TEAM18 REPORT

Food delivery time prediction

- **Introduction**
  - **Business objectives**

    It is very important for food delivery services to designate a time in which they can deliver the order. Accurately predicting delivery times will help services retain customers and attract new ones. So, an automated model that determines the delivery time based on the order details would be a appropriate solution in the food delivery industry. It will save time and money and improve customer base that is valuable from the point of business.
- **Data Description**
  - **Data Characteristics**

    Food delivery dataset contains information about deliverer (person id, person age, person rating), restaurant location and delivery location (latitude and longitude), type of order (snack, drinks, ets.), type of vehicle and delivery time.

    The data consists of 45451 records and 11 fields. As a target attribute for the problem of was selected the `time_taken` column. The `id` field is unique and characterize each specific order. So, it does not affect the target values.
- **Architecture of data pipeline**
  - **The input and the output for Stage 1**

    The purpose of this stage was to bild relational database inside cluster. The input on this stage was original `.csv` file. The output: the relational database `food_delivery`, results from some test SQL queries and HDFS table serialized in AVRO.
  - **The input and the output for Stage 2**

    The main idea of this stage is data storage, preparation and EDA. The input of this is stage is presenting as relational database and AVRO files. The output is created Hive tables and the result of EDA(charts)
  - **The input and the output for Stage 3**

    This stage is presenting the predictive data analysis in PySpark for a regression task. The input data is the Hive table food_delivery from the database team18_projectdb. The output: the code saves the trained models and their predictions to HDFS in JSON and CSV formats. It also evaluates the performance of the models ans saves their results in CSV format.
  - **Stage 4**

    The main purpose of last stage was to present results of accomplished work. At the final step the web dashboard in Apache Superset was published. It presents the data characteristics and the results of EDA and PDA.
- **Data preparation**
  - **ER diagram**

food_delivery	
id	VARCHAR
delivery_person_id	VARCHAR
delivery_person_age	INTEGER
delivery_person_ratings	FLOAT
restaurant_latitude	FLOAT
restaurant_longitude	FLOAT
delivery_location_latitude	FLOAT
delivery_location_longitude	FLOAT
type_of_order	VARCHAR
type_of_vehicle	VARCHAR
time_taken	INTEGER

- **Some samples from the database**

  Some simple from the dataset (for more, you can see dashboard or the dataset in GitHub):

id	delivery_person_id	delivery_person_age	delivery_person_ratings	restaurant_latitude	restaurant_longitu
4607	INDORES13DEL02	37	4.9	22.745049	75.89247

- **Creating Hive tables and preparing the data for analysis**

  To handle big data and improve query performance we created Hive tables. The first one was partitioned by `type_of_order` as it is a categorical feature. And then we divided it into 10 buckets to optimize searching queries.

- **Data analysis**
  - **Charts (Analysis and Interpretations)**

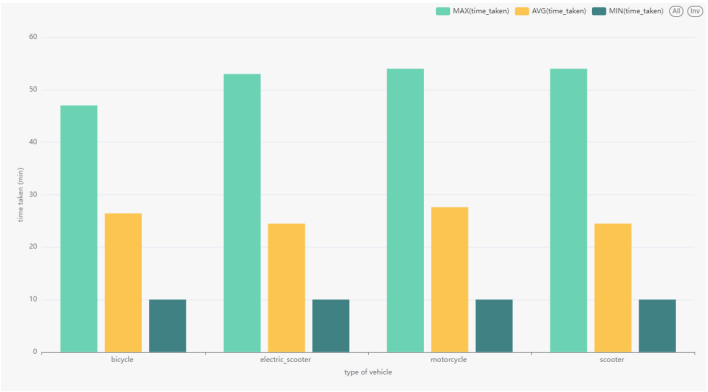

Chart 1

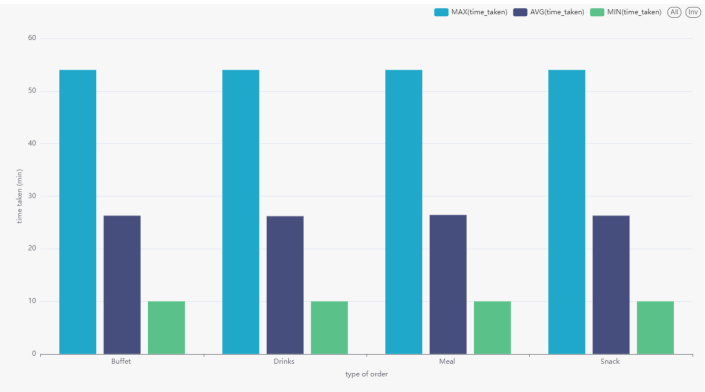

Chart 2

Charts 1 and 2 are showing that `type_of_vehicle` and `type_of_order` fields relative to `time_taken` have approximately the same max, min and mean. Hence, the model could not learn the dependencies between this fields and the target.

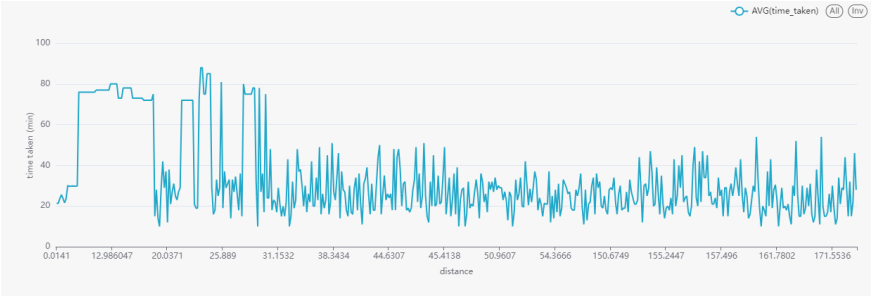

Chart 3

Before the data analizing we make a hypothesis that the `time_taken` can be dependent on the geographical distance. The 3rd chart presents the relationships between `time_taken` and the calculated distance between rastaurant and delivery locations. No obvious linear relationship was found that is why the ML model should be learned not on the distance features, but on the initial coordinates transformed into XYZ system.

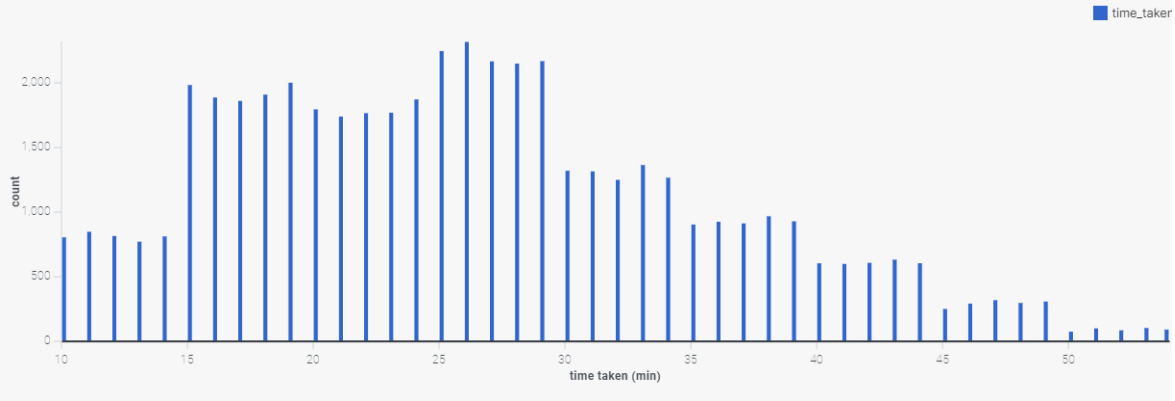

Chart 4

The Chart 4 shows distribution of the numbers of values in `time_taken` field. There is distribution with not significantly heavy left tail that approximately normal. Hence, the target field can be called balanced.

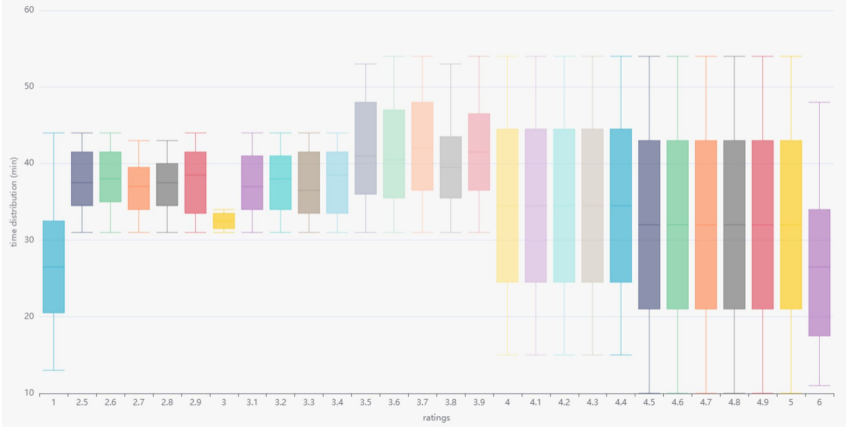

Chart 5

The chart 5 is presenting the relationship between `time_taken` and `delivery_person_rating`. It shows that orders delivered by couriers with higher ratings have, on average, shorter delivery times compared to those couriers with lower ratings.

- **ML modeling**
  - **Feature extraction and Data preprocessing**

    After analyzing the dataset, we proceeded to the feature extraction phase. As mentioned above, the `id` field is unique and does not affect the target field. The `type_of_vehicle` and `type_of_order` fields also have no effect on the target field: they have approximately the same distribution and we can conclude that they do not contribute significantly to the result. Thus, we decided to drop these fields and saves the results (models checkpoints, evaluation and predictions) as files `<name>_others`, while the results of training on the dataset without only `id` as `<name>`.

    For preprocessing geodetic coordinates (latitude, longitude) into ECEF coordinates (x, y, z), we customized the Transformer using the algorithm presented in that article. Since there is no altitude in our dataset we assume that all points are at the same constant height and set it as 0. That approach does not have a negative impact on the scalability.

    Before training, we use a pipeline for preprocessing and feature extraction and split the resulted dataset into train and test datasets in the ratio of 60% to 40%.

  - **Training and Fine-tuning**

    We select Linear Regression Model (LR) and Random Forest Regressor (RF) as models for our problem. We trained both base models and moved on to the stage of tuning parameters. As metrics, we took the two most common for regression problems: the RMSE and R2 score.

    For tuning, we chose the regularization `regParam` and the elastic network `elasticNetParam` as hyperparameters of `ParamGridBuilder` for the LR model and, for the RF model, the depth `maxDepth` and number of trees `numTrees` hyperparameters. We perform 5-fold cross validation with RMSE and R2 metrics, select the best models and test them.

  - **Evaluation**

    Performance of both best LR and RF models are presented in the dashboard. We compare performances of both models and can say that RF model achieve higher results than LR model.

    Results in `evaluation_others.csv`:

	model	RMSE	R2
1	LinearRegressionMod…	8.340792649581509	0.20937751588262754
2	RandomForestRegre…	7.595226195213991	0.3444045427628445

  - **Steps and Improvements**

    As steps of ML model we can highlight training the model on two different datasets: the dataset without meaningless columns (like `ID` and `Delivery_person_ID`) and the dataset with only matter columns (excluding `Type_of_order` and `Type_of_vehicle`). This approach shows how the model's performance can improve with more relevant features.

    The improvements in model performance are observed through the reduction in RMSE and the increase in R2 score with each step of feature selection.

- **Data presentation**
  - **In the dashboard we have three charts: Data description, Insights and ML**
  - **Data description**

    In this chart we have represented dataset overview, types of columns and a sample of the dataset limited to first 10 lines.

- **Insights**

  In this part we show up the graphics from the stage II with some explanations.

- **ML**

  In modeling part we added the results of our models that expressed as evaluation, prediction, hyper-parameter optimization and characteristic of feature extraction of our models

- **Conclusion**
  - **Summary of the report**

    In conclusion, the project aimed to develop an automated model for predicting food shipping times, which is essential for boosting consumer delight and retention in the meals delivery enterprise. By leveraging a comprehensive dataset, the team constructed a data pipeline, performed exploratory data analysis, and implemented machine learning techniques to predict delivery times accurately. The process concerned disposing of irrelevant functions, preprocessing geospatial statistics, and fine-tuning models to optimize performance. The effects tested that the Random Forest Regressor model outperformed the Linear Regression version, achieving higher accuracy in predicting delivery times. The project displays the effectiveness of the proposed methodology and opens avenues for further upgrades and applications in similar domains

- **Reflections on own work**
  - **Challenges and difficulties:**

    1. Teams with large datasets (team22 and some others) grabbed kernel resources.
    2. One of the challenges was the implementation of custom transformer for converting geospatial features (latitude and longitude) into ECEF coordinates (x, y, z).

  - **Recommendations**

    It is recommended to have suitable equipment to run the scripts.

  - **The table of contributions of each team member**

Project Tasks	Irina Shchetinina	Arina Yartzeva	Ksenia Shchekina	Polina Bazhenova	Deliverables	Hours spent
Searching for the dataset and describing	25	25	40	10		2
Stage I (data collection and ingestion)	0	60	10	30		6
Stage II (data storage and preparation)	0	50	10	40		12
Stage II (EDA)	0	50	0	50		2
Stage III (preparation, feature extraction)	100	0	0	0		0.5
Stage III (custom transformer for geospatial features)	5	0	90	5		1
Stage III (model 1 - Linear Regression model)	90	5	0	5		10
Stage III (model 2 - Random Forest model)	5	5	60	30		10

Stage III (compare models)	25	10	25	40		0.5
Stage IV (create a dashboard)	30	30	10	30		4
Stage IV (report)	25	10	35	30		3
Github managing	0	100	0	0		1