Disclaimer and Scope

This subsection contains a curated set of components that are explicitly cleared for public sharing. It includes only work authored by me, with emphasis on feature–engineering methods and lightweight machine–learning implementations that produced viable predictive models in a constrained, production–adjacent setting. No proprietary datasets, confidential parameters, or operational playbooks are disclosed; sensitive material in the full report has been removed or anonymised.

Purpose To document technical contributions that enabled operational decision support in public transport, specifically: (i) optimisation of driver routes; (ii) bus–dispatch scheduling tuned to headway deviation; and (iii) mitigation of bus bunching via predictive classification and control heuristics.

Provenance This excerpt is drawn from a 170-page technical report and accompanying software delivered by a five-person team to RapidKL on 21 April 2025.

Authorship and IP Unless stated otherwise, the design, implementation, and evaluation of the feature–engineering pipeline and baseline models were led by the author. All organisational names and trademarks remain the property of their respective owners. Sharing this excerpt does not imply endorsement and does not transfer intellectual-property rights.

Data Protection and Ethics Any person-level data were aggregated and/or anonymised; no personal data are included in this excerpt. External identifiers have been redacted where necessary.

6.2.3 Mathematical Definitions

Definitions

s Stop index.

A_f, A_b Actual times (let A_f denote the time for the front bus and A_b for the back bus).

S_f, S_b Scheduled times (let S_f denote the scheduled time for the front bus and S_b for the back bus).

L Lateness, defined by $L_i(s) = A_i(s) - S_i(s)$ for $i \in \{f, b\}$.

T In-traffic travel time between stops.

D Dwell time at a bus stop (the time taken for a bus to stop, open doors, board/alight passengers, and depart).

Δ Represents the change (difference) between the front and back buses for a given quantity.

$H^{\text{act}}(s)$ Actual headway, defined as $A_f(s) - A_b(s)$.

$H^{\text{sch}}(s)$ Scheduled headway, defined as $S_f(s) - S_b(s)$.

$\mathcal{S}$ Set of bus stops along a route.

$\#\mathcal{S}$ Total number of bus stops along a route.

P Set of bus pairs along a route.

Let the average headway deviation be defined as $\overline{H}$. Then, we have:

$$\overline{H} = \frac{1}{\#\mathcal{S}} \sum_{s \in \mathcal{S}} \Big| \big(H^{\text{act}}(s)\big) - \big(H^{\text{sch}}(s)\big) \Big| \tag{6.2.3.1}$$

Let the average headway deviation in terms of lateness be defined as $\overline{H}_L$. Then, using $L_i(s) = A_i(s) - S_i(s)$, we have:

$$\overline{H}_L = \frac{1}{\#\mathcal{S}} \sum_{s \in \mathcal{S}} \Big| L_f(s) - L_b(s) \Big| \tag{6.2.3.2}$$

Let the headway deviation in the degenerate case be defined as H. If there are fewer than two buses (i.e. $\#\text{bus} < 2$), then:

$$H = 0 \quad \text{if } \#\text{bus} < 2 \tag{6.2.3.3}$$

Let the headway deviation at a specific stop m be defined as H_m. Then, we have:

$$H_m = \Big|\big(A_f(m) - A_b(m)\big) - \big(S_f(m) - S_b(m)\big)\Big| \tag{6.2.3.4}$$

Let the daily route headway deviation be defined as H_{day}. For a given route and and a given day,

$$H_{\text{day}} = \frac{1}{\#P}\sum_{p\in P} \overline{H}^{(p)} \tag{6.2.3.5}$$

During a trip the a front and back bus can arrive at a bus stop m at a different time. Let the average bus arrival deviation at stop m be defined as H_m^{arr}. Then:

$$H_m^{\text{arr}} = \Big|\big(A_f^{\text{arr}}(m) - A_b^{\text{arr}}(m)\big) - \big(S_f^{\text{arr}}(m) - S_b^{\text{arr}}(m)\big)\Big| \tag{6.2.3.6}$$

During a trip the a front and back bus can leave a bus stop m at a different time. Let the average bus departure deviation at stop m be defined as H_m^{dep}. Then:

$$H_m^{\text{dep}} = \Big|\big(A_f^{\text{dep}}(m) - A_b^{\text{dep}}(m)\big) - \big(S_f^{\text{dep}}(m) - S_b^{\text{dep}}(m)\big)\Big| \tag{6.2.3.7}$$

We can rephrase the arrival component of headway deviation at stop m. Define

$$\Delta L_{m-1} = L_f^{\text{dep}}(m-1) - L_b^{\text{dep}}(m-1) \quad \text{and} \quad \Delta T_{m-1} = T_f(m-1,m) - T_b(m-1,m).$$

Then:

$$H_m^{\text{arr}} = \Big|\Delta L_{m-1} + \Delta T_{m-1}\Big| \tag{6.2.3.8}$$

Let the departure component of headway deviation at stop m be defined as H_m^{dep}. With the additional dwell-time difference

$$\Delta D_m = D_f(m) - D_b(m),$$

we have:

$$H_m^{\text{dep}} = \Big|\Delta L_{m-1} + \Delta T_{m-1} + \Delta D_m\Big| \tag{6.2.3.9}$$

For each k from 1 to $m-1$, define

$$\Delta T_k = T_f(k,k+1) - T_b(k,k+1) \quad \text{and} \quad \Delta D_k = D_f(k) - D_b(k) \tag{6.2.3.10}$$

Let the cumulative headway deviation at stop m be defined as H_m. Then,

$$H_m = \left|\Delta L_1 + \sum_{k=1}^{m-1} \Delta T_k + \sum_{k=1}^{m} \Delta D_k\right| \tag{6.2.3.11}$$

Let the headway deviation at an intermediate point x (where $0 \leq x \leq 1$) between bus stop m and $m+1$ be defined as $H_m(x)$. Then, we approximate it by

$$H_m(x) = H_m + \sum_{k=1}^{K(x)} \Delta T_k, \tag{6.2.3.12}$$

where $K(x)$ denotes the (discrete) number of sub-segments between stop m and the point corresponding to x, capturing the cumulative change in travel time differences over that interval.

Let us now refine the representation of the travel time differences by employing k-means clustering on the bearing data to reduce the segment lengths between bus stops. Suppose this clustering partitions the route into a total of R route segments. Then, replacing the summation over k with a summation over route segments, we define:

$$\sum_{r=1}^{R} \Delta T_r = \sum_{k=1}^{m-1} \Delta T_k, \tag{6.2.3.13}$$

where ΔT_r denotes the travel time difference for route segment r (with $r = 1, \ldots, R$). This approach clusters together parts of the route with similar driving characteristics (e.g., intersections and traffic patterns).

Assume that the distance over each route segment can be modeled by a continuous parameter x with $0 \leq x \leq 1$. Then, the cumulative travel time difference over the entire route can be approximated by the difference between the cumulative travel times of the front and back buses over all route segments. That is,

$$\sum_{r=1}^{R} \Delta T_r = \sum_{r=1}^{R} R_f(r,u) - \sum_{r=1}^{R} R_b(r,v) \tag{6.2.3.14}$$

where $R_f(r,u)$ denotes the travel time for the front bus in route segment r (as a function of time u), and $R_b(r,v)$ denotes the travel time for the back bus in the same segment (as a function of time v). This formulation provides a continuous approximation of the travel time differences along the route, enabled by the refined segmentation.

Rephrasing (6.2.3.11),

$$H_m = \left|\Delta L_1 + \left(\sum_{r=1}^{R} R_f(r,u) - \sum_{r=1}^{R} R_b(r,v)\right) + \sum_{k=1}^{m} \Delta D_k\right|$$

Since each of the values are negative and independent of each other,

$$H_m = \Big|\Delta L_1\Big| + \Big|(\sum_{r=1}^{R} R_f(r,u) - \sum_{r=1}^{R} R_b(r,v))\Big| + \Big|\sum_{k=1}^{m} \Delta D_k\Big|$$

Let the cumulative headway deviation at stop m be rephrased based on the refined segmentation. Assume that the dwell-time differences now depend on an additional variable (e.g., time) so that we write them as $D_f(k,u)$ and $D_b(k,v)$, and similarly, the initial lateness difference is expressed as

$$\Delta L_1 = L(u) - L(v).$$

Also, let $R_f(r,u)$ and $R_b(r,v)$ denote the travel time differences for the front and back buses in route segment r as functions of exogenous data u and v (which may include time, seasonality, etc.). Then, we have:

$$H_m = \Big|L(u) - L(v)\Big| + \Big|\sum_{r=1}^{R} R_f(r,u) - \sum_{r=1}^{R} R_b(r,v)\Big| + \Big|\sum_{k=1}^{m} D_f(k,u) - \sum_{k=1}^{m} D_b(k,v)\Big| \tag{6.2.3.15}$$

This suggests that headway deviation can be modeled as the sum of three components:

(1) The difference in initial lateness

(2) The cumulative travel time difference across route segments

(3) The cumulative dwell-time difference at stops

With sufficient exogenous data to model each of the components, we can better predict headway deviation through a linear model, a time series model as well as a model that takes into account of sequential data.

6.3.3 Mathematical Definitions

Let the travel time for the front bus on route segment r, denoted by $R_f(r,u)$, be further decomposed into the constant, Euclidean distance traveled $D_f(r)$ in that segment divided by the average velocity $\bar{v}_f(r)$ of the front bus over that segment. Similarly, let the travel time for the back bus on route segment r, denoted by $R_b(r,v)$, be decomposed into the constant, Euclidean distance traveled $D_b(r)$ divided by the average velocity $\bar{v}_b(r)$ of the back bus. Hence, we have:

$$R_f(r,u) = \frac{D_f(r)}{\bar{v}_f(r)} \quad \text{and} \quad R_b(r,v) = \frac{D_b(r)}{\bar{v}_b(r)}$$

Equivalently, the difference in cumulative travel time differences across all R route segments can be written as:

$$\sum_{r=1}^{R} R_f(r,u) - \sum_{r=1}^{R} R_b(r,v) = \sum_{r=1}^{R} \left(\frac{D_f(r)}{\bar{v}_f(r)} \right) - \sum_{r=1}^{R} \left(\frac{D_b(r)}{\bar{v}_b(r)} \right) \tag{6.3.3.1}$$

It is important to note that the distance used in these calculations is a constant, defined as the Euclidean distance between successive route segments. This choice provides a consistent and simplified representation of the spatial separation between segments. We did consider the Haversine Distance (the great-arc distance between two points on the Earth) instead of the Euclidean Distance however we deemed that the difference is negligible for when segments are less than 100 metres and it is reasonable to approxible point distances as a straight line and uncurved plane. In cases where the path is curved, our method assumes that as the number of segments increases, the effective straight-line distance decreases, thereby supporting a robust postulate that the effect of curvature on overall distance measurement is marginal.

From a mathematical standpoint, if the distance between two points is taken as d, the largest semi-circular arc that can connect these points is $\frac{1}{2}\pi d$. This implies that the maximum curved distance is $\frac{\pi}{2}$ times the straight-line distance. Given that travel time is ultimately computed by dividing distance by velocity, the amplified distance due to curvature is effectively moderated. Additionally, our experimental findings confirm that d is never more than 100 metres, further validating our approach.

Revising Equation (6.2.3.15) by substituting the travel time difference component defined in (6.3.3.1), the headway deviation at bus stop m is now given by:

$$H_m = \left| L(u) - L(v) \right| + \left| \sum_{r=1}^{R} \left(\frac{D_f(r)}{\bar{v}_f(r)} \right) - \sum_{r=1}^{R} \left(\frac{D_b(r)}{\bar{v}_b(r)} \right) \right| + \left| \sum_{k=1}^{m} D_f(k,u) - \sum_{k=1}^{m} D_b(k,v) \right| \tag{6.3.3.2}$$

We can thus define the headway deviation between consecutive bus stops m and $m+1$ as an extension of Equation (6.3.3.2) and Equation (6.2.3.12). In this case, the travel time difference component is computed over a number of trips – including any partial route – with the summation now taken over T_u trips. (note u is used to not confuse between f and b, so instead of front and back we use up and down.) Thus, the inter-stop headway deviation becomes:

$$H_m(x) = H_m^{dep} + \left| \sum_{t=1}^{T_u} \left(\frac{D_f(t)}{\bar{v}_f(t)} \right) - \sum_{t=1}^{T_u} \left(\frac{D_b(t)}{\bar{v}_b(t)} \right) \right| \tag{6.3.3.3}$$

Let $T_u + T_d$ be all the trips between bus stop m and $m+1$. Thus, we can further get the inter-stop headway devation with respect to $H_m + 1$.

$$H_m(x) = H_{m+1}^{arr} - \left| \sum_{t=1}^{T_d} \left(\frac{D_f(t)}{\bar{v}_f(t)} \right) - \sum_{t=1}^{T_d} \left(\frac{D_b(t)}{\bar{v}_b(t)} \right) \right| \tag{6.3.3.4}$$

With this being said, each real time point does not need to coincide with a bus stop. Real time data can be recorded between two bus stops and we can further translate this point to get the departure time of the bus with relation to the previous bus stop. We can also translate this point to get the arrival time of the bus with relation to the next bus stop.

As we can see, the real-time data need not coincide with a bus stop. Instead, by removing the travel time component due to the distance-velocity ratio, we can determine effective headways that allow for the estimation of both arrival and departure timings at a given stop.

Accordingly, we define the effective departure headway at stop m as

$$H_m^{dep} = H_m(x) - \left| \sum_{t=1}^{T_u} \left(\frac{D_f(t)}{\bar{v}_f(t)} \right) - \sum_{t=1}^{T_u} \left(\frac{D_b(t)}{\bar{v}_b(t)} \right) \right|, \tag{6.3.3.5}$$

Similarly, the effective arrival headway at stop m is given by

$$H_m^{arr} = H_{m-1}(x) - \left| \sum_{t=1}^{T_d} \left(\frac{D_f(t)}{\bar{v}_f(t)} \right) - \sum_{t=1}^{T_d} \left(\frac{D_b(t)}{\bar{v}_b(t)} \right) \right|, \tag{6.3.3.6}$$

Thus, the dwell time can be computed as the difference between the effective departure and arrival headways owing to $T_u + T_d$ being all the trips between bus stop m and $m+1$.

Imputation step: Note that in the extreme cases of lack of data, we can do a rough imputation of the the effective departure headway H_m^{dep} by expressing it in terms of the headway at the GPS data after the next stop:

$$\begin{aligned} H_m^{dep} &= H_m(x_1) - \left| \sum_{t=1}^{T_{u_1}} \left(\frac{D_f(t)}{\bar{v}_f(t)} \right) - \sum_{t=1}^{T_{u_1}} \left(\frac{D_b(t)}{\bar{v}_b(t)} \right) \right| \\ &\approx H_{m+1}(x_2) - \left| \sum_{t=1}^{T_{u_2}} \left(\frac{D_f(t)}{\bar{v}_f(t)} \right) - \sum_{t=1}^{T_{u_2}} \left(\frac{D_b(t)}{\bar{v}_b(t)} \right) \right| - \left[\overline{H}_{m+1} - \overline{H}_m \right]. \end{aligned} \tag{6.3.3.8}$$

Let x_1 be an example of a point in between m and $m+1$. Let x_2 be an example of a point in between $m+1$ and $m+2$. Note that since Headway deviation at bus stop $\overline{H}_{m+1}, \overline{H}_m$ is some prediction (which in this case, the mean was used in the imputation step) since the data is unavailable. This reformulation shows that departure timings at stop m can be triangulated with information arriving even after the $m+1^{th}$ stop. The same can be done for H_m^{arr} using GPS data before stop $m-1$. Within a route, this should be done only when there is insufficient data present since there is no exisiting information of the current bus stop.

6.3.4 Mathematical Analysis: K-means and application on road bends

Lemma 6.3.4.1

Proof: Let a continuous route be sampled as a sequence of points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ in a flat plane, i.e. $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subset \mathbb{R}^2$. Let the road comprise of S predominantly straight segments separated by $B = S - 1$ significant bends with curvatures. When K-means clustering is applied with $k \geq S$, the optimal partitioning intentionally asigns each straight segment to a distinct cluster, with cluster boundaries aligning at or near curved regions.

Importance: At $k \geq S$, K-means separates straight segments by placing cluster boundaries at bends. This helps us identify transitions between straight paths and bends, as well as effective road segments. From 6.3, distance metric, D contains minimal number of cross sectional paths if the following constraint $k \geq B + 1$ exists. From a code standpoint, $k \gtrsim 4N$, since k is initialised at $4N$.

Problem: Road Modelling

Consider a road that is represented as a sequence of points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, where each $\mathbf{x}_i \in \mathbb{R}^2$ is a 2D coordinate (x_i, y_i) on a flat plane (Haversine Distances are ignored as it is assumed to be negligible). These points are collected by sampling the road at regular intervals, such as every meter along its path. The road consists of S straight segments, which are nearly linear stretches, connected by $B = S - 1$ bends, where the road turns significantly. For example, a road with two straight segments ($S = 2$) has one bend ($B = 1$) connecting them, forming a pattern like straight-bend-straight. We can apply K-means clustering to group these points into k clusters, aiming to separate the straight segments from the bends.

Defining Key Concepts

Variance Within a Cluster (σ_j^2)

In K-means clustering, points are grouped in clusters, and each cluster C_j (where j labels the cluster, from 1 to k) is given a centroid $\boldsymbol{\mu}_j$. Mathematically, the variance for cluster C_j is defined as:

$$\sigma_j^2 = \frac{1}{\#C_j} \sum_{\mathbf{x}_i \in C_j} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2$$

The variance of each cluster σ_j^2 suggests how tightly packed the points are around their centroid. A smaller σ_j^2 means the points are closer to the centroid, forming a more compact cluster. Conversely, a large σ_j^2 indicates that the points are more scattered, making the cluster less compact.

Total Within-Cluster Sum of Squares (J)

K-means algorithm aim to make all clusters as compact as possible by minimising a the within-cluster sum of squares, where we denote as J. This is the sum of the squared distances from each point to its cluster's centroid, across all clusters:

$$J = \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in C_j} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2$$

It is important to note mathematically that the inner sum $\sum_{\mathbf{x}_i \in C_j} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2$ is the total squared distance for cluster C_j, which is $|C_j| \cdot \sigma_j^2$. On the other hand, the outer sum adds these contributions across all k clusters. Intuitively, the possibility of a small J means that points in each cluster are close to their centroids.

Curvature of the Road (κ)

Curvature, naturally denoted as κ, measures how bending at any point along its path and it further quantifies the rate of turning. If a line is straight, there is no twist or turn, thus $r \to \infty$, $\kappa(s) \approx 0$. For a circular arc of radius r, the curvature is constant and given by:

$$\kappa = \frac{1}{r}$$

Thus, a small r is associated with a tight turn, (e.g. a corner) and high curvature. The existence of a large r means low curvature. At bends, κ is higher, and points follow a curved path, which affects how they cluster.

Straight Road Segments vs Bended Road Segments

In a straight road segment, the curvature is $\kappa(s) \approx 0$, suggesting highly collinear points $\mathbf{x}_i$. This points are characterised by minor deviations from GPS sampling noise. If we form a cluster C_j from the points in one straight segment, we can expect the centroid $\boldsymbol{\mu}_j$ to be on the line.

Because the points are nearly collinear, their distances from the centroid are small. The primary deviation is mainly dominated from their positions along the line and compared to sampling noise. Variance definitely smaller than when a segment is curved significantly and σ_j^2 remains low, reflecting the compact, linear arrangement of points.

On the other hand, at a bend, the road curves, and the curvature $\kappa(s)$ increases significantly. For the sake of making the explanation simpler, a 90-degree turn ($\theta = \frac{\pi}{2}$) with $R = 10$, has a much larger σ_j^2 compared to a straight segment of the same 20 units, where the variance is mostly due to noise.

Behaviour with respect to Different Numbers of Clusters

- When $k = S$: K-means assigns one cluster to each straight segment. The boundaries between clusters align with the $B = S - 1$ bends, and each centroid lies along its straight segment. Since each cluster contains mostly collinear points, σ_j^2 is low, and J is minimised effectively.
- When $k > S$: Extra clusters may form. For example, a long straight segment might be split into two smaller clusters, each still with low variance, and that is still good for us. Alternatively, if a bend has many sampled points, an extra cluster might focus on the bend itself. However, $k = S$ is sufficient to ensure that cluster boundaries coincide with the bends, capturing the road's structure.

Conclusion

When applying K-means clustering to a road with S straight segments and $B = S - 1$ bends, setting $k \geq S$ is effective in minimising J thus forcing K-means algorithm to place cluster boundaries at the bends, especially those high value curves in a bid to keep cluster variance σ_j^2 low.

6.3.4.2 K-means Algorithm for route segments

Note: References sklearn.cluster.KMeans algorithm, rephrased in my own words.

Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^d$, select $4\#S$ points randomly to form: $C^{(0)} = \{\boldsymbol{\mu}_1^{(0)}, \boldsymbol{\mu}_2^{(0)}, \ldots, \boldsymbol{\mu}_{4\#S}^{(0)}\}$

while True :

Let $A^{(t)} = \{a_1^{(t)}, a_2^{(t)}, \ldots, a_n^{(t)}\}$, where:

$$\forall i \in \{1, \ldots, n\}, \quad a_i^{(t)} = \arg\min_{j \in \{1,\ldots,k\}} \|\mathbf{x}_i - \boldsymbol{\mu}_j^{(t)}\|^2$$

$$\|\mathbf{x}_i - \boldsymbol{\mu}_j^{(t)}\|^2 = \sum_{m=1}^{d} (x_{i,m} - \mu_{j,m}^{(t)})^2$$

Let $S_j^{(t)} = \{\mathbf{x}_i \in X \mid a_i^{(t)} = j\}, \quad \forall j \in \{1, \ldots, k\}$

$$\boldsymbol{\mu}_j^{(t+1)} \leftarrow \begin{cases} \frac{1}{|S_j^{(t)}|} \sum_{\mathbf{x}_i \in S_j^{(t)}} \mathbf{x}_i & \text{if } |S_j^{(t)}| > 0 \\ \boldsymbol{\mu}_j^{(t)} & \text{else} \end{cases}$$

Stop if:

$$A^{(t)} = A^{(t-1)} \vee \max_j \|\boldsymbol{\mu}_j^{(t+1)} - \boldsymbol{\mu}_j^{(t)}\| < \epsilon \vee t \geq T_{\max}$$

$t \leftarrow t + 1$

return $(A^{(t)}, C^{(t)})$

6.3.4.3 Evaluation and hyperparameter tuning to curvature as a stopping criterion

To generalise the K-means routine in 6.3.4.2 even further specific to route segments, we can look at the ratios of eigenvalues of the first axis compared to the second axis for each cluster, and use this as some sort of hyperparameter tuning technique to give us the best cluster count that is not statically $4\#S$.

For each 2 Dimensional Cluster, we can decompose the cluster into its eigenvalues. Let $\lambda_{j1} \geq \lambda_{j2}$ be its sorted eigenvalues.

If:

- $\lambda_{j1} \approx \lambda_{j2}$, the cluster is almost circular.
- $\lambda_{j2} \ll \lambda_{j1}$, the cluster is almost collinear and close to a straight line.

Theoretically, λ_{j2} reflects deviation from the straight line, and the deviation is orthogonal, and we can use the $\lambda_{j1} : \lambda_{j2}$ ratio to get an estimate of curvature. Thus, we introduce a threshold $\tau = 0.05$. The iterative process stops when clusters satisfy $\frac{\lambda_{j2}}{\lambda_{j1}} > \tau$.

6.3.4.4 K-Means Algorithm for route segments with hyper-parameter tuning

Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^2$, and $k_0 = 4\#S$. Let $\tau = 0.05$, $k \leftarrow k_0$, $C^{(0)} = \{\boldsymbol{\mu}_1^{(0)}, \boldsymbol{\mu}_2^{(0)}, \ldots, \boldsymbol{\mu}_{4\#S}^{(0)}\}$, $T_{\max} = 10000$

while True:

$C^{(0)} \leftarrow$ random sample of k points from X

$t \leftarrow 0$

while $t < T_{\max}$:

$$A^{(t)} \leftarrow \left\{ a_i^{(t)} = \arg\min_j \|\mathbf{x}_i - \boldsymbol{\mu}_j^{(t)}\|^2 \mid i = 1, \ldots, n \right\}$$

$$S_j^{(t)} \leftarrow \{\mathbf{x}_i \in X \mid a_i^{(t)} = j\}, \quad \forall j$$

$$\boldsymbol{\mu}_j^{(t+1)} \leftarrow \begin{cases} \frac{1}{|S_j^{(t)}|} \sum \mathbf{x}_i & \text{if } |S_j^{(t)}| > 0 \\ \boldsymbol{\mu}_j^{(t)} & \text{otherwise} \end{cases}, \quad \forall j$$

if $A^{(t)} = A^{(t-1)} \vee \max_j \|\boldsymbol{\mu}_j^{(t+1)} - \boldsymbol{\mu}_j^{(t)}\| < \epsilon$:

break

$t \leftarrow t + 1$

collinear $\leftarrow$ True

for all j with $|S_j^{(t)}| > 2$:

$$\Sigma_j \leftarrow \frac{1}{|S_j^{(t)}|} \sum (\mathbf{x}_i - \boldsymbol{\mu}_j^{(t+1)})(\mathbf{x}_i - \boldsymbol{\mu}_j^{(t+1)})^\top$$

$\lambda_{j1}, \lambda_{j2} \leftarrow$ sorted eigenvalues of Σ_j

if $\frac{\lambda_{j2}}{\lambda_{j1}} > \tau$:

collinear $\leftarrow$ False

break

if collinear == True:

break

$k \leftarrow k + 1$

return $(A^{(t)}, C^{(t)})$

6.3.5 Mathematical Analysis: Extension of K-means Centroids with Bearing

Building from 6.3.4, K-means favour partitions along straight line segments and setting boundaries at points of high curvature or bending in a bid to minimise the within-cluster variance, we will like to further evalute this point given additional information extracted from GPS data. Such additional information can include bearing of the bus (direction of the bus travel with relation to magnetic North), which can gives us further hints of road conditions crucial for later components. With a sufficiently large sample of GPS data collected, the statistical distribution of the bearing values over a road segment tells us more about the road geometry.

If a road is a straight line, the vehicle's bearing will cluster around two distinct values corresponding to the two opposite directions since the direction of travel is strictly forwards and 180 degrees of forwards (backwards). Hence, when applying K-means clustering to a huge set of bearing data of a straight road, the optimal number of clusters is expected to be 2. Conversely, if the road has a bend or junction, the variation in bearing will be greater. For a T or t-junction, the optimal K-means clustering may yield 4 clusters as the road geometry forces bearings to diverge into multiple directions. The directions should then be one pair of x and $180 - x$ degree points signifying two lines with sufficient data of the road. Furthermore, for a curved road, the spread in bearing values is even more pronounced, and the optimal number of clusters is expected to exceed 4 and the distances between the centroids in the space of all bearing values will be higher compared to those observed in straight or junction-dominated segments.

Thus, the number of clusters and the inter-centroid distances in the bearing space can be used as indicators of road conditions:

- **Straight Road:** Bearing clusters are centred around two opposing directions (approximately 180° apart), thus the optimal k-means k is 2.
- **T or T-Junction:** Bearing clusters are one pair of x and $180 - x$ degree points. The presence of additional travel directions leads to an optimal k of approximately 4.
- **Curved Road:** Increased variation in the bearing values results in either a higher optimal k or larger distances between the bearing centroids, signifying the curve.

Algorithm for Road Segment Analysis via Bearing-enhanced K-means

To better understand and classify the road segments based on the bearing information, we propose the following algorithm:

1. **Initial Clustering:** For each route segment, apply K-Means Clustering specific to the metric: bearing.
2. **Optimal Cluster Determination:**
   - Use silhouette score to determine the optimal number of clusters k^*.
3. **Segment Classification:**
   - If $k^* = 2$ and the centroids are approximately 180° apart, classify the segment as **Straight**.
   - If $k^* \approx 4$ and the centroid distribution reflects a distinct splitting of bearings typical of a T or T-junction, classify the segment as a **Junction**.
   - If $k^* > 4$ or the distances between centroids are significantly larger than those in the straight-road scenario, infer that the segment is **Curved**.
4. **Output:** For each road segment, report the optimal number of clusters k^*, the centroid positions in the bearing space, and the inferred road condition (Straight, Junction, or Curved).

Conclusion: With the following information, we get directional information of the route without being physically present leveraging off K-means's natural ability to aligns with dominant travel directions, and we can infer if a route segment is straight, a junction or a curve.

6.4.6: Imputation using Machine Learning

In place of the $\overline{H}_{m+1} - \overline{H}_m$ component in equation (6.3.3.8), we now explore two additional imputations to better approximate bus stop arrival and departure timings. This can be done using K Nearest Neighbours (KNN) and Multiple Imputation by Chained Equations (MICE). Define

$$\widehat{\Delta H}_m^{\text{KNN}} \quad \text{and} \quad \widehat{\Delta H}_m^{\text{MICE}}$$

as the imputed headway increments at bus stop m. Thus the imputed, effective departure headway can be represented as

$$H_m^{\text{dep}} \approx H_{m+1}(x_2) \; - \; \left| \sum_{t=1}^{T_{u_2}} \frac{D_f(t)}{\overline{v}_f(t)} \; - \; \sum_{t=1}^{T_{u_2}} \frac{D_b(t)}{\overline{v}_b(t)} \right| \; - \; \widehat{\Delta H}_m^{\text{KNN}} \tag{6.4.6.1}$$

$$H_m^{\text{dep}} \approx H_{m+1}(x_2) \; - \; \left| \sum_{t=1}^{T_{u_2}} \frac{D_f(t)}{\overline{v}_f(t)} \; - \; \sum_{t=1}^{T_{u_2}} \frac{D_b(t)}{\overline{v}_b(t)} \right| \; - \; \widehat{\Delta H}_m^{\text{MICE}} \tag{6.4.6.2}$$