1. В алгоритм k-fold на что влияет параметр k? В каких случаях стоит выбирать его большим, в каких маленьким?

k влияет на объем выборки для валидации и на кол-во различных выборок.
Чем больше k тем менее предвзятую оценку алгоритма мы будем получать.
В теории лучшее значение для k это кол-во всех обучаемых объектов (Leave-One-Out Cross-Validation), что будет давать несмещенную оценку, но это будет требовать больших вычислений и займёт много времени.
Если мы хотим проверить небольшое кол-во гипотез (или параметров для одного и того же алгоритма), то k стоит выбирать большим.
Если же гипотез или параметров много, то стоит взять небольшое k, чтобы в принципе была возможность их проверить.

2. Почему в практических задачах делят выборку на три части: тренировочную, валидационную и тестовую? Почему не хватает первых двух?

Делят потому что существует возможность переобучиться на валидационных данных. А если использовать еще и независимую тестовую выборку, то тогда можно показать реальный, более объективный результат.

3. Рассмотрим альтернативу алгоритму k-fold — алгоритм «leave-all-out». В этом алгоритме мы разбиваем всеми возможными способами выборку на две части — на одной части тренируем алгоритм, а на другой (отложенной) измеряем качество. Затем все полученные измерения усредняем и используем в качестве оценки L_D. В чём преимущества и недостатки данного подхода по сравнению с k-fold?

Преимуществом данного алгоритма является то, что при таком варианте мы будем обучаться и валидироваться на всевозможных разбиениях обучаемой выборки и таким образом она будет как бы перемешана. Т.е. в k-fold алгоритме мы последовательно разбиваем выборку на куски, а тут объекты выборки для обучения и для валидации могут попасться из любого места выборки.

Недостаток такого подхода в том, что мы не можем регулировать кол-во итераций для валидации (в этом случае этот будет константа зависящая от набора выборки). И при больших объемах выборки не будет возможности регулировать время работы алгоритма и оно всегда будет большим.

И я не вижу особо смысла обучаться на маленьком размере выборки и валидироваться на большом (т.е. в leave-all-out алгоритме будет итерация, где обучаемся на одном элементе и валидируемся на всём остальном).