FastPrice

Nosso objetivo é revolucionar a forma como os usuários escolhem serviços de transporte, garantindo que sempre encontrem a melhor opção em termos de custo-benefício. Para isso, estamos desenvolvendo o FatsPrice, um site inteligente que compara em tempo real os preços das corridas em diferentes plataformas de transporte.

Atuamos no setor de transporte, onde identificamos um espaço no mercado: a dificuldade dos usuários em comparar os preços de corridas entre diferentes aplicativos, como Uber e 99. Nossa missão é transformar a experiência do usuário, fornecendo uma plataforma intuitiva, confiável e eficiente, que o ajude a tomar decisões rápidas, econômicas e inteligentes.

Acreditamos na inovação e agilidade como diferencial competitivo e, por isso, estamos investindo na integração de IA para oferecer um serviço que garanta o melhor preço em tempo real.

Além do segmento de transporte rápido, planejamos expandir nossa solução para o mercado aéreo, permitindo que os usuários comparem preços de passagens e encontrem o melhor preço para suas viagens.

Nosso compromisso é com a transparência, acessibilidade e inovação, proporcionando uma ferramenta fácil de usar, rápida e confiável. Ao garantir que nossos usuários tenham acesso às melhores opções de transporte de maneira simples.

Com um modelo de negócios baseado em planos de assinatura e parcerias estratégicas, nossa visão é nos tornarmos referência no setor, oferecendo a melhor plataforma de comparação de preços.

Ciência de Dados e Big Data

Objetivo: ENTENDER OS DADOS

- Coletar dados Iniciais
- Descrever os Dados
- Explorar os Dados
- Verificar a Qualidade dos Dados

Rideaddress_v1

Ao analisarmos a tabela rideaddress_v1, identificamos que ela pode ser utilizada para determinar os locais de início e fim das corridas. Observamos que a coluna RideID sempre se repete duas vezes para o mesmo ID, e que a coluna RideAddress segue um padrão em que um valor aparece primeiro e outro logo abaixo. Após uma análise detalhada, concluímos que os registros com o mesmo RideID representam a mesma corrida, sendo que a primeira ocorrência (1 em RideAddress) corresponde ao ponto de partida e a segunda (2 em RideID) ao destino final.

A coluna **Address**, utilizada para identificar o endereço da corrida, não está padronizada corretamente, tornando difícil a localização exata em alguns casos. Como alternativa, a equipe decidiu utilizar as colunas **Lat** (Latitude) e **Lng** (Longitude), para encontrar o endereço do início e fim da corrida.

No entanto, os valores dessas colunas estão em um formato incompatível com serviços de geolocalização, como o Google Maps. O primeiro passo será padronizar essas colunas para obter os endereços corretos de forma confiável.

RideAddre	Address	Street	Number	Neighborhood	City	State	Lat	Lng	RideAddre	RideID
2334277	Rua JoÃ£o Pinheiro,	Rua JoÃ£o Pinheiro	585	Rua JoÃ£o Pinheiro	9	Brasil	-26.329.754.299.999.900	-48.840.427.999.999.900	1	1183200
2334278	Av. Dr. Nereu Ramc	Av. Dr. Nereu Ramos, 450	450	9	9	9	-262.554.657	-486.434.197	2	1183200
2334279	Rodovia Rafael da R	Rodovia Rafael da Rocha P	1883	Rodovia Rafael da Rocl	9	Brasil	-274.919.788	-48.528.287.999.999.900	1	1183201
2334280	Angeloni Ingleses (Fl	Angeloni Ingleses (FlorianÃ	6375	9	9	9	-274.371.486	-4.839.824.309.999.990	2	1183201
2334281	Rua BarÃ£o do Rio I	Rua BarÃ£o do Rio Branc	12	Rua BarÃ£o do Rio Br	9	Brasil	-198.495.799	-44.019.915.999.999.900	1	1183202
2334282	R. AntÃ´nio de Albu	R. AntÃ´nio de Albuquerqu	1080	9	Belo Horizonte	Minas Gerais	-19.936.899	-439.401.603	2	1183202
2334283	Tv. Duzentos e Sesse	Tv. Duzentos e Sessenta e	72	9	9	9	-239.624.233	-4.625.465.759.999.990	1	1183203
2334284	Semar Supermercadc	Semar Supermercados Bert	2141	9	9	9	-238.373.074	-461.321.725	2	1183203
2334285	Rua Argentina, 160 -	Rua Argentina	160	Rua Argentina	9	Brasil	-109.198.019	-37.077.441.799.999.900	1	1183204
2334286	R. SimeÃ£o Aguiar,	R. SimeÃ£o Aguiar, 430 - I	430	9	Aracaju	Sergipe	-109.071.288	-370.877.194	2	1183204
2334287	Rua JoÃ£o CirÃ-lo d	Rua JoÃ£o CirÃ-lo de Oliv	5	Rua JoÃ£o CirÃ-lo de (	9	Brasil	-228.735.015	-435.714.019	1	1183205
2334288	Av. CesÃ¡rio de Mel	Av. CesÃ¡rio de Melo, 108	10809	9	9	9	-229.173.726	-436.330.438	2	1183205
2334289	Avenida AngÃ©lica,	Avenida AngÃ©lica	2573	Avenida AngÃ©lica	9	Brasil	-235.542.807	-46.662.731.799.999.900	1	1183206
2334290	Av. Sen. TeotÃ´nio V	Av. Sen. TeotÃ´nio Vilela,	4029	9	9	9	-237.342.905	-466.986.279	2	1183206
2334291	Tv. Duzentos e Sesse	Tv. Duzentos e Sessenta e	72	9	9	9	-239.624.233	-4.625.465.759.999.990	1	1183207

Depois que limpamos e formatamos as colunas Lat e Lng, chegamos nesse resultado:

Quando fomos mexer na tabela vimos que ao todo são 1 milhão de linhas, o que dificulta a utilização de filtros, criação de formulas entre outras consultas, sendo assim, criamos um script em pandas que exclui as colunas Address, Street, Number, Neighborhood, City e State.

```python
import pandas as pd

rideaddress_v1 = pd.read_csv('rideaddress_v1.csv', sep=';', low_memory=False)

rideaddress_v1 = pd.DataFrame(rideaddress_v1)

rideaddress_v1 = rideaddress_v1.drop(columns=["Address", "Street", "Number", "Neighborhood", "City", "State"])

rideaddress_v1.to_csv('rideaddress_v1_Limpo.csv', sep=';', index=False)
```

Essa decisão foi tomada porque entendemos que as coordenadas de latitude e longitude já fornecem a localização exata do início e do fim de cada corrida.

O próximo passo será unificar as colunas Lat e Lng e ajustar para o padrão com que o serviço de geolocalização consiga ler.

```python
import pandas as pd

# Carregar o CSV
df = pd.read_csv('rideaddress_v1.csv', sep=';', low_memory=False)


df["Lat"] = df["Lat"].astype(str)
df["Lng"] = df["Lng"].astype(str)


def ajustarCoordenadas(value):

    limpar = value.replace('.', '')
    formatar = limpar[:3] + "." + limpar[3:]
    return formatar

df["Lat"] = df["Lat"].apply(ajustarCoordenadas)
df["Lng"] = df["Lng"].apply(ajustarCoordenadas)

df["Coordenadas"] = df["Lat"] + ", " + df["Lng"]

df.to_csv('FormatarLatLng.csv', sep=';', index=False)

print("Arquivo 'FormatarLatLng.csv' salvo com sucesso!")
```

Com esse script conseguimos chegar nesse resultado

RideAddressID	Lat	Lng	RideAddressTypeID	RideID	Coordenadas
2334277	-26.329.754.299.999.900	-48.840.427.999.999.900	1	1183200	-26.329754299999900, -48.840427999999900
2334278	-262.554.657	-486.434.197	2	1183200	-26.2554657, -48.6434197
2334279	-274.919.788	-48.528.287.999.999.900	1	1183201	-27.4919788, -48.528287999999900
2334280	-274.371.486	-4.839.824.309.999.990	2	1183201	-27.4371486, -48.39824309999990
2334281	-198.495.799	-44.019.915.999.999.900	1	1183202	-19.8495799, -44.019915999999900
2334282	-19.936.899	-439.401.603	2	1183202	-19.936899, -43.9401603
2334283	-239.624.233	-4.625.465.759.999.990	1	1183203	-23.9624233, -46.25465759999990
2334284	-238.373.074	-461.321.725	2	1183203	-23.8373074, -46.1321725
2334285	-109.198.019	-37.077.441.799.999.900	1	1183204	-10.9198019, -37.077441799999900

Dessa forma conseguimos chegar nas coordenadas exatas.

RideAddressID	Lat	Lng	RideAddressTypeID	RideID	Coordenadas	
2334277	-26.329.754.299.999.900	-48.840.427.999.999.900	1	1183200	-26.329754299999900, -48.840427999999900	R. João Pinheiro, 585 - Floresta, Joinville - SC, 89210-170
2334278	-262.554.657	-486.434.197	2	1183200	-26.2554657, -48.6434197	Rocio Grande, São Francisco do Sul - SC, 89240-000
2334279	-274.919.788	-48.528.287.999.999.900	1	1183201	-27.4919788, -48.528287999999900	Rod. Rafael da Rocha Pires, 1882-1916 - Sambaqui, Florianópolis - SC, 88051-001
2334280	-274.371.486	-4.839.824.309.999.990	2	1183201	-27.4371486, -48.39824309999990	SC-403, 6375 - Ingleses Norte, Florianópolis - SC, 88058-001
2334281	-198.495.799	-44.019.915.999.999.900	1	1183202	-19.8495799, -44.019915999999900	R. Barão do Rio Branco, 23-1 - Nacional, Contagem - MG, 32185-070
2334282	-19.936.899	-439.401.603	2	1183202	-19.936899, -43.9401603	R. Antônio de Albuquerque, 1069-973 - Funcionários, Belo Horizonte - MG, 30112-011

Rideaddress_v3

RideEstimativeID	RideID	ProductID	WaitingTime	Price	FareID	Selected	RideReasonSelectedEstimativeID	Fee
8619946	1183200	Flash	8	89.00	c6aaac64-5f89-4fc4-8b66-0251ec1c78a8	0	NULL	0.00
8619947	1183200	UberX	6	89.00	ff3cc941-93a8-4d0e-a274-bb988576d7d4	0	NULL	0.00
8619948	1183200	Comfort	10	116.50	d7708871-2f2c-447d-81e6-a2d121863a2f	0	NULL	0.00
8619949	1183200	poupa99	5	170.21	NULL	0	NULL	0.00
8619950	1183200	pop99	7	170.21	NULL	0	NULL	0.00
8619951	1183200	turbo-taxi	6	151.05	NULL	0	NULL	0.00
8619952	1183200	regular-taxi	6	151.05	NULL	1	4	0.00

Na tabela **rideaddress_v3**, é possível verificar quais serviços estavam sendo analisados para a corrida correspondente na tabela **rideaddress_v1**. Essa comparação pode ser realizada utilizando a coluna **RideID**, que é comum em ambas as tabelas.

Dessa forma a equipe conseguirá verificar o valor da corrida pela visão de várias empresas e serviços diferentes, fazendo em paralelo o valor da corrida pela distância percorrida.

Um problema que verificamos, é que na tabela **rideaddress_v3,** existem milhares de linhas, o que pode prejudicar o tempo do projeto, sendo assim, decidimos verificar quais tipos de serviços existem na coluna ProductID.

```python
import pandas as pd

Valores_V3 = pd.read_csv('rideestimative_v3.csv', sep=';', low_memory=False)

Valores_V3 = Valores_V3['ProductID'].drop_duplicates()

print(Valores_V3)
```

```
0                  Flash
1                  UberX
2                Comfort
3                poupa99
4                  pop99
5              turbo-taxi
6            regular-taxi
14                   Moto
16              UberFlash
33             Uber Flash
42                  Black
48                    Bag
49              Black Bag
50                   Taxi
51             Uber Promo
67             Flash Moto
120              WPP-42-1
121              WPP-1-1
122              WPP-5-5
131              WPP-7-6
223                top99
595             comfort99
1208           delivery99
1726          UberX Promo
2531          Flash Bikes
1158448        Prioridade
1247652       Uber Planet
1247956    Comfort Planet
1555848          Original
1585439             PTaxi
1708318        Moto Flash
1947840   delivery-moto99
1947938            moto99
1998041    UberX Priority
Name: ProductID, dtype: object
```

Com essa visualização conseguimos ver quais serviços temos disponíveis para analisar, quais serão usados e quais serão tirados da tabela.

Dessa forma teremos uma tabela mais limpa, com os dados que realmente utilizaremos e de fácil entendimento.