Electric Wars:

Chargers Vs Vehicles

Agenda

1. Project Overview
2. Resources
    a. Methodology
    b. Machine Learning: Predict Range
    c. Tech Wars: PySpark Vs Pandas
3. App demo

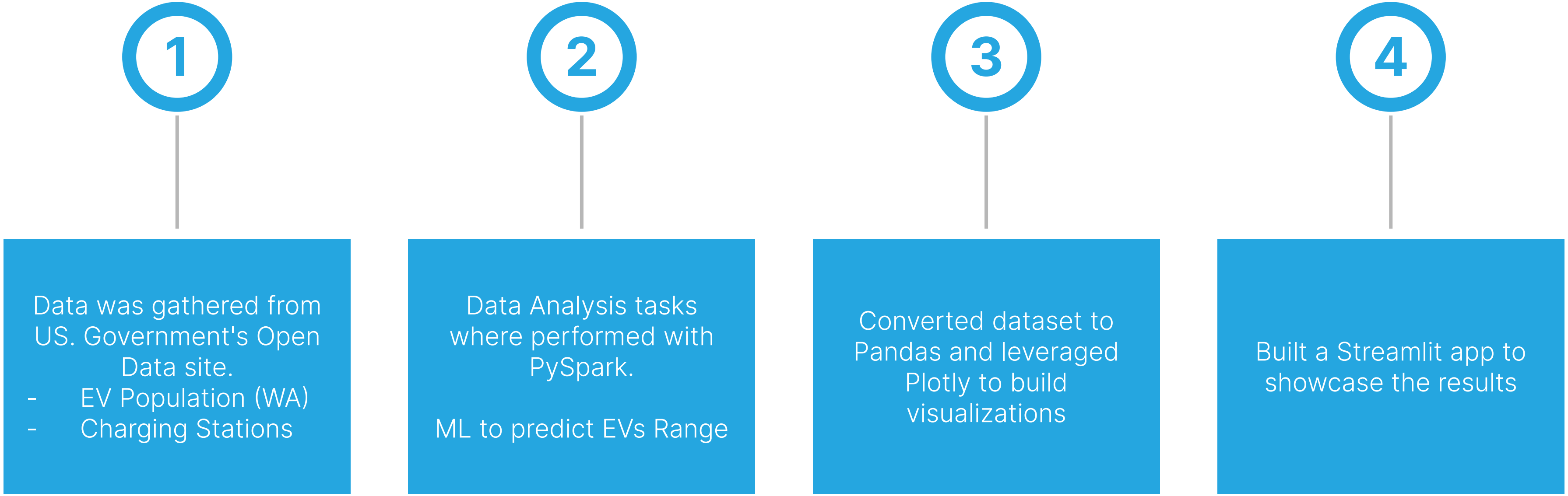
Project Overview

This project has two main goals:

- Analyse the EV and CS market in the US and in a given State.
- Draw conclusions from the analysis.

- Perform the analysis leveraging new tech: PySpark.
- Compare tech performance between Pandas & PySpark.

Methodology



1

Data was gathered from US. Government's Open Data site.

- EV Population (WA)
- Charging Stations

2

Data Analysis tasks where performed with PySpark.

ML to predict EVs Range

3

Converted dataset to Pandas and leveraged Plotly to build visualizations

4

Built a Streamlit app to showcase the results

DATA

PySpark

pandas plotly

Streamlit

ML: Predict Electric Range

Machine Learning Framework

Preprocessing

1. StringIndexer
2. OneHotEncoder
3. VectorAssembler

Model Definition

1. Random Split: Test & Train
2. Model: Linear Regression
3. Fit the model: Train set

Evaluation

1. Performance: 3.6 rmse

Electric Range	Prediction
291	289.49
322	319.58
19	18.77
151	147.74
204	197.42
215	215.11
84	84.12
125	120.40

Tech War: PySpark Vs Pandas

PySpark is the Python API that is used for Spark.

Spark is a big data computational engine, whereas Python is a programming language.

Pandas is a software library written for the Python programming language used for data manipulation and analysis.

Technology	Action	Details	CPU Time	Wall Time
PySpark	Install	!pip install pyspark	423 ms	40.8 s
Pandas	Import libraries	Pandas, Plotly	297 ms	1.59 s
PySpark		pyspark, SparkConf, SparkContext, SparkSession, SparkFiles	44.7 ms	54.9 ms
Pandas	Read data	Github url	62.5 ms	7 s
PySpark		Session + Github url	184 ms	23.9 s
Pandas	Display data	df.head()	0 ns	0 ns
PySpark		df.show()	9.58 ms	779 ms
Pandas	Information data	df.info()	46.9 ms	161 ms
PySpark		df.printSchema()	3.04 ms	12.9 ms
Pandas	Check null	df.isna().sum()	15.6 ms	153 ms
PySpark		df1.select([count(when(isnan(c) | col(c).isNull(), c)).alias(c) for c in df1.columns]	109 ms	7.87 s
Pandas	Filter data	df1 = df[df['State']=='WA']	0 ns	27.2 ms
PySpark		df1 = df.filter(df.State == 'WA')	2.22 ms	19.7 ms
Pandas	Count values	df.City.value_counts()	0 ns	9.52 ms
PySpark		df.groupBy('County').count().orderBy('count', ascending=False).show()	25.8 ms	2.59 s
Pandas	Plot	df1 = df.Model.value_counts().sort_values(ascending=False) / df1[:10].plot(kind='barh')	46.9 ms	1.38 s
PySpark		to.Pandas() + Plotly	73.7 ms	1.22 s

**CPU Time: how many seconds the CPU was busy.*
*** Wall Time: Works as a stopwatch, time elapsed between the start of the process and 'now'.*

Streamlit App

-DEMO-