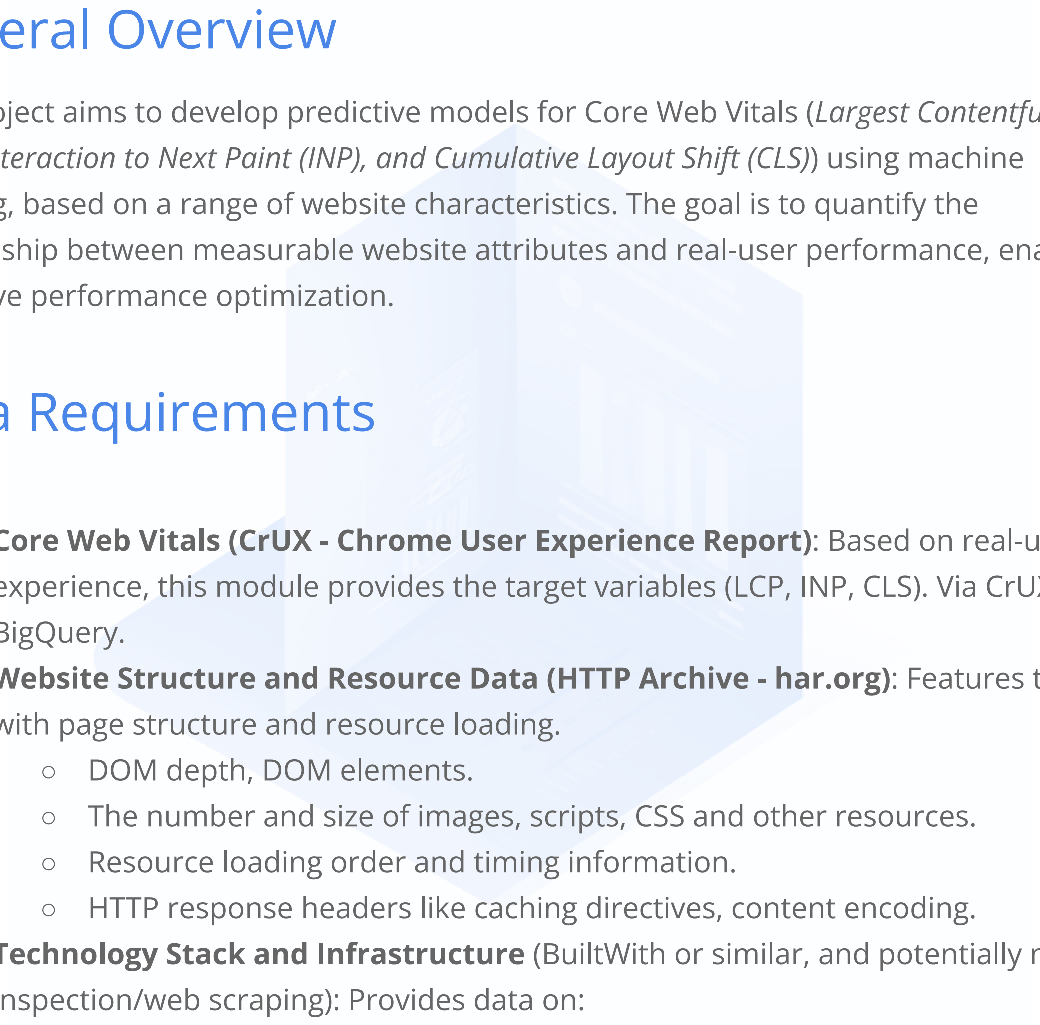
Problem Statement

A prediction model for Core Web Vitals Performance based on Website Characteristics.



General Overview

This project aims to develop predictive models for Core Web Vitals (*Largest Contentful Paint (LCP), Interaction to Next Paint (INP), and Cumulative Layout Shift (CLS)*) using machine learning, based on a range of website characteristics. The goal is to quantify the relationship between measurable website attributes and real-user performance, enabling proactive performance optimization.

Data Requirements

1. **Core Web Vitals (CrUX - Chrome User Experience Report)**: Based on real-user experience, this module provides the target variables (LCP, INP, CLS). Via CrUX API or BigQuery.
2. **Website Structure and Resource Data (HTTP Archive - har.org)**: Features to do with page structure and resource loading.
   - DOM depth, DOM elements.
   - The number and size of images, scripts, CSS and other resources.
   - Resource loading order and timing information.
   - HTTP response headers like caching directives, content encoding.
3. **Technology Stack and Infrastructure** (BuiltWith or similar, and potentially manual inspection/web scraping): Provides data on:
   - Use of specific frameworks (e.g., *React*, *Angular*, *Vue js.*).
   - Content Management Systems (CMS) like Adobe Experience Manager, Wordpress.
   - Use of CDN.

Potential Techniques

1. **Regression Analysis** (using Linear Regression, Random Forests, or Support Vector Regression):
    - ***Linear Regression***: A baseline model for predicting the continuous Core Web Vitals.
    - ***Random Forests***: An ensemble method which is suitable to control the non linear relations between Core Web Vitals and website characteristics. It performs feature importance analysis.
    - ***Support Vector Regression (SVR)***: Effective in high-dimensional spaces and it supports complex relationships.
2. **Feature Engineering and Dimensionality Reduction** (using Principal Component Analysis):
    - Feature engineering will create new features out of raw data (e.g.,total image size, ratio of text to images, number of blocking scripts, JavaScript execution time).
    - ***Principal Component Analysis (PCA)***: If the number of features is high, we will reduce dimensionality via PCA, while preserving important information and at the same time maximizing performance or minimizing complexity.
3. **Clustering**:
    - ***K-Means Clustering****:* It can classify group websites with approximately the same performance metric Core Web Vitals or site features. This could allow them to detect patterns and even to make mission specific regression models for different website clusters.
    - ***DBSCAN (Density-Based Spatial Clustering of Applications with Noise)***: It is able to identify clusters of differing sizes and shapes and is particularly useful to identify outliers or noise in the data (for example, websites with extremely strange performance characteristics). It gives you a way to identify websites that are clear outliers in terms of their website performance.