AB-Rent is a multinational car rental company. They have a fleet of 500K vehicles, ranging from sub-economy cars to 12 foot trucks. Annual sales are about 6B$ per year with a profit of 100M$. AB-Rent follows an extensive maintenance schedule to avoid the rental of defective cars. There are check-ups (pre-determined by the car manufacturer) scheduled by mileage and age of the car. After every rental, mechanics visually inspect and drive the cars to check for defects. A defective car, if it were rented, would generate both direct costs (replacement, towing, refunds) and indirect costs (reputation, customer churn). Maintenance costs total about 5% of annual sales (~300M$).

To decrease maintenance and defect costs, AB-Rent decided to launch a Big Data project. The goal is to predict more accurately when a car needs a check-up and repair.

1. **What do we want to predict and how do we measure the quality of the prediction?**
   1. How will a better prediction improve the bottomline?  
      *If I can predict, with high accuracy if a car fails next week, I can send it to a check-up only if it needs repair/maintenance. It will decrease the maintenance costs and decrease the number of cars which breakdown while rented.*
   2. You want to have decision support, a fully automated system, or to know just the factors which are important? How will the agent use the system? Why are these important questions to ask?  
      *The output of the system, the inner architecture and the techniques to be used, the evaluation metrics will all depend on this decision. A fully automated system can be completely black box: data in, decision out. There is no HCI, no dashboard, only a quantitative prediction which can be then handled some other automatic modules.   
      A decision support system may need to provide explanation in the form of a report about* why *the car should be repaired. The box should be highlighted if we need to understand the factors that cause defects. The final output is likely a human-interpretable dashboard with information that the agents can combine with their expert knowledge.  
      The data scientist will have to know from the beginning what the final product and the usage is, since it will generate early design, architectural, and organizational choices.*
   3. What should be the quantitative prediction?  
      *The problem can be cast in at least four different types of prediction problems, leading to quite different solutions.* 
      1. *Predicting the* mean lifetime *(the car will need repair in x±y days/weeks/months) is a* regression *problem. This is a relatively weak information since it makes it hard to input it into risk computations over various decision scenarios.*
      2. *Predicting the probability of failure in the next x days/weeks/months is a binary classification problem (will it fail or not next week?). If the time interval is adequate, this information can be chained into an automatic module and can also serve for decision support. It can also be used for limited risk assessment.*
      3. *Predicting an ordering of the cars to be repaired (which car will fail first) is a* ranking *problem. It is an adequate information if, for example, the maintenance throughput has an upper limit (say, one car per day). It can also serve as input for a limited dashboard for decision support. It’s hard to input it into risk assessment computations.*
      4. *Predict the full posterior distribution function p(t): for every time t, integral\_0^t p(t) dt is the probability that the car fails before t. This is the richest type of predictor: all the three predictions above can be derived from it. Unfortunately, this is the hardest to obtain. It also “spreads” capacity equally in time which may lead to misuse resources (computational, data scientists): we probably don’t care if the prediction p(t) is bad for t > year.*
   4. How do we measure success?  
      *Standard score metrics come with each of these predictions. Whether they are adequate depends on how the KPI is related on the predictor.*
      1. *The classical score (loss) of regression predictors is the* squared error (SE)*: the squared difference between the real lifetime and the predicted lifetime. It gives the same penalty to a lifetime of 1 year mispredicted by a week and a lifetime of 1 day mispredicted by a week. Alternatively, we can measure the* relative (percentage) error*. It’s not immediately clear how an improvement of this measure will lead to a better KPI.*
      2. *Binary classification has a wide variety of quality scores. The simplest one is the zero-one error: the prediction is “will fail”, in reality it didn’t fail, or vice versa. Given the estimated probability of the two types of errors, one can derive the exact loss of different decision scenarios, connecting the quantified score to the KPI.*
      3. *The classical ranking loss is called AUC (area under the ROC curve) which represents the probability that two cars are mis-ordered by the prediction. The problem with this measure is that it gives the same weight to misordering two very reliable cars and two cars that are just on the verge of failing. From the point of view of risk (and KPI), this latter should have more weight in the score.*
      4. *Density estimation has its classical score called the likelihood (or log likelihood). The problem is as with ranking and regression: it is not calibrated to the risk.*
2. **What data do we need to develop a predictor?**
   1. The input or the predictor. *Once the quality metric (score) is set, the design should concentrate on finding data sources which are available at decision time, and which are likely to improve the prediction quality. Domain expertise is very useful at this stage. Data collection, cleaning, formatting, alignment, augmentation are important steps and doing them well may accelerate a lot the development of the data science workflow.  
      Modern data science algorithms can be fed with a lot of different kinds data, although the development time, difficulty, and the required data size can vary orders of magnitude by the type of input data.* 
      1. *Classical tabular (numerical or categorical, database) data is easy. It can take some time for IT to gather all information from different databases, but other than that costs are low. Examples: brand, type of car, engine, color, age, mileage, geographic location.*
      2. *Historical, time series, functional sensor data is slightly more complex, it can be bigger, and decision on aggregation should be carefully made. Examples: meteorological history at location, driving history, GPS history, maintenance events.*
      3. *Text and images, depending on the complexity, may require orders of magnitude more data, more time to tune, more expensive experts, and bigger computers. Examples: maintenance logs, customer reviews.*
      4. *Video is in R&D phase now, but we’ll get there.*
   2. The label/ground truth/tag.  
      *Any predictor will need* training data *to be built and evaluated. This means that we will need a sample where the* label *of whatever we want to predict is available. In our case this would be the real lifetime, maintenance records, and failure history of the cars. An important concept in this particular case is sampling bias. It is important to have a sample that reflects the future, otherwise both model building (training) and its evaluation will be biased. In predictive maintenance, this problem arises when the available length history is comparable to the lifetime of the cars (so most of the cars have not yet failed) or worse, if only cars that survived till the time records started to be taken are in the database (so it seems that older cars are more reliable).*Possible sources of labels:
      1. historical records
      2. human-annotated records
      3. simulated cases

http://www.automotive-fleet.com/article/story/1968/08/hertz-maintenance-is-a-two-word-operation-checking-and-uniformity.aspx