1. Методами машинного обучения (не статистическими тестами) показать, что разбиение на трейн и тест репрезентативно.

Пусть у нас имеется процедура train_test_split(). Делаем сплит, отбираем данные из обоих кучек для hold-out валдиации, обучаем модель классификации, пусть будет лог.рег. на основе данных предсказывать флаги train и test. Смотрим на валидации метрику ROC-AUC, т.к. это мера “перемешанности”, т.е. на сколько хорошо по заданному числовому параметру, в нашем случае вероятность принадлежности классу, возвращаемая лог.регрессией, мы ранжируем данные. Повторяем процедуру n_раз, со случайным разбиением. Смотрим на распределение меры ROC-AUC, если оно в окрестности 0.5, то мы наше разбиение на трейн и тест репрезентативно. По ЦПТ, с увеличением количества n: дисперсия будет падать и меры ROC-AUC будут с меньшим размахом распределяться в окрестности 0.5.

2. Есть кластеризованный датасет на 4 кластера (1, 2, 3, 4). Бизнес аналитики посчитали, что самым прибыльным является кластер 2. Каждый клиент представлен в виде 10-тимертного вектора, где первые 6 значений транзакции, а оставшиеся: возраст, пол, социальный статус (женат (замужем)/неженат (не замужем)), количество детей. Нужно поставить задачу оптимизации для каждого клиента не из кластера 2 так, чтобы увидеть как должен начать вести себя клиент, чтобы перейти в кластер 2.

Для каждого кластера X решаем задачу классификации "кластер 2" vs " кластер X" при помощи логистической регрессии и получаем для каждого кластера вероятность, зависящую от b (найденные веса) и x (значения фичей). Для каждого клиента, в каждом кластере решаем задачу максимизации этой вероятность по x (фичам). Если мы предполагаем, что клиент не может менять свой возраст, пол, количество детей или социальный статус, то эти фичи надо исключить. Конкретное решение зависит от постановки исходной задачи. Строго говоря, и свой пол, клиент тоже может сменить, в текущий исторический момент :)

3. Что лучше 2 модели случайного леса по 500 деревьев или одна на 1000, при условии, что ВСЕ параметры кроме количества деревьев одинаковы?

Тривиальный случай. Если к параметрам модели также относится random_state, как это реализовано в sklearn, то две модели по 500 - это одна и та же модель => если для наших данных 500 деревьев не достаточно для хорошей классификации, то 1000 будет лучше работать.

Менее тривиальный случай, а если мы random_state не фиксируем, т.е. его значение = None.

Модель случайного леса не переобучается от роста числа деревьев. Увеличение числа деревьев приводит только к увеличению качества. После некоторого N это качество перестает расти и стабилизируется. Таким образом, можно представить себе набор данных, при котором 500 деревьев будет будет недостаточно для робастной работы модели. Если усреднять предсказания двух моделей по 500 деревьев, то это никак не повлияет на Bias, однако две модели по 500 деревьев при усреднении могут давать больший Variance, чем одна модель на 1000 деревьев.

Всё это становится ещё более критично, если был неправильно подобран параметр min_samples_leaf, т.к. при его слишком маленьком значении случайный лес может переобучаться => ошибка возрастает, если N не достаточно велико. Увеличение N не может компенсировать переобучение в результате неверно настроенного параметра min_samples_leaf.

Эксперименты на эту тему в файле RF_test.ipynb. В конце взято разбиение на две модели по одному дерево и одну модель на два дерева, но таковые данные. Очевидно, можно найти такие данные, где аналогичное разбиение будет проходить в окрестности 500.

4. В наличии датасет с данными по дефолту клиентов. Как, имея в инструментарии только алгоритм kmeans получить вероятность дефолта нового клиента.

Проводим кластеризацию k_means, предположим, что мы уже нашли оптимальное количество кластеров. Далее в каждом кластере считаем вероятность дефолта обычным частотным методом: т.е. число дефолтных точек / на число точек в кластере. Назовем это вероятностью дефолта кластера.

Далее можно предложить несколько вариантов:

1) Для новой точки, предсказываем кластер обычным, hard, образом, выбирая таковой по минимальной дистанции. Вероятность дефолта кластера и будет искомая вероятность.
2) Можно модифицировать задачу и перейти к soft clustering, используя, к примеру функцию softmax. Тогда вероятность дефолта будет рассчитываться, как условная вероятность, т.е. вероятность принадлежности новой точки к кластеру * вероятность дефолта кластера

3) Можно развить вариант (2) и брать средневзвешенное значение условных вероятностей, рассчитанных для каждого кластера. Вес рассчитывать исходя из меры расстояния от точки до кластера.

5. Есть выборка клиентов с заявкой на кредитный продукт. Датасет состоит из персональных данных: возраст, пол и т.д. Необходимо предсказывать доход клиента, который представляет собой непрерывные данные, но сделать это нужно используя только модель классификации.

Можно факторизовать доход. Разделив его на интервалы. Самый просто вариант: “низкий”, “высокий” - тогда это будет бинарная классификация. Можно делить на меньшие интервалы (размер которых будет зависеть от объема данных) и тогда будет решать задача мульти-классовой классификации.