УНИВЕРСИТЕТ
ИННОПОЛИС

Predicting Taxi Driver Payments: A Big Data Pipeline

Team Members

01 Anastasia Pichugina - Data Collection & Preparation, EDA

02 Arthur Gubaidullin - Feature Engineering, ML Modeling

03 Israel Adewuyi - Dashboard, Charts

04 Anna Gromova - Report writing, GitHub maintenance

Project Goal

01 Problem Statement

Taxi companies need to balance driver earnings with customer pricing

02 Objective

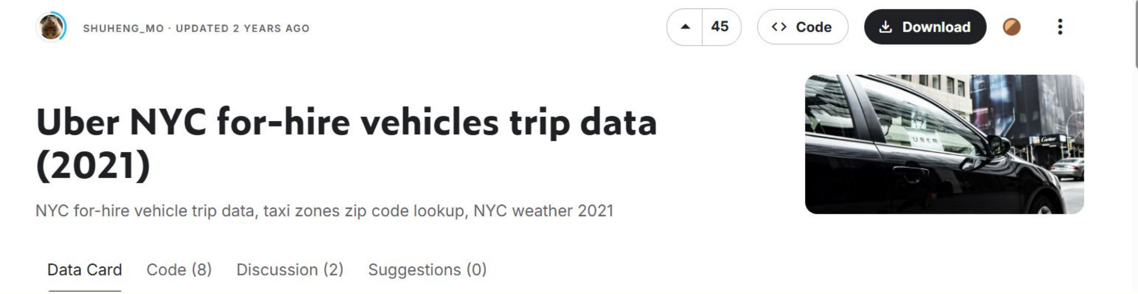
Build a model to predict *driver_pay* based on trip features to:

- Detect unfair payments
- Optimize dynamic pricing
- Improve driver satisfaction

03 Business Impact

Driver churn reduction potential

Datasets



01 Trips Data (Parquet files)

- 2.4M trips (January)
- Key fields: driver_pay, timestamps, locations

02 Taxi Zones (CSV)

- 265 zones
- Fields: borough, zone, service_zone

03 Weather (CSV)

- 356 days
- Key fields: temperature, humidity, snow, wind, uv, visibility, etc.

Data Collection & Ingestion

- Converted collected from Parquet to CSV
- Designed PostgreSQL schema with optimized tables (*trips, taxi_zones, weather*)
- Ingested data to HDFS using Sqoop with Avro and Snappy compression

taxi_zones		
int	location_id	PK
varchar	borough	
varchar	zone	
varchar	service_zone	

weather		
date	date_id	PK
varchar	station_name	
varchar	station_address	
varchar	resolved_address	
float	temperature	
float	feels_like	
float	dew_point	
float	humidity	
float	precipitation	
int	precipitation_prob	
varchar	precipitation_type	
float	snow	
float	snow_depth	
float	wind_gust	
float	wind_speed	
float	wind_direction	
float	sea_level_pressure	
float	cloud_cover	
float	visibility	
int	uv_index	
float	severe_risk	

trips		
bigserial	trip_id	PK
varchar	hvfhs_license_num	
varchar	dispatching_base_num	
varchar	originating_base_num	
timestep	request_datetime	
timestep	on_scene_datetime	
timestep	pickup_datetime	
timestep	dropoff_datetime	
int	pu_location_id	FK
int	do_location_id	FK
float	trip_miles	
int	trip_time	
float	base_passenger_fare	
float	tolls	
float	bcf	
float	sales_tax	
float	congestion_surcharge	
float	airport_fee	
float	tips	
float	driver_pay	
char(1)	shared_request_flag	
char(1)	shared_match_flag	
char(1)	wav_request_flag	
char(1)	wav_match_flag	
date	request_date	FK
time	request_time	

	station_name character varying (50)	station_address character varying (100)	resolved_address character varying (100)	date_id [PK] date	temperature double precision	feels_like double precision	dew_point double precisic
1	nyc	nyc	New York, NY, United States	2021-07-17	27.5	29.3	
2	nyc	nyc	New York, NY, United States	2021-09-16	23.3	23.3	
3	nyc	nyc	New York, NY, United States	2021-11-06	8.1	7.3	
4	nyc	nyc	New York, NY, United States	2021-10-11	19	19	
5	nyc	nyc	New York, NY, United States	2021-03-15	-0.2	-5.7	

	location_id [PK] integer	borough character varying (50)	zone character varying (100)	service_zone character varying (50)
1	81	Bronx	Eastchester	Boro Zone
2	220	Bronx	Spuyten Duyvil/Kingsbridge	Boro Zone
3	203	Queens	Rosedale	Boro Zone
4	159	Bronx	Melrose South	Boro Zone
5	178	Brooklyn	Ocean Parkway South	Boro Zone

Data Storage & Preparation

- Created Hive tables with:
  - Partitioning by *dispatching_base_num*
  - Bucketing by location IDs (8 buckets)
- Implemented dynamic partitioning for future data

Analysis Results

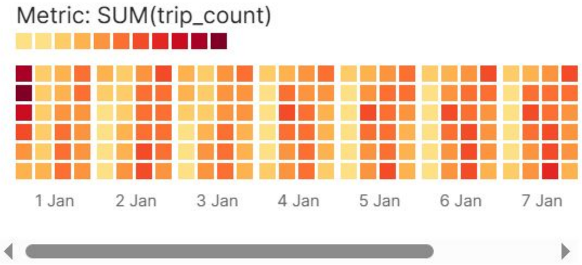
- HiveQL for aggregations (trip counts, avg. payments)
- Spark SQL for complex joins (trips + weather)

Average number of trips per day

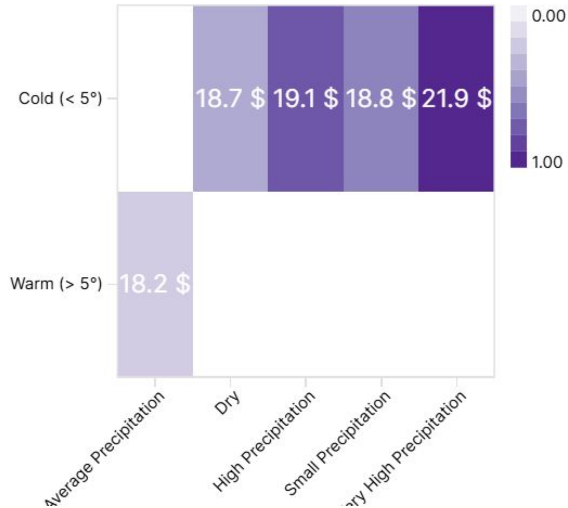
307k

Trips per day

Heatmap: trips count by hours

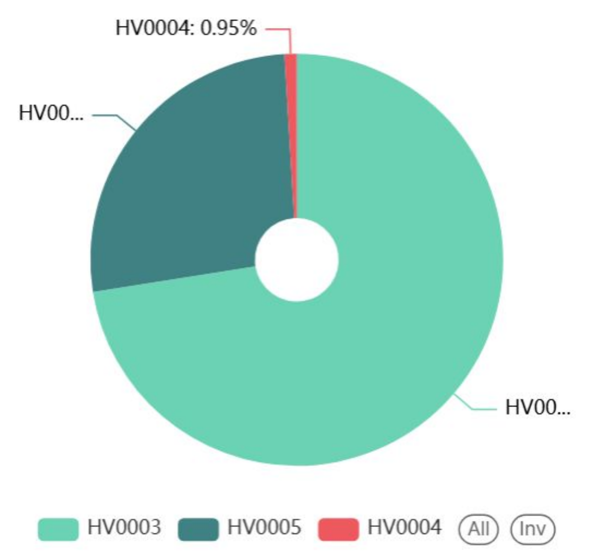


Heatmap with prices



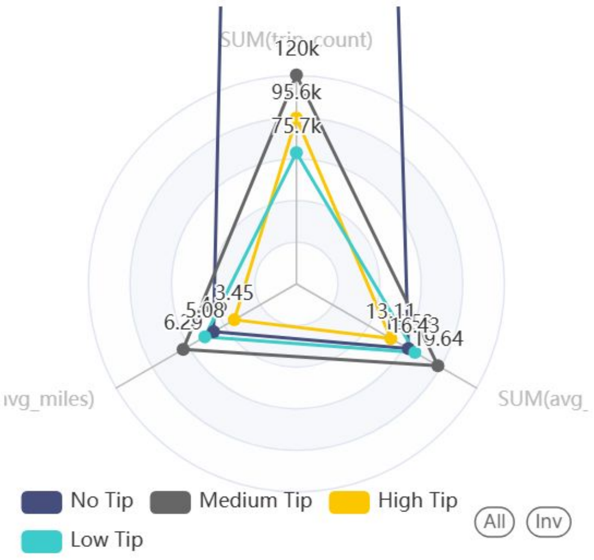
Analysis Results

Companies distribution

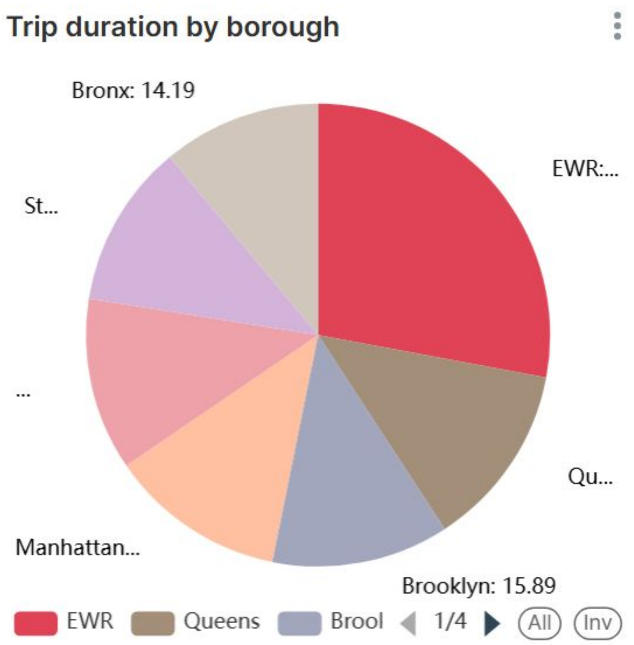


Most popular destinations

Tips



Analysis Results



Feature Engineering

- Engineered features:
  - Cyclical encoding for *day/month* (sine/cosine)
  - One-Hot Encoding for *borough/service_zone/license_num*
- Dropped high-cardinality and irrelevant columns
- Handled missing values
- Joined tables

ML Stage

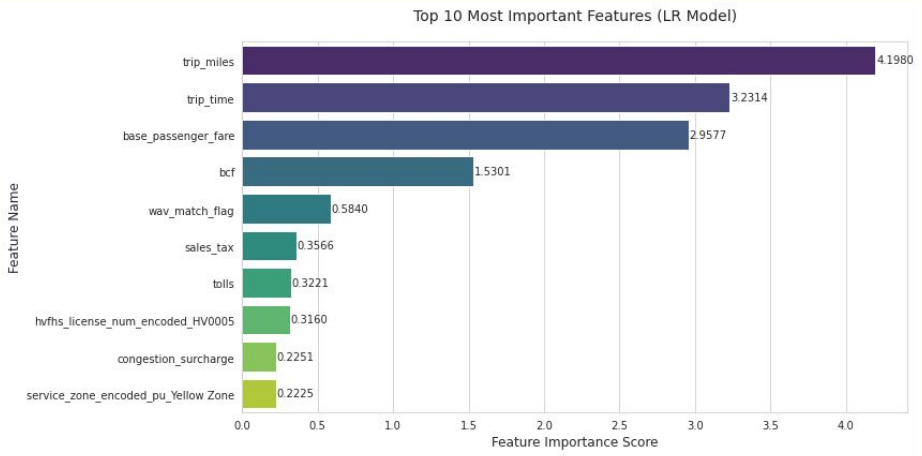
- Trained/tested (70/30 split):
    - Train/test sizes: (1714935, 734381)
    - Linear Regression
    - Gradient Boosted Trees
- Optimized via 3-fold CV with grid search

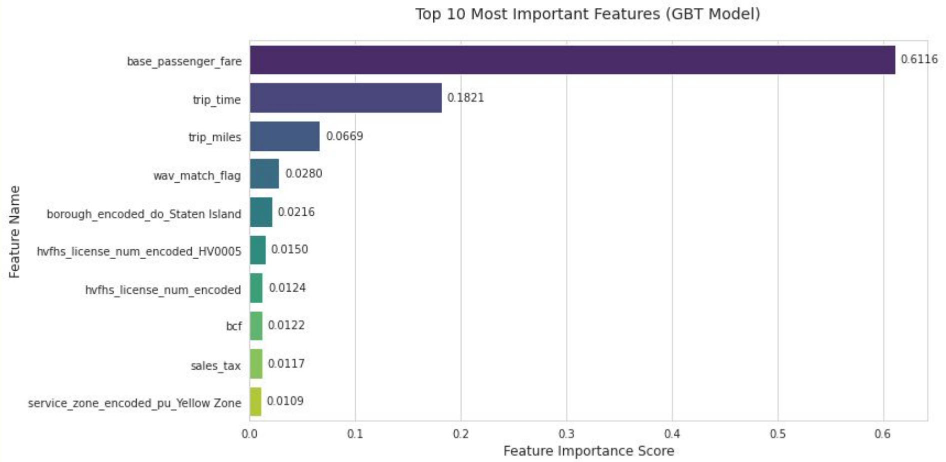
Key results:

- Best model: Linear Regression (R^2 0.934, RMSE 2.84)
- Outperformed GBT (R^2 0.911, RMSE 3.37)

Model	RMSE	MAE	R^2
Linear Regression	2.84	1.42	0.934
GBT	3.37	1.40	0.911

Feature Importance Analysis





Challenges & Solutions

- Memory constraints during data preprocessing & fine-tuning:
  - Solved by using Spark distributed system;
  - Solved by using Gradient-based optimization model.
- Weather data alignment (dates format):
  - Addressed via date features preprocessing and careful joining of datasets.

Demo