Bike Sharing Demand

Niranjan Reddy Masapeta
dept of Computer & Software Engineering, San Jose State University
rajashekarreddy.kommula@sjsu.edu

Purna Sai Mahesh Goud Palagani
dept of Computer & Software Engineering, San Jose State University
purnasaimaheshgoud.palagani@sjsu.edu

Sujan Rao Chikkela
dept of Computer & Software Engineering, San Jose State University
sujanrao.chikkela@sjsu.edu

Utsav Savaliya
dept of Computer & Software Engineering, San Jose State University
utsav.savaliya@sjsu.edu

***Abstract*— Bicycle-sharing programs have grown in quantity and popularity in cities all around the world during the last decade. Users can borrow bicycles for a fee on a very short-term basis through bicycle-sharing programs. The bicycle-sharing system has grown in popularity in numerous cities across the world in recent years. This allows users to borrow a bike from point A and return it to point B in Washington D.C, USA, however, they may just go on a ride and return it to the same area. Regardless, each bike may serve several people each day. It's a system that allows you to hire bicycles at an affordable price.**

A user of the system may easily reach a dock inside the system to unlock or return bicycles, thanks to advances in information technology. These technologies also generate a plethora of data that may be utilized to investigate how people use these bike-sharing systems.

Keywords—Bike sharing, Model, time-series, demand, customers.

1. INTRODUCTION

One of the most popular analysis methods of the past has been time-series analysis. The goal of time series forecasting in statistics is to predict changes that occur over time. The historical data we observe over time is used to anticipate the future patterns and changes. In many different domains, time-series analysis and forecasting is used, such as retail market analysis, sales forecasts, stock market analysis, and many more. Our problem is to forecast the number of customers who will rent bikes based on the conditions around them. When customers rent bicycles for the day, we have a time series of data pertaining to this.

The dataset that we have contains rows for each day of data collected with variables such as temperature, weather conditions, wind speed, and the number of customers who rented the bicycles. As part of our analysis, we need to examine the data from the rentals of bikes, including the temperature, the time of year, the weather conditions, and all other factors relevant to the rentals. Since the dataset is a time-series data, we are trying to predict the number of customers based on past data.

2. RESEARCH METHODOLOGY

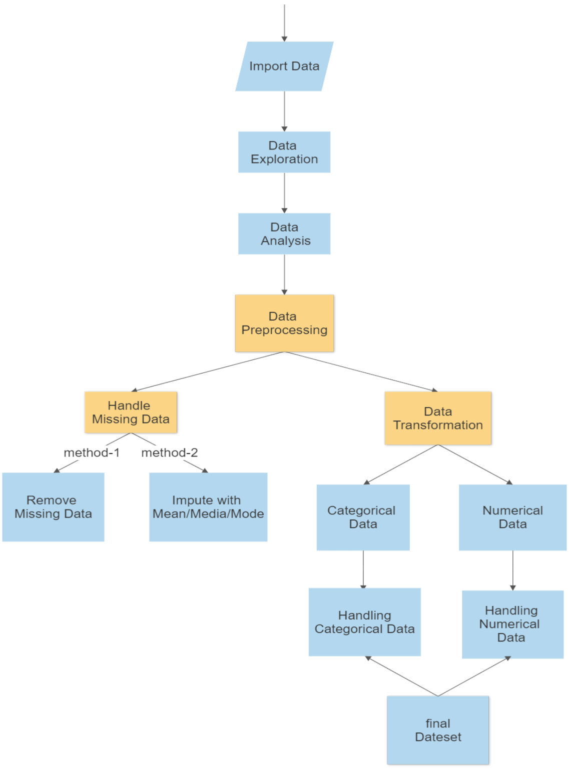


Fig. 1. Methodology

2.1 Dataset

We got a dataset from openML. The link to the dataset is https://www.openml.org/search?type=data&status=active&id=43486&sort=runs , The dataset contains 2922 entries and 23 features. The entries are from Jan 1st 2011 to Dec 31st 2018.

2.2 Pre-Processing

2.2.1 Data cleaning:

We have cleaned the data by removing null values from the data. As this is a time-series data, we cannot directly remove the entries even if the number of missing values is less as time series is a continuous data. So, we have filled the missing values using meaningful methods such as forwardfill and values wherever it is appropriate. In the case of weather features, we replaced missing values with 0s as the data columns contain either 0 or 1 as its value by meaning. In the case of customers, we used the forward filling method as the data is time-series, so we assumed data would be closer to the entry of before date. In the case of temp_avg, we calculated the average of max temperature and min temperature for now. (We are planning to fit a linear estimate with available data using temp_max, temp_min as features and temp_avg as target variable. Once we fit the model, we will fill in the missing values using the trained model to get more accurate data**)**

2.2.2 Create new features

We have created new features by merging the weather attributes to simplify the data analysis and modelling. All the similar weather conditions have been categorized into three groups: ice, rain and fog. We also created year, month and day of the week from the given datetime variable to gain insights from the data on customer behavior.

2.2.3 Data Analysis and Visualization

We have plotted the following plots to infer a few observations:

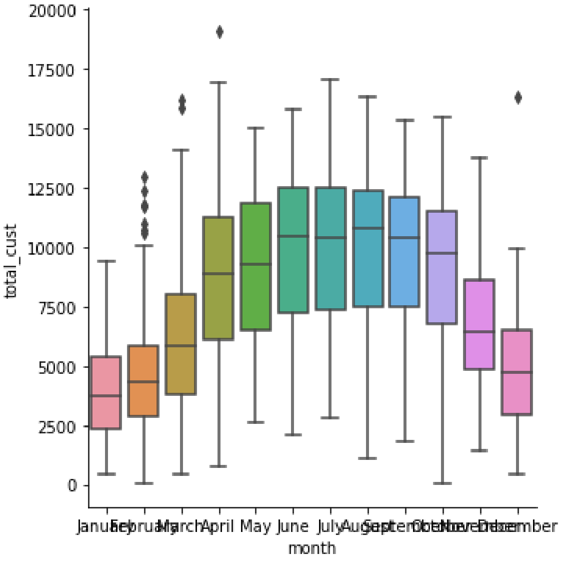


Fig. 2. Total customers during each month

We can infer from the above plot that the number of customers were higher during the months of April to October typically during Summer and Fall where temperatures are at high compared to Spring and Winter where temperatures are too low so we can see that there are less number of customers overall during those seasons. We can also observe that there are few outliers on a particular day in february for example, more number of people registered. We can assume there might be few special events that have caused more users to register on that particular day.

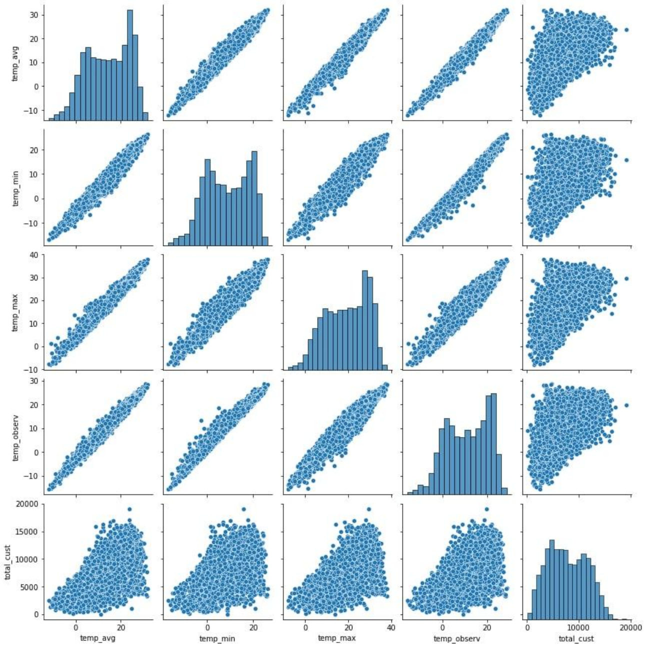


Fig. 3. Scatter plots of temperatures vs total customers

In the above plot, we are plotting scatter plots between the temperature values and total customers to observe the relation between them. We can see that there is a perfect linear relation between all temperature values. Also, the temperature features have a medium level linear relationship with the total number of customers registered.

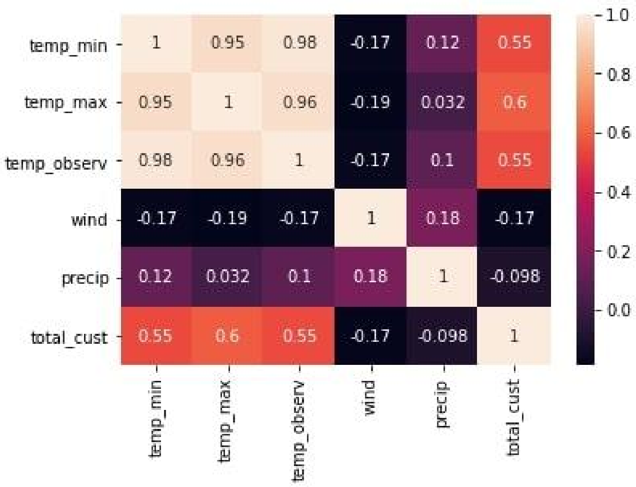
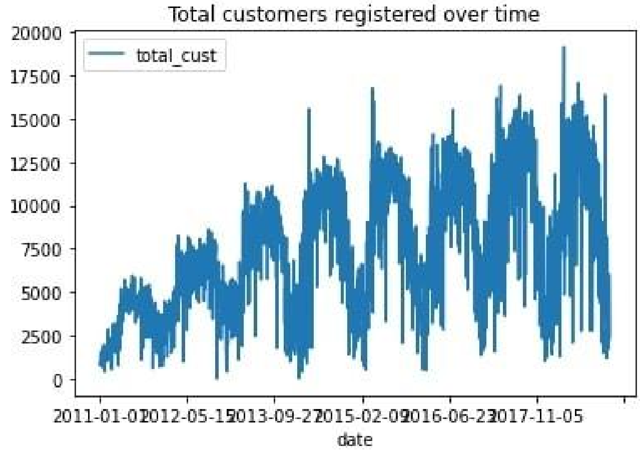


Fig. 4. Heatmap plot for correlation

The heatmap above provides a more accurate relation among the variables. There is no significant correlation between wind, precip and the target variable but we can see that the temperature features have good amount with the target variable

2.2.4 Time series specific analysis

We are dealing with time series data. Apart from looking at the correlation plots and distribution plots, we have to see if the data is stationary or non stationary. We will first plot line plots over time to check if the data is stationary or not.



We can see from the above plot that our time series problem is non-stationary. We must deal with this before modeling part.

What is Stationary?

When working with time-series data, 'stationarity' is one of the most important concepts that you'll encounter. Generally, stationary series are those whose properties don't change over time, such as the mean, variance, and covariance. The upward trend in the mean is clearly visible, since it varies (increases) with time. In other words, this is not a stationary series. A series is termed stationary if it does not exhibit a trend.

2.2.5. Further Analysis: *We plan to do further analysis to find out if our series is stationary or non-stationary using some popular statistical tests. One of the tests is the ADF(Augmented Dickey Fuller) test. It can be used to determine whether a series is stationary and determine whether a unit root exists. As a result, it can also identify whether a series has diverged. If we are unable to reject the null hypothesis, we can say the series is non-stationary, and the other test we can use for the analysis is (KPSS) Kwiatkowski–Phillips–Schmidt–Shin test, we can get an idea if a time series is stationary around a mean or linear trend, or if it is non-stationary because of the unit root. A stationary time series has statistical characteristics that remain constant over time, such as the mean and variance. ADF test has hypothesis testing for difference stationary whereas KPSS test looks for trend stationarity in the series. If our series has non-stationarity, we will convert our series to stationary using following methods:*

1. *Differencing: Removes series dependencies on time.*
2. *Detrending: Removes trend effects from dataset*
3. *Transformation: Converts into stationary using log transfer method*

2.2.6 Data Modelling

After converting our time-series data to stationary, we will move to modelling the data. Our tentative plan for now is to perform a naive-univariate prediction only using the target variable. We will next use Random forest and then use ensemble methods at advanced level to analyse and compare the performance of different approaches.

3.Models

Ensemble means combining multiple models to make predictions instead of one model.Ensemble learning offers an out of the box approach to combine the predictive power of multiple learning models. The models that participate in forming the ensemble are called base learners. Ensemble uses two types of methods:

3.1. Bagging

It means that to create a different set of training subsets with replacement and output is decided based on the majority voting. For instance Random Forest

3.2. Boosting

It uses sequential models with the combination of weak learners and strong learners that result in high accuracy.For instance XG BOOST, ADA BOOST, Gradient BOOST. Here, the focus is to build trees sequentially in such a way that each subsequent tree is aimed at reducing the errors of the previous tree model.

3.3. Bagging meta-estimator

It is an ensembling method which includes steps like creating random subsets from dataset which includes all features, than a base estimator is set on each of these sets and at the end all the predicted results from each of the subset model is combined to get result.The hyper parameters used are

base_estimator: It indicates the base estimator to fit a random subsets of a dataset

n_estimators: Number of base estimator which is required

n_jobs: number of jobs to be run in parallel

random_state:It is used to specify the method of random split. This parameter is used when comparison between 2 models is to be needed.

3.4. Adaptive Boost

AdaBoost is a decision tree with one level which is a decision tree with just one split. It builds a model and gives equal weights to all the data points and then assigns higher weights to points which show highest errors. Now all the points which have higher weights are given more importance in the next model. It will keep training models until and unless a low error is received for the regression problem. When there is some non-linearity in our dataset this algorithm helps as it captures these nonlinearities which in the end contributes to better accuracy on the regression problem.

3.5. Gradient Boosting

As stated above sequential models are built here and they try to reduce the errors of previous models. The speciality is that errors are reduced by building a new model based on the errors of the previous model. It minimizes loss function by adding weak learners with help of gradient descent.

The hyper parameters that we have chosen are n_estimators,Learningrate,max_depth. N_estimators is the number of trees (weak learners) that should be there in the model. For the learning rate the low value always works better provided that there are sufficient numbers of trees present. Max_depth is related to the height of the decision. There are various extensions of Gradient boosting which can be explored.

3.6. XGBoost

XGboost stands for eXtreme Gradient Boosting.It is one of the methods under boosting. It outperforms other models due to following features: XGboost penalizes models through L2 and L1 regularization which is used to prevent overfitting. Our data contains one hot encoded value which shows that data is sparse. So there is a need for a sparsity aware split finding algorithm for handling different types of sparsity patterns in data. In terms of computation also XGboost outperforms as it uses multiple cores on CPU. It is used when there are a large number of training samples. Also when there is a mix of numerical and categorical data XGboost is preferred.

3.7. CatBoost

To handle categorical variables is a tiresome operation, especially when there is a large number of such variables. When your categorical variables have many labels, performing one-hot-encoding such data exponentially increases the dimension of the model and it becomes really tedious of preprocessing and training the dataset while avoiding overfitting.

The benefit that CatBoost provides is that it can automatically deal with categorical variables and does not require extensive data preprocessing before giving the data to the machine learning model.

4. Comparisons

The main step in any machine learning model is the evaluation of model accuracy. The Mean absolute error, Mean Squared Error, R-Squared or Coefficient of determination and Root Mean Squared Error metrics are used to evaluate the performance/accuracy of the model in regression analysis problems.

MAE: The Mean absolute error is the average of the absolute difference between the predicted values and the actual values.

MSE: It represents the average of the squared difference between the actual and predicted values. It measures the variance of the errors.

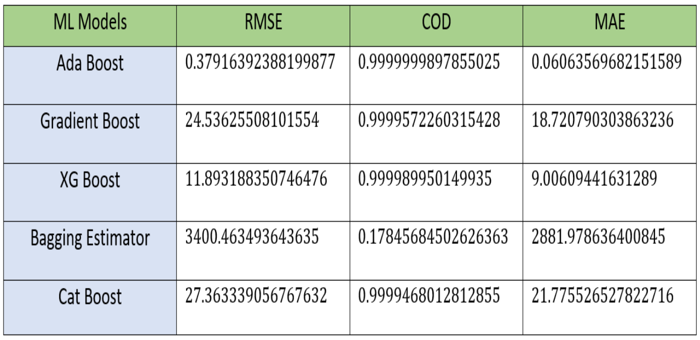
RMSE: The square root of Mean Squared error is Root Mean Squared Error.

R-squared or coefficient of determination: It is the proportion of the variance in the dependent variable. It is a scale-free score i.e. the value of R square will be less than one only even if the values are small or large.

How to choose between them for the determination of the accuracy of the model?

Mean Squared Error(MSE) and Root Mean Square Error penalizes the large prediction errors with regard to Mean Absolute Error (MAE). But RMSE is generally preferred than MSE for the evaluation of the performance of the regression problems compared to other models. The lower value of MAE, MSE, and RMSE indicates higher accuracy of a regression model. MSE is easy in terms of computation time unlike MAE. It is because the former is differentiable while the later is non-differentiable. Thus, mostly RMSE is used as a metric for calculating accuracy in terms of some loss function. For the comparison of accuracy of different models, RMSE is preferred over R-Squared. To conclude both RMSE and R-Squared tells how well a regression model is best suited for a particular dataset . To be specific RMSE tells how a regression model is used to predict the value of a response variable while on the other hand R-Squared tells how well the predicted variable is able to tell about the variation in the response variable type.

Also random_state hyperparameter of model discussed is used for comparison between different models



ML Models	RMSE	COD	MAE
Ada Boost	0.37916392388199877	0.9999999897855025	0.06063569682151589
Gradient Boost	24.53625508101554	0.9999572260315428	18.720790303863236
XG Boost	11.893188350746476	0.999989950149935	9.00609441631289
Bagging Estimator	3400.463493643635	0.17845684502626363	2881.978636400845
Cat Boost	27.363339056767632	0.9999468012812855	21.775526527822716

We can say that the XG boost model outperformed well though Adaboost model gave the lowest errors, XG boost is better so as to avoid overfitting of the model.

5. Conclusion

To conclude,in this project, we tried to explore various regression techniques on a bike sharing dataset. We compared the performance of these regressors by using many metrics used for determination of the accuracy. AdaBoost regressor significantly outperformed other regressors in accuracy in terms of all the metrics used for comparison but we will choose XG boost so as to avoid overfitting.

We can improve the predictions accuracy by including more features in the dataset in future by making the model robust. Model strength can be increased by implementing advanced Machine Learning models.

6. References

1. https://scikit-learn.org/stable/modules/ensemble.html
2. https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-to-time-series-analysis/
3. https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e