This project proposal is based on: https://www.kaggle.com/c/new-york-city-taxi-fare-prediction

Domain background.

In cities taxi tips is very popular. Help us to predict a price for our customers, because They want to know possible price before ride. Taxi company also could save money and protect the environment because they know where cabs are needed the most. Machine learning is good for this because we have a million of an real taxi tips so human cant calculate this efficiently. There are many attempts to solve this problems. For example http://www.vivekchoksi.com/papers/taxi_pickups.pdf. In this scientific paper an author use machine learning for predicting pickups so they will be able to optimize where taxis should be located. Another scientific paper about our domain is http://cs229.stanford.edu/proj2016/report/AntoniadesFadaviFobaAmonJuniorNewYorkCityCabPricing-report.pdf. The Author try to predict fare and duration of taxi trip.

Problem statement.

In other words we are trying to predicting a fare amount for a taxi. We have a data contains many real life taxi trips in New York. This is our inputs: pickup date, pickup location, dropout date, dropout location, number of passengers, sample fare amount. Our desired output is total fare amount for new data (not known before). This is regression supervised problem.

Datasets and inputs.

We use dataset provided by google.

- train.csv - Input features and target fare_amount values for the training set (about 55M rows).
- test.csv - Input features for the test set (about 10K rows). Your goal is to predict fare_amount for each row.
- sample_submission.csv - a sample submission file in the correct format (columns key and fare_amount). This file 'predicts' fare_amount to be $11.35 for all rows, which is the mean fare_amount from the training set.

Descripion of datafields:

ID:

- key - Unique *string* identifying each row in both the training and test sets

Features

- pickup_datetime- *timestamp* value indicating when the taxi ride started.
- pickup_longitude - *float* for longitude coordinate of where the taxi ride started.
- pickup_latitude - *float* for latitude coordinate of where the taxi ride started.
- dropoff_longitude - *float* for longitude coordinate of where the taxi ride ended.
- dropoff_latitude - *float* for latitude coordinate of where the taxi ride ended.
- passenger_count - *integer* indicating the number of passengers in the taxi ride.

Target

- fare_amount - `float` dollar amount of the cost of the taxi ride. This value is only in the training set; this is what you are predicting in the test set and it is required in your submission CSV.

Solution statement.

As you can read above, this is regression type problem. There are many algorithms for this kind of problem. For example linear regression, support vector regression and also deep neural network (DNN). More about this algorithms:
https://www.analyticsindiamag.com/top-6-regression-algorithms-used-data-mining-applications-industry/
https://towardsdatascience.com/selecting-the-best-machine-learning-algorithm-for-your-regression-problem-20c330bad4ef

I think I will use support vector regression or neural network because tensorflow calculate really fast on a gpu. For example a similar solution solved using DNN:
https://www.researchgate.net/publication/324706525_Taxi_Fare_Rate_Classification_Using_Deep_Networks

Benchmark model.

We can train selected model by using the train data set and test with the test data set. After training we measure difference between the predictions and the truth. Finally we could compre our model with another solution, for example our SVM to DNN . Example solutions: https://www.kaggle.com/c/new-york-city-taxi-fare-prediction/kernels

Evaluation metrics.

Evaluation metric for this project is the root mean-square error (RMSE). RMSE measures difference between the predictions of a model and the ground truth.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y)^2}$$

where yi is the ith observation and $\hat{y}_i$ is the prediction for that observation.

Project design.

1. Deep analyze the problem.
   - what kind of problem it is.
   - which data we have. Look at the data inputs and outputs section.
2. Data exploration.
   - total amount of our data

- do we need full data set or can we use data range.

3. Feature selection.
   - which feature is relevant
   - maybe we should group features or use specific data ranges.
4. Select model.
   - SVM or DNN (maybe if I have enough time I test two models)
5. Train model, make predictions.
6. Do some benchmarks. Look at section evaluation metrics
7. Make decision if model is good enough. Compare evaluation metrics with another competitors model: https://www.kaggle.com/c/new-york-city-taxi-fare-prediction/kernels
8. Tune model parameters if needed and go to step 5.