



PEGASUS

Abbildung multimodaler Pendlerströme

Forschungskooperation TU Wien – AK Wien

Technische Dokumentation

Florian Pühringer, Selim Banabak, Robert Kalasek, Hans Kramar

Forschungsbereich Stadt- und Regionalforschung, Institut für Raumplanung



Stand: 23.10.2023

Inhalt

1.	Zweck dieser Dokumentation	3
2.	Berechnung der Fahrzeiten	3
2.1	Aufbau Erreichbarkeitsmodell	3
2.2	Verwendete Datengrundlagen.....	4
2.3	Bestimmung der Quellen und Ziele des Erreichbarkeitsmodells	5
2.4	Verkehrsgraph für den Fußverkehr.....	7
2.5	Verkehrsgraph Radverkehr	7
2.6	Verkehrsgraph MIV	8
2.7	Verkehrsgraph Öffentlicher Verkehr.....	9
3.	Aufbau Fahrzeitendatenbank und PGRouting.....	11
3.1	Datenbankstruktur.....	11
3.2	Datenintegration.....	12
3.3	Abfrage aus der Fahrzeitendatenbank.....	14
3.4	Nutzung der Datenbank für pgrouting.....	15
4.	Modal-Choice MNL	17
4.1	Ziel	17
4.2	Modellklasse	17
4.3	Modellspezifikation.....	18
4.4	Parameterschätzung.....	19
4.5	Modellselektion	19
4.6	Prediction mit dem MNL-Modell	20
5.	Literatur und Verweise	21

1. Zweck dieser Dokumentation

Die vorliegende Dokumentation ist als technische Dokumentation zu verstehen. Sie enthält notwendige Schritte, die zur Reproduktion der Ergebnisse notwendig sind, ohne dabei ins Detail zu gehen. Dieses Schriftstück wird ergänzt mit etlichen Skripten (R, sql, batch, ...), die ebenso als Teil der technischen Dokumentation zu sehen sind.

Inhaltliche Aspekte werden ausgespart – dazu sei auf die umfangreichen Folien der Abschlusspräsentation verwiesen, wo die einzelnen Module und Ziele im Detail vorgestellt werden.

Ablauf des Projekts

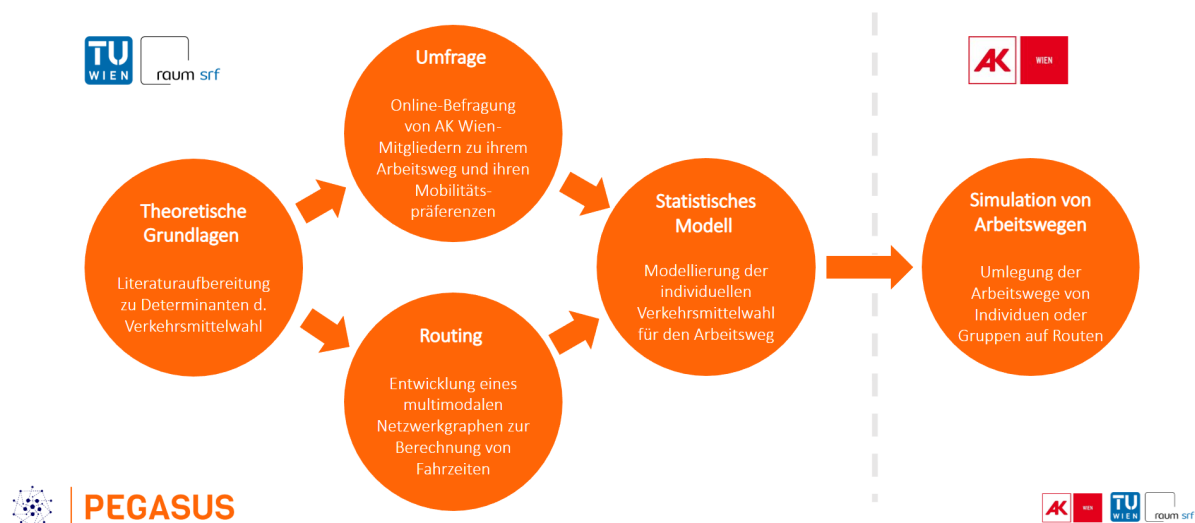


Abbildung 1: Projektdesign

2. Berechnung der Fahrzeiten

2.1 Aufbau Erreichbarkeitsmodell

Im Rahmen von PEGASUS lag der Fokus im Zuge der Fahrzeitberechnung auf einfachen und dafür flexibel anpassbaren Erreichbarkeitsmodellen. Dabei wurden folgende Ziele verfolgt:

- Verwendung frei verfügbarer Datengrundlagen
- Flexibilität hinsichtlich zukünftiger Adaptionen
- Realitätsnahe Modellierung

Für das Erreichbarkeitsmodell werden grundsätzlich folgende Datengrundlagen benötigt:

- Der **Verkehrsgraph** ist ein topologisches Netz aus Knoten und Kanten und beschreibt das jeweilige Verkehrssystem. Er beinhaltet eine Reihe an weiteren notwendigen Informationen (z.B. Raumwiderstände) sowie ergänzende Informationen zur Steigerung der Realitätsnähe der Modellierung (z.B. Abbiegerelationen, Einbahnrestriktionen, etc.)
- Die **Quellen** sind verortete Punktdaten und repräsentieren die Wohnstandorte der AK-Mitglieder.

- Die **Ziele** sind ebenso verortete Punktdaten und repräsentieren die Arbeitsstandorte der AK-Mitglieder.

Über die Kombination von Verkehrsgraph, Quellen und Ziele gelingt der Aufbau des Erreichbarkeitsmodells, welches Basis für die Berechnung der Fahrzeiten darstellt.

Allgemein muss unterschieden werden zwischen Berechnung der Fahrzeiten des Umfragesamples zum Training des statistischen Modells (**Trainingsdaten**) und der Berechnung der Fahrzeiten inkl. Geometrien für die Fahrzeitendatenbank (**Datenbank**).

Die Umsetzung erfolgte grundsätzlich mit ArcGIS Pro unter Nutzung der umfangreichen Werkzeuge der „Network Analyst Extension“.

2.2 Verwendete Datengrundlagen

Die verwendeten Datengrundlagen sind in untenstehender Tabelle aufgelistet. Das jeweilige Referenzjahr dient dabei nicht nur Quellenangabe, sondern auch als Indiz dafür, welche gebauten Infrastrukturen oder geplanten Öffentlichen Verkehre in der Analyse berücksichtigt wurden. Allgemein wurde versucht, möglichst aktuelle Datensätze zu verwenden, weshalb teilweise auch unterschiedliche Referenzjahre zu finden sind.

Tabelle 1

Thema	Verwendeter Datensatz	Jahr	Quelle
Quellen (Wohnstandorte)	Regionalstatistische Raster (Statistik Austria) Entfernungsskaliert von Wien ausgehend (250m – 500m – 1000m)	2020	Statistik Austria
Ziele (Arbeitsstandorte)	Regionalstatistische Raster (Statistik Austria) 250m – bebautes Gebiet innerhalb von Wien	2020	Statistik Austria
Verkehrsgraph Fußverkehr	GIP (Graphenintegrationsplattform)	2022	gip.gv.at
Verkehrsgraph Radverkehr	GIP	2022	gip.gv.at
Steigungsinformationen Radverkehr	Digitales Geländemodell Österreich, 10x10m	2015	Geoland.at – data.gv.at
Verkehrsgraph motorisierter Individualverkehr (MIV)	GIP (inklusive Turntable)	2022	gip.gv.at
Verkehrsgraph öffentlicher Verkehr – Fahrpläne	Maschinenlesbare Fahrplandaten im Format GTFS – Mobilitätsverbünde (Verbunddaten VOR und Eisenbahn)	2022	Mobilitätsverbünde – data.mobilitaetsverbunde.gv.at
Verkehrsgraph öffentlicher Verkehr – Zugangsdaten Fußverkehr	GIP	2022	gip.gv.at

Der GIP-Datensatz stellt eine der wesentlichen Bestandteile der Forschungs Kooperation dar. Dieser enthält das österreichweite Straßen- und Wegenetz (inkl. Bahn, Seilbahn, Fähren, etc.) mit einer großen Zahl an Attributen. Von größter Bedeutung ist dabei die Beschränkung der Befahrbarkeit je Verkehrsmittel. Diese wird in den Feldern „ACCESS_TOW“ und „ACCESS_BKW“ in bzw. gegen die Digitalisierungsrichtung angegeben. Diese wird mit einer Bitmaske modelliert, wobei Tabelle untenstehende Tabelle einen Überblick über die Bedeutung der Bits hinsichtlich der jeweiligen Verkehrsmittel gibt:

Tabelle 2

Bit Nr. (gemäß LSB-0 Nummerierung)	Wert	Verkehrsmittel
0	1	Fußverkehr
1	2	Radverkehr
2	4	MIV
3	8	Bus
4	16	Eisenbahn
5	32	Straßenbahn
7	128	Schiffsverkehr
...		

Insgesamt werden folgende 22 Verkehrsmittel berücksichtigt: Garbage Collection, Hazardous Combustibles, Camper, Car Ferry, Cable Railway, Rack Railway, Motorcycle, Trolley Bus, Motor Coach, Emergency Vehicle, Taxi, Truck, High Occupation Car, Ferry, Subway, Tram, Railway, Public Bus, Private Car, Bike, Pedestrian.

Sind mehrere Verkehrsmittel auf einer Kante zulässig, wird die Befahrbarkeit in dezimalen Werten angegeben, der in binäre Werte umgerechnet werden muss. So hat beispielsweise der dezimale Wert 2383631 binär die Aussage: Die Bits 0-3 sind gesetzt, also sind Fußgänger, Radfahrer, PKW und Bus erlaubt, die Bits 4-7 sind nicht gesetzt das bedeutet, dass Eisenbahn, Straßenbahn, U-Bahn und Schiffsverkehr nicht fahren dürfen. Die Bits ab Stelle 8 werden zwar anhand automatisierter Regeln gesetzt, werden aber in der GIP nicht gewartet und sind daher nicht relevant.

Zum einfacheren Handling wurde die im Rahmen von PEGASUS verwendete GIP dahingehend aufbereitet, als die dezimalen Werte umgerechnet und je Verkehrsmittel in eigener Spalte erfasst, ob dieses auf der jeweiligen Kante fahren darf (=1) oder nicht (=0). Diese Spalten wurden zur späteren Erstellung der Restriktionen verwendet.

2.3 Bestimmung der Quellen und Ziele des Erreichbarkeitsmodells

Die **Quellpunkte** repräsentieren die Wohnstandorte der AK-Mitglieder. Im Rahmen der Erstellung der Fahrzeiten als Trainingsdaten für das statistische Modell wurden die geokodierten Adressinformationen der vom IFES durchgeführten Umfrage berücksichtigt. Die Adressinformationen wurden von Seite der AK-Wien zur Versendung der Push-to-Web-Briefe verwendet und später mit den Antworten der Umfrage-Teilnehmer:innen verschnitten.

Die **Quellpunkte** der Fahrzeiten-Datenbank repräsentieren die Mittelpunkte des regionalstatistischen Rasters der Statistik Austria, wobei hier die Auflösung entfernungsskaliert wurde:



Tabelle 3

Quellstandort	Auflösung Regionalstatistischer Raster
Innerhalb Wiens	250m
Umkreis 25km um Wien	250m
Umkreis 25km bis 45km um Wien	500m
Umkreis 45km bis 60km um Wien	1000m
Außerhalb Umkreis 60km um Wien	Nicht berücksichtigt

Die Entscheidung der Entfernungsskalierung begründet sich mit der Reduktion der Ressourcen hinsichtlich Rechenzeit und Speicherbedarf.

Die **Zielpunkte** repräsentieren die Arbeitsstandorte der AK-Mitglieder. Als Trainingsdaten wurden hier wiederum für die Umfrage genutzte, geokodierte Adressinformationen verwendet. Die Zielpunkte der Fahrzeiten-Datenbank repräsentieren die Mittelpunkte des regionalstatistischen Rasters der Statistik Austria mit einer Auflösung von 250m, wobei nur laut Flächenmehrzweckkarte der Stadt Wien (data.gv.at) bebaute Gebiete innerhalb der Wiener Landesgrenzen berücksichtigt wurden.

Die Start- und Zielpunkte sind elementare Basis für die Fahrzeitberechnung. Zur Integration in das Erreichbarkeitsmodell müssen diese einer Netzkante des jeweiligen Verkehrsgraphen zugeordnet werden. Für diese Zuordnung wurden folgende Filter- und Suchkriterien am Verkehrsgraphen angewendet:

Tabelle 4

Modus	Zuordnung zu Kanten	Filter- Suchkriterium (SQL)
ÖV	Nur Netzkanten, auf denen Fußverkehr gestattet ist (keine Autobahnen, etc.). Keine topologischen Kanten der ÖV-Linien (Zugang nur über Verkehrsgraph und Stationen)	BKW_PEDESTRIAN = 1 OR TOW_PEDESTRIAN = 1
Radverkehr	Nur Netzkanten, auf denen Radverkehr gestattet ist	BKW_BIKE = 1 OR TOW_PEDESTRIAN = 1
Fussverkehr	Nur Netzkanten, auf denen Fußverkehr gestattet ist	BKW_PEDESTRIAN = 1 OR TOW_PEDESTRIAN = 1
MIV	Nur Netzkanten, auf denen Autoverkehr gestattet ist und die dem niederrangigen Netz angehören (kein direktes „aufspringen“ auf Autobahn, ...)	FRC != 0 AND (BKW_PrivateCar = 1 OR TOW_PrivateCar = 1)

2.4 Verkehrsgraph für den Fußverkehr

Der Verkehrsgraph für den Fußverkehr stellt den simpelsten in PEGASUS genutzten Verkehrsgraphen dar. Folgende Impedanzen und Restriktionen wurden dafür umgesetzt:

Tabelle 5

Fußverkehr	Festlegung je Merkmal	Function / Field Script (Python-Code)
Impedanz	Einheitlich 4,5km/h (=75 Meter pro Minute)	<code>WalkSpeed=75 !Length! / !WalkSpeed!</code>
Restriktion	Gehen nur auf Kanten, die laut GIP für Fußverkehr freigegeben sind.	<code>restricted = False if str(!TOW_Pedestrian!) in ["0"]: restricted = True restricted = False if str(!BKW_Pedestrian!) in ["0"]: restricted = True</code>

2.5 Verkehrsgraph Radverkehr

Im Bereich des Radverkehrs war ein Ziel, die (maßgebliche) Beeinflussung der Geschwindigkeit durch die Topographie (Steigungen) im Rahmen der Fahrzeitberechnung abzubilden. Dazu wurde in ArcGIS Pro ein Modell aufgebaut, welches der GIP je für jeden Kantenabschnitt diese Information hinzufügt. Dieses Modell durchläuft folgende Schritte:

Tabelle 6

Schritt #	Befehl	Funktionalität
1	Generate Points Along Lines	Erstellt einen neuen Punktdatensatz, welche entlang der GIP jede 200m einen Punkt enthält.
2	Split Line at Point	Die Kanten der GIP werden an den in Schritt 1 erstellten Punkten zerstückelt.
3	Calculate Field	Die Länge der neuen Kantenstücke wird berechnet.
4 und 5	Feature Vertices to Points	Je Kantenstück wird der Start- bzw. Endpunkt identifiziert.
5 und 6	Add Surface Information	Jeder Start- bzw. Endpunkt wird mit Informationen aus dem Digitalen Höhenmodell angereichert (Z-Wert)
6 und 7	Alter Field	Das Feld mit der Höheninformation wird umbenannt.
8	Join Field	Die Höheninformationen der Start- und Endpunkte je Kantenstück werden zusammengeführt.
9	Calculate Field	Die Steigung je Kantenstück wird berechnet. <code>(!Z_end!-!Z_start!) / !SlopeLength!</code>
10 und 11	Join Field + Select	Die Informationen werden zusammengeführt und in einen neuen Datensatz exportiert.

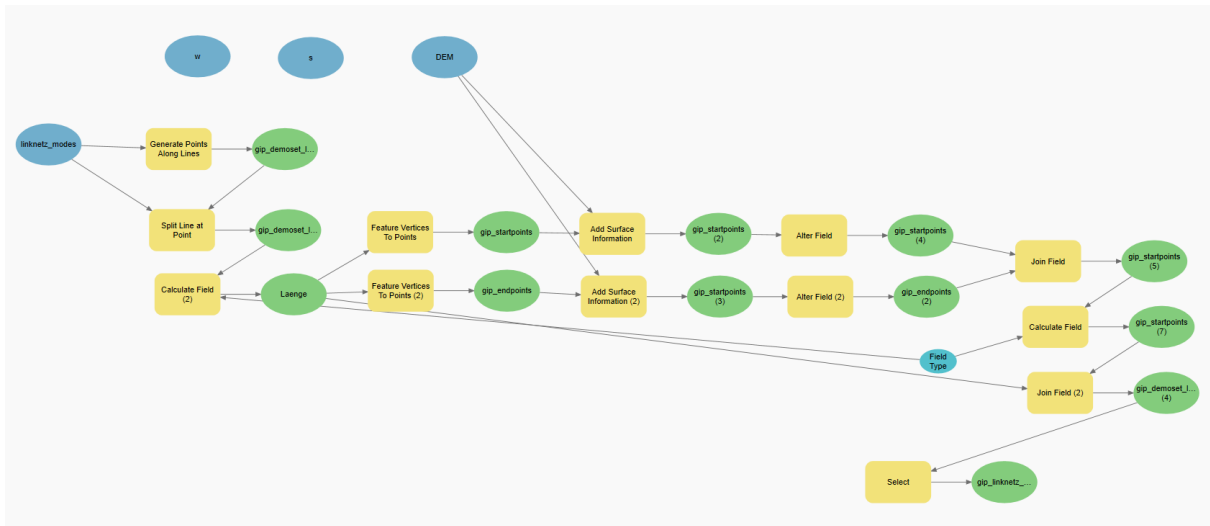


Abbildung 2

Die Steigungsinformationen wurden später im Zuge der Impedanz berücksichtigt. Die Wahl der Geschwindigkeit beruht dabei auf unterschiedlichen Quellen:

Tabelle 7

Radverkehr	Festlegung je Merkmal	Function / Field Script (Python-Code)
Impedanz	Basisgeschwindigkeit 18km/h, Steigungsabhängige Beeinflussung (langsam – schneller)	<pre>!Shape_Length! / (myCalc(!Slope!*100) * 16.66) def myCalc(slope): if slope <= -8: return 29 elif slope >=12: return 4 else: return 19-(25/20)*slope (!Shape_Length!) / (myCalc(!Slope!*-100) * 16.66) def myCalc(slope): if slope <= -8: return 29 elif slope >=12: return 4 else: return 19-(25/20)*slope</pre>
Restriktion	Radfahren nur auf Kanten, auf denen laut GIP Radfahren gestattet ist	<pre>restricted = False if str(!TOW_Bike!) in ["0"]: restricted = True restricted = False if str(!BKW_Bike!) in ["0"]: restricted = True</pre>

2.6 Verkehrsgraph MIV

Beim MIV wurden für die Trainingsdaten auch Abbiegeinformationen – konkret Ge- und Verbote – verarbeitet. Diese sind im GIP in der TurnUse-Tabelle enthalten und beschreiben je Kante die Möglichkeiten, auf andere Kanten zu wechseln.

Da die TurnUse-Tabelle für einige Relationen (abseits der Trainings-Relationen) fehlerhafte bzw. nicht brauchbare Informationen beinhaltet wurde bei der Erstellung der Fahrzeitendatenbank davon abgesehen, diese zu nutzen.



Tabelle 8

MIV	Festlegung je Merkmal	Function / Field Script (Python-Code)
Impedanz	Durchschnittsgeschwindigkeit laut GIP (belastete Kante)	<code>!Shape_Length! / (!SPEEDCAR_T! * 1000/60)</code> <code>!Shape_Length! / (!SPEEDCAR_B! * 1000/60)</code>
Restriktion	Fahren nur auf Kanten, die laut GIP für den MIV freigegeben sind	<code>restricted = False</code> <code>Select Case UCase([ONEWAY_CAR])</code> <code> Case "0","-1": restricted = True</code> <code>End Select</code> <code>restricted = False</code> <code>Select Case UCase([ONEWAY_CAR])</code> <code> Case "1","-1": restricted = True</code> <code>End Select</code>

2.7 Verkehrsgraph Öffentlicher Verkehr

GTFS (General Transit Feed Specification) ist ein digitales Standardformat zur Abbildung maschinenlesbarer Informationen im ÖV-System. Kernelement sind Fahrplandaten, welche gemeinsam mit räumlichen Informationen zu Stationen ein relationales Netz abbilden, dass für Erreichbarkeitsanalysen direkt in ArcGIS Pro genutzt werden kann. Ergebnis ist ein zeitgebundener und fahrplanfeiner ÖV-Graph. Um diesen routingfähig zu betreiben ist als zweite Information ein Graph notwendig, über welchen die Fusswegeverbindungen abgebildet werden können. Beim Aufbau des ÖV-Verkehrsgraphen wurde der GTFS-Graph und der Fusswegegraph (TLM) in einem weiteren Arbeitsschritt im Bereich der Stationen über Hilfskanten verbunden. Diese Hilfskanten bilden in weiterer Folge die Zustiege in das ÖV-System ab – so können auch die Anzahl der Umstiege erfasst werden.

Bei Analyseaufgaben auf Basis von Fahrplaninformation ist die Angabe eines Startzeitpunkts (in Form von Uhrzeit und Datum) von Bedeutung – durch den Fahrplan liefert eine Analyse der Fahrzeit von A nach B um 08:00 andere Ergebnisse als beispielsweise 3 Minuten später um 08:03, da möglicherweise Fahrten knapp erreicht bzw. verpasst werden oder zu diesem Zeitpunkt andere Fahrtkombinationen (Umsteigerelationen) möglich sind. Dieser Effekt beeinflusst vor allem in Gebieten mit weniger dichten Taktfolgen das Ergebnis: wird z. B. an einem Bahnhof mit 30-Minuten-Takt eine Fahrt knapp um 3 Minuten verpasst, enthält die berechnete Gesamtfahrzeit eine Wartezeit von 27 Minuten auf die nächste Verbindung, was das Analyseergebnis dementsprechend beeinflusst. Um diesen Effekt zu minimieren, wurden im Rahmen des Projekts sämtliche Analysen in knapp aufeinanderfolgenden Abständen von 9 Minuten für insgesamt sieben Startzeitpunkte werktags zwischen 07:00 und 07:54 wiederholt. Diese sieben Ergebnisse wurden in einem weiteren Schritt aggregiert und für jede Relation sowohl die minimale ÖV-Fahrzeit (inkl. Zu- und Abgangswege) als auch die durchschnittliche ÖV-Fahrzeit (ebenfalls inkl. Zu- und Abgangswege) gespeichert und weiterverarbeitet. Das folgende Beispiel illustriert diesen Analyseschritt: (vgl. Eisinger et. al 2023)

Beispielrelation	Analysezeitpunkt/Startzeitpunkt	Kürzeste Fahrzeit
Quellpunkt 1 – Zielpunkt 13	07:00 Uhr	21:23 min
Quellpunkt 1 – Zielpunkt 13	07:09 Uhr	23:01 min
Quellpunkt 1 – Zielpunkt 13	07:18 Uhr	19:54 min
Quellpunkt 1 – Zielpunkt 13	07:27 Uhr	17:43 min
Quellpunkt 1 – Zielpunkt 13	07:36 Uhr	20:43 min
Quellpunkt 1 – Zielpunkt 13	07:45 Uhr	19:23 min
Quellpunkt 1 – Zielpunkt 13	07:54 Uhr	22:13 min
Aggregat Relation 1 – 13		
Minimale Fahrzeit	17:43 min	
Durchschnittliche Fahrzeit	20:37 min	

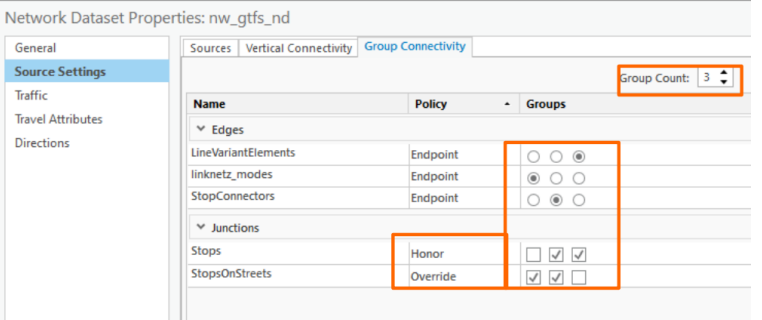
Abbildung 3

Die Entscheidung das Minimum der Iterationen als Fahrzeit zu verwendet begründet sich mit der Hypothese, dass Fahrgäste ihre Wartezeiten durch einen Blick in den Fahrplan (bzw. Fahrplan-Apps) optimieren. Das Verhältnis von minimaler Fahrzeit zu durchschnittlicher Fahrzeit wurde später als Proxy für die Frequenz des ÖV-Angebots verwendet (Taktung).

Der Aufbau des GTFS-Netzwerks wurde wiederum in ArcGIS Pro durchgeführt:

Tabelle 9

Schritt #1	Tool	Kurzbeschreibung
1	GTFS To Public Transit Data Model	Auf Basis der GTFS-Daten (je GTFS-Set – z.B. VOR – muss ein eigener Ordner angelegt werden) werden die neuen Features „Stops“ und „LineVariantElements“ angelegt, welche die ÖV-Topologie darstellen.
2	Connect Public Transit Data Model To Streets	Mit diesem Tool erfolgt die Erstellung von Hilfskanten, die von nun an die ÖV-Topologie mit dem Straßennetz (für Zu- und Abgänge im Fußverkehr) verbinden.
3	Network Dataset Build	Network Dataset erstellen
4	Group Connectivity	Über die Gruppenkonnektivitätseinstellungen wird gesteuert, in welcher Abfolge die unterschiedlichen Netzelemente bei Analysen durchlaufen werden.



Danach erfolgt wiederum die Konfiguration der Impedanzen und Restriktionen als Vorbereitung für die intermodale Fahrzeitberechnung:

Tabelle 10

ÖV	Festlegung je Merkmal	Function / Field Script (Python-Code)
Impedanz Fußverkehr	Einheitlich 4,5km/h (=75 Meter pro Minute)	WalkSpeed=75 !Length! / !WalkSpeed!
Impedanz ÖV	Definiert durch Fahrplan	Type = PublicTransit Same as along
Restriktion	Gehen nur auf Kanten, die laut GIP für Fußverkehr freigegeben sind.	restricted = False if str(!TOW_Pedestrian!) in ["0"]: restricted = True restricted = False if str(!BKW_Pedestrian!) in ["0"]: restricted = True

Für jede ÖV-Berechnung wurden auch die Anzahl der Umstiege bestimmt. Dazu wurde den Hilfskanten, die den GTFS-Graphen mit der GIP verknüpfen, ein Wert (=1) zugewiesen. Somit konnte die Anzahl der passierten Hilfskanten je Relation mitgezählt werden. Ein Wert von 2 entspricht dabei einer Verbindung ohne Umstieg, da die Hilfskanten nur je einmal für den Ein- und Ausstieg in den ÖV passiert wurden. Ein Wert von 4 entspricht dementsprechend einem Umstieg, 6 zwei Umstiegen, usw.

3. Aufbau Fahrzeitendatenbank und PGRouting

Die Fahrzeitendatenbank („Datenbank“) beinhaltet eine n:n Matrix über alle Relationen hinweg für die vier Hauptverkehrsmittel Fußverkehr (walk), Radverkehr (bike), motorisierter Individualverkehr (car) und öffentlicher Verkehr (transit).

Die Datenbank wurde als Postgres-Datenbank aufgesetzt. Um die Geometrien und später auch das Routing ordnungsgemäß betreiben zu können, ist die Installation der beiden Erweiterungen Postgis und Pgrouting zwingend erforderlich.

Name: pegasus_results

Passwort: Srf!admin

```
CREATE EXTENSION postgis;
CREATE EXTENSION pgrouting;
```

Für PGRouting sind weitere Schritte notwendig, die im Kapitel x.x näher beleuchtet werden.

3.1 Datenbankstruktur

Für die Ergebnisse wurde je Modus ein eigenes Schema erstellt.

bike.results	relation biketime the_geom	Quell-Ziel-ID / Statistik Austria ID R250 Fahrzeit in min (Limit 60min) Routengeometrie [3035]
car.results	relation cartime the_geom	Quell-Ziel-ID / Statistik Austria ID R250 Fahrzeit in min Routengeometrie [3035]
walk.results	relation walktime the_geom	Quell-Ziel-ID / Statistik Austria ID R250 Fahrzeit in min (Limit 60min) Routengeometrie [3035]

transit.results	relation min_tbl transitime r1 r2 r3 the_geom	Quell-Ziel-ID / Statistik Austria ID R250 Quelle des Min.Values Fahrzeit des Min.Values in min Fahrzeit Value 1 / Zeitpunkt T1 Fahrzeit Value 2 / Zeitpunkt T2 Fahrzeit Value 3 / Zeitpunkt T3 Routengeometrie des Min.Values [3035]
transit.anz_umst_je_rel	relation umst_coutf s_geom	Quell-Ziel-ID / Statistik Austria ID R250 Anzahl Umstiege je Route Geometrie der Umsteigekanten [3035] je Route

3.2 Datenintegration

Die Ergebnisse des Routings in ArcGIS Pro sind in einer ESRI-Geodatenbank und unterschiedlichen Feature-Datasets gespeichert. Dies begründet sich daher, da das Routing in 121 Samples unterteilt wurde, die iterativ durchlaufen wurden. So konnte bei technischen Problemen auf Teilergebnisse zurückgegriffen werden.

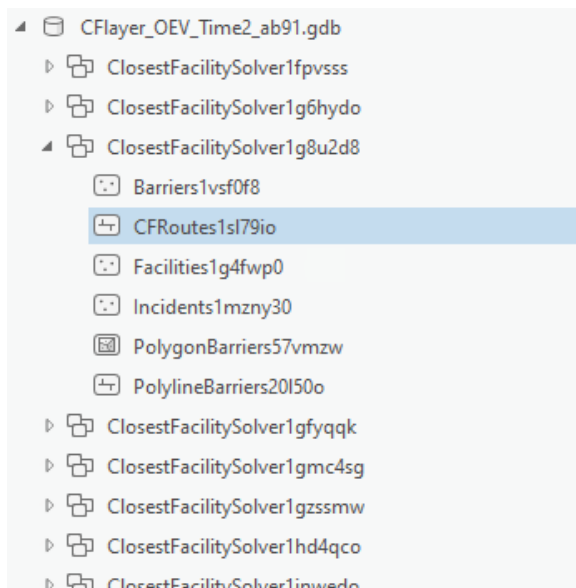


Abbildung 4

Die Integration der Daten aus den verschiedenen FeatureDatasets in die Postgres-Datenbank erfolgte mithilfe kleiner Batch-Skripte, die ogr2ogr-Befehle ausführten.

Dazu wurden in einem ersten Schritt mit `ogrinfo` die Bezeichnungen der unterschiedlichen FeatureDatasets der jeweiligen Geodatenbank ausgelesen.

Danach wurde mit folgendem Batch-Skript (beispielhaft für bike) der Import durchgeführt:

```
@echo off
setlocal

rem Skriptvariablen (anpassen):
```



```
set "PGUSER=postgres"
set "PGPASSWORD=Srf!admin"
set "PGDATABASE=pegasus_results"
set "PGHOST=localhost"
set "PGPORT=5432"
set "TARGET_SCHEMA=bike"
set "TARGET_TABLE=results"
set "TARGET_SRID=3035"
set "SOURCE_FILE=C:\temp\pegasus\RoutingDB_Results\CFLayer_Bike\CFLayer_Bike.gdb"

rem Liste der Layer-Namen (zuvor mit ogrinfo exportieren)
set "SOURCE_LAYERS=CFRoutes1b3v0wc CFRoutes4clzl4 CFRoutes1tdwtu0 CFRoutesjubt5k
CFRoutes13t0apo CFRoutes1pqzqi4 CFRoutes1mv5tic CFRoutes4ul6mo CFRoutesyzcx2c
CFRoutes4tpsns CFROUTESSAHGNO CFRoutes1ghw4k0 CFRoutes12kw7go CFRoutesx8vw7o
CFRoutes1utcnqc CFRoutesqpflw8 CFRoutes1ylud6k CFRoutesgen4q0 CFRoutes7d1x34
CFRoutesi6lho8 CFRoutespud4zk CFRoutespzazl4 CFRoutes1p0ifno CFRoutes2ti65k
CFRoutes14e773s CFRoutes1uq2hjk CFRoutes1w5u21g CFRoutes1q3zkto CFRoutesrn3gkk
CFRoutes47rmps CFRoutes1xg2r2w CFRoutes1drqq48 CFRoutesbnrdzs CFRoutes1b4ilvo
CFRoutesnqus04 CFRoutesm3vivk CFRoutesbqqilo CFRoutes12l9hjjg CFRoutes2tkyk4
CFRoutes1z05g64 CFRoutes15v1xtk CFRoutes1jklmr8 CFRoutes198qthc CFRoutes1o3sx14
CFRoutes1abnc0 CFRoutes1jujmq4 CFRouteshpq0a4 CFRoutes15n0jus CFRoutes15fu84g
CFRoutes7029xw CFRoutesd7uv0 CFRoutes1cb651c CFRoutesmu8mo8 CFRoutes2izwps
CFRoutes4bbagk CFRoutesb3c3mw CFRoutes2na69o CFROUTESTRKDO CFRoutes1j3mf8
CFRoutes1livltyw CFRoutes1sudku0 CFRouteslanfkn0 CFRoutesdsp534 CFRoutesdzyvbo
CFROUTESS8PDE0 CFRoutesler9q4k CFRoutesjmunfw CFRoutes4sjup0 CFRoutesbwojs4
CFRoutes1fig9r8 CFRoutes1ptieec CFRoutes1iyjj6s CFRoutesgyus74 CFROUTESSMIZXG
CFRoutesv9hvow CFRoutes10tyk0o CFROUTESTZAMV8 CFRoutesfnkxn8 CFRoutes1pwoap0
CFRoutesucloyw CFRoutes1bvp7v8 CFRoutesk7en7g CFRoutes1oeqfrc CFRoutesbry3ys
CFRoutes1ulxtx8 CFRoutes104rzpk CFRoutes13zev38 CFRoutes1dv50eo CFRoutesg535bc
CFRoutes1depjp0 CFRoutes1xh0vo CFRoutes1vjzjkc CFRoutes1oqn2r4 CFRoutes1u7i2po
CFRoutesb7btn0 CFRoutesmhilf8 CFRoutes26jw10 CFRoutes1nbwz94 CFRoutesmviy90
CFRoutes1s7tdpw CFRoutes1lo5vqc CFRoutesm48vw8 CFRoutes16wz2g8 CFRoutes1ej9hsc
CFRoutesakis5s CFRoutes1sdyke4 CFRoutes6gvhi0 CFRoutesi8lzs8 CFRoutes1h2mii0
CFRoutes131tk4o CFRoutes7bfdxw CFRoutes1qaeyk CFRoutes1fsifzs CFRoutesemix2k
CFRoutes1cbean8 CFRoutes1vjdel0 CFRoutesktlypo CFRouteso5npzg CFRoutes1dy76g
CFRoutesxgtwt8 CFRoutes1niy8gc CFRouteskqarfs CFRoutes16mcs4"

rem Schritt 1: aus Geodatenbank bzw. FeatureDatasets in PostGIS importieren
for %%G in (%SOURCE_LAYERS%) do (
    ogr2ogr -f "PostgreSQL" PG:"host=%PGHOST% port=%PGPORT% user=%PGUSER%
password=%PGPASSWORD% dbname=%PGDATABASE%" ^
        -nln "%TARGET_SCHEMA%.%TARGET_TABLE%" -nlt "MULTILINESTRING" -a_srs
"EPSG:%TARGET_SRID%" -lco "GEOMETRY_NAME=the_geom" ^
        -sql "SELECT name AS relation, Total_BikeTime AS biketime, SHAPE AS geom FROM
%%G" "%SOURCE_FILE%"
)

rem Skriptende
echo Importvorgang abgeschlossen!
```

pause

3.3 Abfrage aus der Fahrzeitendatenbank

Sind alle Modi in die Datenbank exportiert sollten noch entsprechende Primärschlüssel vergeben sowie Indizes erstellt werden, um spätere Datenbankoperationen zu beschleunigen.

Folgende Abfrage für den „Demonstrator Baden“ zeigt beispielhaft, wie die Daten genutzt werden können und wie eine Abfrage aufgebaut werden kann:

```
-- Abfrage auf Basis der Schmea car, transit, walk und bike für die gegebenen
Relationen-Paare erstellen

WITH relations AS (

    SELECT unnest(array['1kmN2865E4826 - 250mN280725E479400', '250mN278525E478500 -
250mN280750E479350', '250mN278600E478625 - 250mN280850E479250', '250mN278250E478950 -
250mN280775E479525', '250mN278600E478650 - 250mN280025E478700', '250mN278425E478675 -
250mN280375E479275', '250mN278625E478650 - 250mN281025E479225', '250mN278450E478675 -
250mN280400E479475', '250mN278525E478550 - 250mN280150E479225', '250mN278575E478625 -
250mN280700E479350', '250mN278475E478550 - 250mN280450E479550', '250mN278500E478625 -
250mN280600E479450', '250mN278475E478675 - 250mN280925E479425', [...]
'250mN278525E478375 - 250mN280925E479275', '250mN278575E478625 -
250mN280850E479400', '250mN278525E478600 - 250mN280050E479150', '250mN278525E478575 -
250mN280825E479475', '250mN278625E478475 - 250mN281000E478925', '250mN278575E478450 -
250mN280425E478875']) AS relation

)

INSERT INTO testing.testresults_demobaden (relation, mobmode, time, geom)

SELECT

    r.relation,

    r.mobmode,

    r.time,

    r.geom

FROM (

    SELECT relation, 'car' AS mobmode, cartime AS time, geom

    FROM car.results

    WHERE relation IN (SELECT relation FROM relations)

    UNION ALL

    SELECT relation, 'transit' AS mobmode, transittime AS time, geom

    FROM transit.results_t1

    WHERE relation IN (SELECT relation FROM relations)

    UNION ALL

    SELECT relation, 'walk' AS mobmode, walktime AS time, geom

    FROM walk.results

    WHERE relation IN (SELECT relation FROM relations)

    UNION ALL
```

```
SELECT relation, 'bike' AS mobmode, biketime AS time, geom
FROM bike.results
WHERE relation IN (SELECT relation FROM relations)
) AS r;
```

3.4 Nutzung der Datenbank für pgrouting

Mit Pgrouting können Fahrzeitberechnungen direkt in Postgres mithilfe von SQL durchgeführt werden. Der Funktionsumfang ist jedoch eingeschränkt – fahrplanbasiertes Routing für den öffentlichen Verkehr ist beispielsweise nicht möglich.

Um das Routing vorzubereiten empfiehlt es sich, zwei neue Schemata `pgrouting_in` (für die Datengrundlage – GIP) und `pgrouting_out` (für die Ergebnisse) zu erstellen.

In `pgrouting_in` wird zunächst das „gip_linknetz“ (z.B. mithilfe von QGIS) importiert. Dabei ist zu achten, dass dies zuvor in EPSG 3035 umprojiziert wird – auch die Erstellung eines spatial index ist empfehlenswert.

Danach kann der Aufbau des Routing-Netzwerks beginnen:

```
-- ALTER TABLE pgrouting_in.gip_linknetz ADD COLUMN source integer;
-- ALTER TABLE pgrouting_in.gip_linknetz ADD COLUMN target integer;
-- SELECT pgr_createTopology('pgrouting_in.gip_linknetz', 0.25, 'geom', 'fid');
```

Durch diesen Aufbau werden die beiden Spalten „source“ und „column“ erstellt. Die Funktion `pgr_createTopology` befüllt die beiden Spalten mit Informationen zur Topologie, die später als Basis für die Routing-Berechnungen dienen.

Die Start- und Zielpunkte sollten als eigene Geometrie vorliegen. Folgendes Skript nimmt Demo-Start- und Zielpunkte und berechnet die Routen zwischen diesen. Die Punkte werden dabei in einem ersten Schritt an die nächstgelegene (geeignete = erlaubte) GIP-Kante gesnappt:

```
-- Drop the table if it already exists
DROP TABLE IF EXISTS link_demo.routeresult_raw;

-- Create a table with start and end points and the shortest path between them
CREATE TABLE link_demo.routeresult_raw AS
WITH
-- Select start points and their geometries
starts AS (
    SELECT id, geom
    FROM link_demo.startpt
),
-- Select end points and their geometries
ziele AS (
```



```
SELECT id, geom
FROM link_demo.zielpt
)
-- Join the start and end points to the shortest path between them
SELECT starts.id AS start_id, ziele.id AS ziel_id, routingresult.*
FROM starts, ziele,
    pgr_dijkstra(
        'SELECT id, source, target, length / (speedcar_t * 1000 / 60) as cost,
length / (speedcar_b * 1000 / 60) as reverse_cost FROM
link_demo.gip_linknetz_demokarlsplatz WHERE tow_private_car = '1' OR
bkw_private_car = '1'',
        -- Select the nearest link to the start point as the source node
        (SELECT target FROM link_demo.gip_linknetz_demokarlsplatz WHERE
(tow_private_car = '1' OR bkw_private_car = '1') AND
ST_Distance(gip_linknetz_demokarlsplatz.geom, starts.geom) = (
        SELECT MIN(ST_Distance(gip_linknetz_demokarlsplatz.geom, starts.geom))
        FROM link_demo.gip_linknetz_demokarlsplatz
        WHERE (tow_private_car = '1' OR bkw_private_car = '1')
        )),
        -- Select the nearest link to the end point as the target node
        (SELECT source FROM link_demo.gip_linknetz_demokarlsplatz WHERE
(tow_private_car = '1' OR bkw_private_car = '1') AND
ST_Distance(gip_linknetz_demokarlsplatz.geom, ziele.geom) = (
        SELECT MIN(ST_Distance(gip_linknetz_demokarlsplatz.geom, ziele.geom))
        FROM link_demo.gip_linknetz_demokarlsplatz
        WHERE (tow_private_car = '1' OR bkw_private_car = '1')
        )),
        directed := true
    ) AS routingresult
-- Only keep rows where the start and end point ids match the minimum ids for their
respective tables
WHERE starts.id = (SELECT MIN(id) FROM link_demo.startpt WHERE id >= ziele.id)
    AND ziele.id = (SELECT MIN(id) FROM link_demo.zielpt WHERE id >= starts.id);
-- Create a table with the shortest paths' geometries
DROP TABLE IF EXISTS link_demo.routes_geom;
CREATE TABLE link_demo.routes_geom AS
    SELECT routeresult_raw.*, gip_edges.geom
    FROM link_demo.gip_linknetz_demokarlsplatz AS gip_edges
    JOIN link_demo.routeresult_raw ON routeresult_raw.edge = gip_edges.id;
```



```
-- Add a column to the table with the route relation string
ALTER TABLE link_demo.routes_geom ADD COLUMN relation TEXT;
UPDATE link_demo.routes_geom SET relation = start_id || ' - ' || ziel_id;
```

4. Modal-Choice MNL

4.1 Ziel

Im Zuge des Pegasus Projektes wurde ein Modell entwickelt, das erlaubt die Verkehrsmittelwahl am Arbeitsweg für die Mitglieder der Arbeiterkammer Wien (AK), mit einer bestimmten Wahrscheinlichkeit abzuschätzen. Um die Verkehrsmittelwahl zu erklären, kann maßgeblich auf die im Zuge des Projekts ermittelten Fahrzeiten (siehe GIS-Routing) zwischen Wohnorten und Arbeitsstätten zurückgegriffen werden. Weiters können der AK bekannte Charakteristika, der Mitglieder und deren Wohnorte herangezogen werden.

4.2 Modellklasse

Die Verkehrsmittelwahl stellt eine Entscheidung zwischen drei ungeordneten Kategorien dar: Motorisierter Individual Verkehr (MIV), Öffentlicher Verkehr (ÖV) und Unmotorisierter Individualverkehr (UIV)¹. Die Wahrscheinlichkeit einer bestimmten Verkehrsmittelwahl, kann entsprechend mit einem Multinomialen Logistische Regressionsmodell (MNL) geschätzt werden.

Sei, y eine kategoriale abhängige Variable mit $c+1$ nominalen Kategorien und y^*_1, \dots, y^*_{c+1} die assoziierten dummy Variablen. p_1, \dots, p_{c+1} sind die Wahrscheinlichkeiten einer Observation in eine der entsprechenden Kategorien zu fallen. Die Kategorie $c+1$ dient dabei als Referenzkategorie. Im konkreten Anwendungsfall ist das der MIV. Daher gilt: $p_{c+1} = 1 - p_1 - \dots - p_c$

Die Wahrscheinlichkeitsdicht ist gegeben durch

$$f(y|\pi_1, \dots, \pi_c) = \prod_{s=1}^{c+1} p^{y^*_s}$$

wobei $p_r = \pi_r / (1 + \pi_1 + \dots + \pi_c)$.

Die log-odds zwischen Kategorie r und der Referenzkategorie $c+1$ sind gegeben durch

$$\log(\pi_r) = \log\left(\frac{p_r}{p_{c+1}}\right) = \eta_r$$

Entsprechend werden die log-odds η_r als Linearkombination von k_r prädiktoren x_j modelliert

$$\eta_r = \sum_{j=1}^{k_r} \beta_0 + \beta_1 x_1 + \dots + \beta_{k_r} x_{k_r}$$

Siehe Fahrmeir, L., Kneib, T., & Lang, S. (2007a). für eine detaillierte Beschreibung von MNL-Regressionen.

¹ Die Kategorie "Zu Fuß" musste aufgrund zu geringer Fallzahlen ausgeschlossen werden.

4.3 Modellspezifikation

Das MNL-Modell kann als Spezialfall eines Structured Additive Regression (STAR) Modells interpretiert werden (Fahrmeir et al., 2007b). STAR-Modelle erlauben unter anderem η_r als Summe nichtlinearer Funktionen der Prädiktoren $f(z_j)$ darzustellen. Diese können sowohl zur Modellierung nichtlinearer Effekte metrischer- und kategorialer Prädiktoren als auch deren Interaktion (Pedersen et al. 2019) verwendet werden. Nichtlineare Effekte metrischer Variablen werden im Folgenden als Smooth Terms bezeichnet und nichtlineare Effekte kategorialer Variablen als Random Effects. Interaktionen können entsprechend Pedersen et al. (2019) sowohl als Shared Global Trend + Group Level Trend with Shared Smoothness (GS) oder Group Level Trend with Shared Smoothness (S) modelliert werden.

Die Verwendeten Prädiktoren können in 3 Kategorien eingeteilt:

- Persönliche Charakteristika
- Streckeninformationen
- Wohnort/Arbeitsort Charakteristika

Die konkret verwendeten Variablen, Skalierungen und Beschreibungen sind in Tabelle 11 aufgelistet. Das angehängte File *pegasus_techdoc_ANNEXA.xlsx* enthält zusätzlich die Wertebereiche, Ausprägungen und Quellen der Variablen.

Tabelle 11: Erklärende Variablen MNL-Modell

Variable	Einheit	Beschreibung	JSON datatype
AcTi	Binär	Akademischer Titel vorhanden	String
Age	Jahre	Alter	Number
BauDichte_H	Anteil	Baudichte der Rasterzelle am Wohnort	Number
BauDichte_W	Anteil	Baudichte der Rasterzelle am Arbeitsort	Number
BevDichte_H	hws/m2	Bevölkerungsdichte am Wohnort auf Basis der Hauptwohnsitze der Rasterzelle	Number
BevDichte_W	hws/m2	Bevölkerungsdichte am Arbeitsort auf Basis der Hauptwohnsitze der Rasterzelle	Number
Changes_cat	Ordinal	Umstiegsategorien: Direct: Umstiege= 0; Often: Umstiege >1; Once: Umstiege = 1	String
DistanzMet_H	km	Distanz (Luftlinie) zur nächsten hochrangigen ÖV Station am Wohnort	Number
DistanzMet_W	km	Distanz (Luftlinie) zur nächsten hochrangigen ÖV Station am Arbeitsort	Number
DistAutoba_H	km	Distanz zur nächsten Autobahnauffahrt am Wohnort	Number
DistAutoba_W	km	Distanz (Luftlinie) zur nächsten Autobahnauffahrt am Arbeitsort	Number
DistParkRi_H	km	Distanz (Luftlinie) zum nächsten Park&Ride-Standorte am Wohnort	Number
Dtime_oev	km	Distanz (Luftlinie) zum nächsten Park&Ride-Standorte am Arbeitsort	Number
EEshare_H	Prozent	Anteil Alleinerziehende Rasterzelle	Number
Gender.W	Binär	Weibliches Geschlecht	Number
Gruppe.MIG	Nominal	Migrationshintergrund	String



HHsize_H	Anzahl Personen	Durchschnittliche Haushaltsgröße in der Rasterzelle	Number
Hochschule_share_H	Prozent	Anteil Personen mit Hochschulabschluss in der Rasterzelle	Number
lat_home	Grad	Latitude Wohnort	Number
lat_work	Grad	Latitude Arbeitsort	Number
lon_home	Grad	Longitude Wohnort	Number
lon_work	Grad	Longitude Arbeitsort	Number
oenace_abs	Nominal	ÖNACE-Abschnitte	String
OEV_MM_R	Verhältnis	ÖV Intervall Indikator: Verhältnis durchschnittliche ÖV Fahrzeit zur Minimalen	Number
PMKshare_H	Prozent	Anteil Paare mit Kind an allen Haushalten der Rasterzelle	Number
PPK	Binär	Parkpickerl-Konflikt (Wenn Wohnbezirk ungleich Arbeitsbezirk)	Number
radinfra_H	Meter	Länge Radinfrastruktur am Wohnort	Number
radinfra_W	Meter	Länge Radinfrastruktur am Arbeitsort	Number
Rtime_bike	Verhältnis	Verhältnis Fahrzeit Rad/Fahrzeit Auto	Number
Rtime_oev	Verhältnis	Verhältnis Fahrzeit ÖV/Fahrzeit Auto	Number
Total_BikeTime	Minuten	Gesamte Fahrzeit Rad	Number
Total_BikeTime_LN	log(Minuten)	Logarithmus der Gesamten Fahrzeit Rad	Number
UR_TYP_NAME_H	Nominal	Statistik Austria Urban-Rural-Typologie.	String
UR_TYP2_H	Nominal	Statistik Austria Urban-Rural-Typologie.	String

4.4 Parameterschätzung

Die Parameter des Modells werden mithilfe eines bayesianischen MCMC Sampling Verfahrens geschätzt. Ein vorgeschaltetes Gradient Boosting wird verwendet um geeignete Startwerte für den MCMC Sampler zu finden. Das Modell wird mithilfe des R-Packages 'BAMLSS' (A Lego Toolbox for Flexible Bayesian Regression (and Beyond) kalibriert. Siehe Umlauf et al. (2021) oder <http://www.bamlss.org/> für detaillierte Information zum entsprechenden R-Package. Nach einem burnin von 2000 Ziehungen wurden 5000 Ziehungen zur Approximation der posterior Verteilung der Modellparameter herangezogen.

Die Parameterschätzung basiert auf der im Zuge des Pegasus Projekts gemeinsam mit dem IFES durchgeführten Befragung zur Verkehrsmittelwahl am Arbeitsweg. Aus der Befragung ergeben sich 2200 Beobachtungen, die für die Kalibrierung herangezogen werden können. Siehe Anhang A für eine detaillierte Beschreibung der abgefragten Variablen. Auf Basis der Abgefragten (bzw der AK bekannten Wohn- und Arbeitsorte) konnten Routen und Fahrzeiten und weitere Streckenspezifika ermittelt werden (Siehe Kapitel GIS-Routing). Weiters wurden mehrere Variablen mit Informationen zum Wohn- bzw Arbeitsort ergänzt.

4.5 Modellselektion

Im Zuge des Projektes wurden 10 Modelle mit zunehmendem Komplexitätsgrad geschätzt. Diese wurden anschließend anhand zweier Metriken verglichen. 1) dem Deviance Information Criterion (DIC), einer Generalisierung des Akaike Information Criterion und 2) dem Anteil der Beobachtungen bei denen das Modell die tatsächliche Verkehrsmittelwahl als die wahrscheinlichste ausgibt. Zusätzlich wurde das zweite Kriterium auch für einen Vergleich auf Basis der out-of-sample

prediction Qualität herangezogen. Dazu wurden die Modelle jeweils iterativ mit 90% der Daten trainiert und die prediction Qualität durch Anwendung auf die restlichen 10% ermittelt.

4.6 Prediction mit dem MNL-Modell

Zur Vorhersage von Modal-Choice Wahrscheinlichkeiten mithilfe des kalibrierten MNL-Modells kann die `bamlss` Funktion `predict.bamlss()` verwendet werden. Um die Verwendbarkeit des MNL-Modells auch unabhängig von der R Fähigkeit der Anwendungsumgebung sicherzustellen, wurde ein Docker Container (<https://www.docker.com/>) Software-Paket gebaut. Der Docker Container kann via rest API auf dem Port 5762 unter dem endpoint `/pega` angesprochen werden und verarbeitet POST requests. Verarbeitet werden JSON files die entsprechend der Variablenbeschreibung in Anhang A aufgebaut sind. Retourniert wird ebenfalls ein JSON file mit den ursprünglichen Variablen plus den geschätzten Wahrscheinlichkeiten der drei Verkehrsmodi. Achtung, aufgrund von Rundungsfehlern bei der Konvertierung in JSON ergänzen sich die drei Wahrscheinlichkeiten nicht exakt auf eins. Unvollständige Beobachtungen werden automatisch aus dem Datensatz entfernt. Liegt der Wert einer numerischen Variable außerhalb des ursprünglichen Wertebereichs bei der Kalibrierung, wird anstelle des Spline-Modells auf ein Lineares Modell zurückgegriffen. Dadurch kann es sein, dass sich die Reihenfolge und Zahl der Beobachtungen verändert. Die einzelnen Beobachtungen sind aber weiterhin mittels der ID-Variable zu identifizieren.

Der Workflow kann wie folgt skizziert werden:

1. User sendet JSON file per POST request an den Port 5762 unter dem endpoint `/pega`
2. Die Daten werden eingelesen und auf die benötigten Variablen reduziert (unnötige Variablen werden gelöscht)
3. Unvollständige Beobachtungen werden gelöscht
4. Variablen der einzelnen Beobachtungen werden in das benötigte Datenformat umgewandelt (Beispiel numerisch -> numeric; string -> factor)
5. Die Wertebereiche der Variablen werden mit dem Datensatz aus der Kalibrierung abgeglichen, Beobachtungen werden entsprechend in Zwei Datensätze getrennt
 - a. Außerhalb des Wertebereich
 - b. Innerhalb des Wertebereichs
6. Das MNL-Modell wird zur Vorhersage der Verkehrsmodi herangezogen
 - a. Das komplexere MNL-Modell wird zur Vorhersage der erwarteten Verkehrsmodi innerhalb des Wertebereichs verwendet
 - b. Das lineare MNL-Modell wird zur Vorhersage der erwarteten Verkehrsmodi außerhalb des Wertebereichs verwendet
 - c. Die ergänzten Datensätze werden wieder zusammengefügt
7. Der Datensatz mit den geschätzten Wahrscheinlichkeiten wird als JSON retourniert

Die Struktur des JSON files, dass an den Docker Container gesendet wird kann wie in Tabelle 11 skizziert werden.

Tabelle 11: Input Datenstruktur

JSON Struktur Beispiel
<pre>[{"key":0.01,"key2":"b",..., "key99":0.02},</pre>

```
....  
{"key":0.03,"key2":"c",.....,"key99":0.04}  
]
```

5. Literatur und Verweise

Eisinger, A., Kopf, E., Peter, M., Uffer, S., Berger, M., Scheuvs, R., Pühringer, F., Sandner, V., Mitteregger, M., Soteropoulos, A. (2023). Räumlich-differenzierte Auswirkungen des automatisierten Fahren. Bundesamt für Straßen ASTRA.

Fahrmeir, L., Kneib, T., & Lang, S. (2007a). Kategoriale Regressionsmodelle. Regression: Modelle, Methoden und Anwendungen, 235-252.

Fahrmeir, L., Kneib, T., & Lang, S. (2007b). Strukturiert-additive Regression. Regression: Modelle, Methoden und Anwendungen, 399-443.

Klein, N., Kneib, T., & Lang, S. (2013). Bayesian structured additive distributional regression (No. 2013-23). Working Papers in Economics and Statistics.

Pedersen, E. J., Miller, D. L., Simpson, G. L., & Ross, N. (2019). Hierarchical generalized additive models in ecology: an introduction with mgcv. PeerJ, 7, e6876.

Umlauf N, Klein N, Simon T, Zeileis A (2021). "bamsls: A Lego Toolbox for Flexible Bayesian Regression (and Beyond)." Journal of Statistical Software, 100(4), 1–53. doi:10.18637/jss.v100.i04.