Билет 1

(1) Теория:

- Машинное обучение: определение, ключевые примеры задач (регрессия, классификация), классификация задач (обучение с учителем, без учителя, обучение с подкреплением).

(2) Расчёт/Выкладка:

- Выведите метод наименьших квадратов (МНК) из принципа наибольшего правдоподобия, предполагая гауссово распределение ошибок.

(3) Практика программирования:

- Напишите на Python код, читающий входной файл, где каждая строка — JSON. Из каждого объекта извлеките несколько нужных полей (например, `user_id`, `item_id`) и сформируйте `pandas.DataFrame`.

Билет 2

(1) Теория:

- Матрица «объект–признак» и её роль в ML. Задачи регрессии и классификации: основные функции потерь (MSE, MAE, logloss и т. п.). Чем отличаются функции потерь от метрик?

(2) Расчёт/Выкладка:

- Выведите оценку параметра бернуллиевской случайной величины из метода максимального правдоподобия.

(3) Практика программирования:

- Прочитайте CSV-файл с выборкой, у которого один из признаков является категориальным (например, `color`). Выполните one-hot encoding этого признака (через `pandas.get_dummies` или `sklearn.preprocessing.OneHotEncoder`).

Билет 3

(1) Теория:

- Работа с категориальными признаками: упорядоченные и неупорядоченные признаки, методы кодирования (one-hot, счётчики, hash trick). В каких случаях предпочтительнее применять счётчики?

(2) Расчёт/Выкладка:

- Аналитическая формула для коэффициентов линейной регрессии (без регуляризации). Краткий вывод из условия минимизации $\|Xw - y\|^2$.

(3) Практика программирования:

- На Python/Numpy: сгенерируйте искусственную выборку (несколько десятков точек), обучите простую линейную регрессию методом МНК (по аналитической формуле или через `np.linalg.solve`), выведите найденные веса и визуализируйте результат.

Билет 4

(1) Теория:

- Линейная модель: L1 и L2 регуляризация, их свойства (разреженность вектора весов при L1, сглаживание при L2). Может ли чисто линейная модель уловить нелинейную зависимость?

(2) Расчёт/Выкладка:

- Аналитическое выражение для линейной регрессии с L2-регуляризацией (Ridge).

(3) Практика программирования:

- Реализуйте на Python функцию для вычисления ROC-кривой и ROC AUC по заданным истинным меткам y_{true} и предсказанным вероятностям $y_{\text{pred\_proba}}$. Сравните с результатом `sklearn.metrics.roc_curve` и `roc_auc_score`.

Билет 5

(1) Теория:

- Метрики задачи бинарной классификации: precision, recall, TPR, FPR, ROC, ROC AUC, F1-мера. Как выбирается порог классификатора?

(2) Расчёт/Выкладка:

- Запишите формулы для precision, recall, F1; опишите процедуру построения ROC-кривой и вычисления ROC AUC.

(3) Практика программирования:

- Напишите функцию, которая принимает список (или несколько) текстовых строк, выделяет из них уникальный словарь (перечень всех встречающихся слов), возвращает этот набор как «алфавит» для дальнейшей обработки (пример: классификация текстов).

Билет 6

(1) Теория:

- Дерево решений: процедура построения (жадный сплит), критерии качества сплита — энтропия, дисперсия, Gini impurity. Как выбирается значение в листе для регрессии?

(2) Расчёт/Выкладка:

- Что такое «жадность» при построении сплита? Запишите формулу Gini Impurity и кратко поясните её смысл.

(3) Практика программирования:

- С помощью scikit-learn обучите Decision Tree (например, на датасете Iris), выведите глубину дерева, важность признаков, визуализируйте дерево (`export_graphviz` или `plot_tree`) и покажите результат.

Билет 7

(1) Теория:

- Бэггинг и Random Forest: в чём идея бэггинга? Как формируется случайный лес? Как влияет выбор гиперпараметров (число деревьев, глубина, `max_features`) на bias–variance компромисс?

(2) Расчёт/Выкладка:

- Формула для энтропии $H(p) = -\sum_i p_i \log p_i$. При каких условиях энтропия минимальна и при каких максимальна (для k равновозможных исходов)?

(3) Практика программирования:

- Напишите функцию на Python, которая принимает список строк (каждая строка — JSON-объект) и возвращает `pandas.DataFrame` только с нужными полями, отбрасывая всё остальное.

Билет 8

(1) Теория:

- Градиентно бустированные деревья: в чём заключается идея «градиентного» шага? Как итеративно строятся деревья, приближая антиградиент функции потерь?

(2) Расчёт/Выкладка:

- Общий вид logloss:

$$\text{logloss} = -\frac{1}{N}\sum_{i=1}^{N}\Big[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\Big].$$

  Покажите, как при минимизации logloss можно вывести оценку параметра Бернулли (равную средней наблюдённой частоте успеха).

(3) Практика программирования:

- Обучите градиентный бустинг (например, `XGBClassifier` или `LightGBM`) на реальных данных. Подберите основные гиперпараметры (`learning_rate`, `n_estimators` и т. д.), оцените качество по ROC AUC.

Билет 9

(1) Теория:

- SVD (сингулярное разложение): основная идея, связь с факторизацией матриц, применение в рекомендательных системах, сжатии изображений и т. д.

(2) Расчёт/Выкладка:

- Запишите выражение для $\mathbf{A} = \mathbf{USV}^\top$. Поясните, что такое матрица ранга 1 и как она выражается в терминах столбцов $\mathbf{U}$, $\mathbf{V}$ и диагональных элементов $\mathbf{S}$.

(3) Практика программирования:

- Сгенерируйте случайную матрицу (например, `np.random.randn(10, 5)`), выполните SVD (`np.linalg.svd`), сохраните `U, S, Vt`. Проверьте, что `U @ np.diag(S) @ Vt` совпадает с исходной матрицей (с учётом вычислительной погрешности).

Билет 10

(1) Теория:

- Доверительные интервалы и проверка гипотез: что такое уровень доверия, мощность теста, нулевая гипотеза, статистическая значимость?

(2) Расчёт/Выкладка:

- Задача условной вероятности (про такси): в городе 15 % такси «Синие» и 85 % — «Зелёные». Свидетель ночью с вероятностью 80 % правильно определяет цвет. Свидетель сказал, что такси — «Синее». Какова вероятность, что такси действительно было «Синим»?

(3) Практика программирования:

- Сгенерируйте выборки из нормального распределения с разными средними и разными размерами. Постройте для каждой доверительный интервал для среднего (например, через `statsmodels` или `scipy.stats`), сравните результаты при разных размерах выборки.

Билет 11

(1) Теория:

- Эмбеддинги: что это такое, зачем нужны? Пример: word2vec — как обобщается идея представления слов в векторном виде?

(2) Расчёт/Выкладка:

- Задача условной вероятности (про вирус): XYZ-вирус встречается у 1 из 1000 человек. Тест даёт 5 % ложноположительных результатов. Тест показал, что человек заражён. Какова вероятность, что он действительно болен?

(3) Практика программирования:

- Реализуйте простую обучаемую embedding-матрицу (в Numpy или PyTorch/TensorFlow) для набора уникальных слов. Покажите на игрушечном примере, как это работает.

Билет 12

(1) Теория:

- Байесовский подход vs. классический (частотный) подход: оценка параметров Гаусса, Бернулли, различия в понимании доверительных интервалов. Формула Байеса.

(2) Расчёт/Выкладка:

- Выведите формулу Байеса из определения условной вероятности и объясните основные элементы: априор, правдоподобие, нормировочная константа.

(3) Практика программирования:

- Реализуйте (на numpy или scikit-learn) наивный Байесовский классификатор (BernoulliNB) для задачи бинарной классификации. Сравните с `sklearn.naive_bayes.BernoulliNB`.

Билет 13

(1) Теория:

- Наивный Байесовский классификатор и сопряжённые распределения для нормального и бернуллиевского случая. Зачем нужна сопряжённость в Байесовских моделях?

(2) Расчёт/Выкладка:

- Выведите производную (градиент) сигмоиды $\sigma(x) = \frac{1}{1+e^{-x}}$.

(3) Практика программирования:

- Реализуйте на numpy backpropagation для логистической регрессии: получите градиент по весам $\mathbf{w}$ с учётом функции потерь logloss.

Билет 14

(1) Теория:

- Обзор методов: KNN, наивный Байес, линейные модели, SVM, деревья решений, ансамбли (бэггинг, бустинг, стакинг). В чём связь и различия с нейронными сетями?

(2) Расчёт/Выкладка:

- Продифференцируйте $\log(\det(\mathbf{A}))$ по матрице $\mathbf{A}$.

(3) Практика программирования:

- Напишите код генерации искусственной 2D-выборки из двух классов. Обучите `sklearn.svm.SVC` и визуализируйте разделяющую границу вместе с точками исходной выборки.

Билет 15

(1) Теория:

- Нейронные сети: что такое backpropagation и матричное дифференцирование? Разные типы функций активации (sigmoid, tanh, ReLU и т. д.), их свойства (монотонность, насыщение, дифференцируемость).

(2) Расчёт/Выкладка:

- Найдите производную по аргументу от $\tanh(x)$. Чем $\tanh(x)$ отличается от $\sigma(x)$ с точки зрения диапазона значений?

(3) Практика программирования:

- Реализуйте однослойный перцептрон на numpy (прямой проход и обратный проход). Проверьте на небольшом синтетическом датасете, что ошибка на обучении уменьшается.

Билет 16

(1) Теория:

- RNN: устройство, какие матрицы и вектора в её составе, как происходит обновление состояния на каждом шаге, в чём идея обучения RNN через backpropagation through time.

(2) Расчёт/Выкладка:

- Производная функции потерь RNN по логитам на каждом шаге. Опишите основные шаги бэктрекинга (backprop through time): как распространяется градиент по временным шагам?

(3) Практика программирования:

- Реализуйте упрощённую RNN для предсказания следующего символа (буквенная генерация). Сгенерируйте небольшой обучающий набор, обучите и попробуйте сгенерировать текст.

Билет 17

(1) Теория:

- LSTM: чем отличается от классической RNN? Идея «долгой краткосрочной памяти» (гейты), преимущества по сравнению с обычной RNN. Короткое упоминание теоремы универсальной аппроксимации.

(2) Расчёт/Выкладка:

- Как реализуется Adagrad на numpy? Какую роль играет экспоненциальное сглаживание в оптимизаторах типа RMSProp/Adam? Приведите формулу экспоненциального среднего.

(3) Практика программирования:

- Реализуйте на Python генерацию текста RNN с параметром «температура». Объясните, как температура влияет на «креативность» генерируемого текста.
