соединение
- c) Создает новую таблицу
- d) Выводит все до определенной даты

24) Приведенная ниже функция

```
def classify(document):
    label = {0: 'negative', 1: 'positive'}
    X = vect.transform([document])
    y = clf.predict(X)[0]
    proba = np.max(clf.predict_proba(X))
```

return label[y], proba

- a) для текстового документа возвращает метку класса
- b) сохраняет в базе данных отзыв
- c) обновляет классификатор текстов

25) Гребневая модель, где добавлен

- a) L1-штраф
- b) L2-штраф
- c) L1 и L2 штрафы

26) Для решения задачи регрессии на основе случайного леса необходимо выполнить следующий фрагмент кода:

A)

```
from sklearn.tree import DecisionTreeRegressor
tree = DecisionTreeRegressor(max_depth=3)
tree.fit(X, y)
```

Б)

```
from sklearn.ensemble import RandomForestRegressor
forest = RandomForestRegressor(n_estimators=10000,
                              criterion='mse',
                              random_state=1777,
                              n_jobs=-1)
```

```
forest.fit(X_train, y_train)
```

27) Алгоритм k-means++ содержит следующие шаги:

- a) Инициализация пустого множество центроидов
- b) Поиск минимального квадратичного расстояния до любого из центроидов
- c) случайный выбор центроида
- d) вычисление внутречластерной связанности

28) Для оценки оптимального числа кластеров используются следующие методы:

- a) Метод локтя
- b) Силуэтные графики
- c) Метод оптимальной группы
- d) Метод KNN

29) Агломеративная иерархическая кластеризация представлена следующими методами:

- a) дивизионный
- b) одиночной связи
- c) полной связи

30) Следующий фрагмент кода

```
model = keras.models.Sequential()
model.add(
    keras.layers.Dense(
        units=50,
        input_dim=X_train_centered.shape[1],
        kernel_initializer='glorot_uniform',
        bias_initializer='zeros',
        activation='tanh'))
model.add(
    keras.layers.Dense(
        units=50,
        input_dim=50,
        kernel_initializer='glorot_uniform',
        bias_initializer='zeros',
        activation='tanh'))
model.add(
    keras.layers.Dense(
```

```
units=y_train_onehot.shape[1],
input_dim=50,
kernel_initializer='glorot_uniform',
bias_initializer='zeros',
activation='softmax'))
sgd_optimizer = keras.optimizers.SGD(
    lr=0.001, decay=1e-7, momentum=.9)
model.compile(optimizer=sgd_optimizer,
              loss='categorical_crossentropy')
history = model.fit(X_train_centered, y_train_onehot,
                  batch_size=64, epochs=50,
                  verbose=1,
                  validation_split=0.1)
```

- a) Создает модель с тремя полносвязными слоями
- b) Инициализирует новую модель
- c) Использует статистический градиентный спуск для оптимизации
- d) Предсказывает метки классов
- e) Производит обучение модели

### **Критерии оценки:**

91-100 баллов выставляется обучающемуся, если процент правильных ответов составляет 90-100.

76-90 баллов выставляется обучающемуся, если процент правильных ответов составляет 70-89.

61-75 баллов выставляется обучающемуся, если процент правильных ответов составляет 50-69.

0-60 баллов выставляется обучающемуся, если процент правильных ответов составляет менее 50.