ABSTRACT

PROJECT TITLE:- PREDICTING THE SALE PRICES OF BULLDOZERS

DOMAIN:- MACHINE LEARNING (REGRESSION)

Machine learning is a growing technology which enables computers to learn automatically from past data. Machine learning uses various algorithms for building mathematical models and making predictions using historical data or information. A very powerful utility of Machine Learning is its ability to automate various decision making tasks. This frees up a lot of time for developers to use their time to more productive use. Currently, it is being used for various tasks such as image recognition, speech recognition, email filtering, Facebook auto-tagging recommender system, and many more.

With the help of machine learning concepts we are going to build a machine learning model which will predict the future sale price of bulldozers given its different characteristics according to the past data of the bulldozer sales. The dataset which we are going to use is available in “www.kaggle.com” . The given dataset is a time series problem and the machine learning concept which we are going to use is regression analysis. The primary technology for this project is Python. The tools which we are going to use for the data analysis are Matplotlib, NumPy and Pandas. And the tool which we are going to use to build the ML model is scikit-learn. The other tools which we are going to use are jupyter notebook and miniconda. The project will have a 6 step process of problem definition, data, evaluation, features, modeling and experimentation.

INTRODUCTION:

With the help of machine learning concepts we are going to build a machine learning model which will predict the future sale price of

bulldozers given its different characteristics according to the past data of the bulldozer sales. The dataset which we are going to use is available in Kaggle .

The given dataset is a time series problem and the machine learning concept which we are going to use is regression analysis and the algorithm which we are going to use is a random forest regressor.

PROBLEM STATEMENT :

To predict the future sale price of a bulldozer,given its characteristics and previous example of how similar bulldozers have been sold for using machine learning algorithms.

PROPOSED SYSTEM :

The proposed system will be able to predict the sale prices of BullDozers using machine learning algorithms.

Predicting the price with utmost accuracy is our top project objective.

SCOPE OF THE PROJECT :

This project will help us to analyze the sale price of bulldozer and see how accurately the model is predicting the price. If the accuracy is high we can use it on vehicle data and predict accurate prices for the vehicle. This will give fair prices of the vehicles to the future customers

TECHNOLOGIES:

- PYTHON
- ANACONDA
- JUPYTER

TOOLS

- NUMPY

• SCIKIT-LEARN
• PANDAS
• MATPLOT-LIB

HARDWARE REQUIREMENTS

• Processor : Intel® CoreTM i3-8130 @2.20GHz 2.21 GHz
• Installed RAM : 4.00 GB(3.84 GB usable)
• Device ID : 9AD7C616-60D3-421D-9ADA-64C7A9777B2B
• System Type : 64-bit operating system,x64-based processor
• Edition : Windows 10 Home Single Language
• Version : 21H
• OS build : 19043.1348

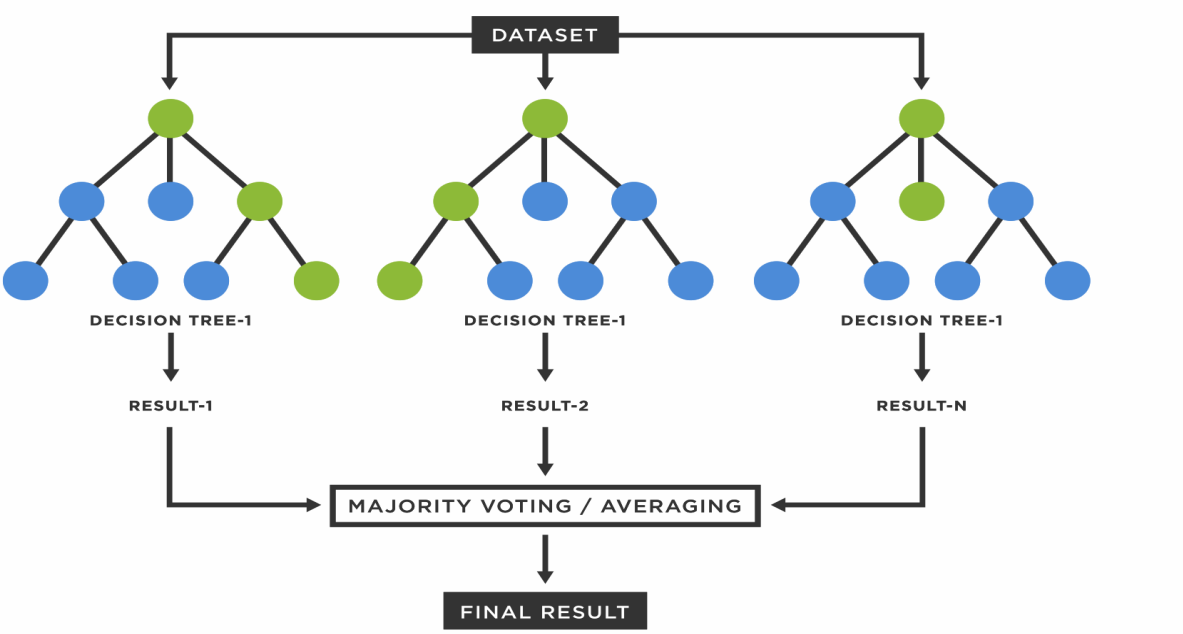
REQUIREMENTs

• Operating system running macos catalan or above windows 7 or above
• Python 3.0 version
• Python 3.0 had an emphasis on removing duplicative constructs and modules, in
• Keeping with “There should be one and preferably only one obvious way to do it”. • Python 3.0 remained a multi paradigm language. The main advantage in software approach is thauser’s network does not change. No extra devices are needed to be
• Installed, and management of the networkremains the same.

ALGORITHM:

A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging.A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to

improve the predictive accuracy and control over-fitting. We can see how random forest works :



Benefits of Random Forest:

Easy to Measure Relative Importance:

It is simple to measure the importance of a feature by looking at the nodes that use that feature to reduce impurity across all the trees in that forest. It is easy to see the difference before and after permuting the variable, and this gives a measure of that variable's importance.

Versatile:

Because a random forest can be used for both classification and regression tasks, it is very versatile. It can easily handle binary and numerical features as well as categorical ones, with no need for transformation or rescaling. Unlike almost every other model, it is incredibly efficient with all types of data.

No Overfitting:

As long as there are enough trees in the forest, there is little to no risk of overfitting. Decision trees can also end up overfitting. Random forests prevent that by building different sized trees from subsets and combining the results.

Highly Accurate:

Using a number of trees with significant differences between the subgroups makes random forests a highly accurate prediction tool

CODE EXPLANATION

Data:

The data and evaluation metric used is (root mean square log error or RMSLE) as this is a constraint from the Kaggle Bluebook for Bulldozers competition.

Looking at the dataset, we can understand that there's a time attribute to dataset. So it is *Time series problem.*

There are 3 datasets:

1. Train.csv - Historical bulldozer sales examples up to 2011 (close to 400,000 examples with 50+ different attributes, including SalePrice which is the target variable).

2. Valid.csv - Historical bulldozer sales examples from January 1 2012 to April 30 2012 (close to 12,000 examples with the same attributes as Train.csv).

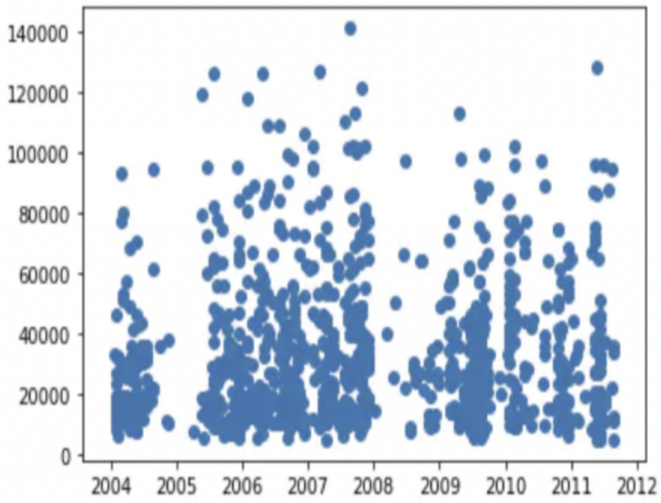
3. Test.csv - Historical bulldozer sales examples from May 1 2012 to November 2012 (close to 12,000 examples but missing the SalePrice attribute, as this is what we'll be trying to predict).

Data Exploration:

(1) Parsing dates : When working with time series data, it's a good idea to make sure any date data is in the format of a datetime object (a Python data type which encodes specific information about dates).
(2) Let's compare SalePrice and Saledate using Scatter plot

```
In [8]: fig, ax = plt.subplots()
        ax.scatter(df["saledate"][:1000], df["SalePrice"][:1000])

Out[8]: <matplotlib.collections.PathCollection at 0x7fdf08527f90>
```



We can see that there are very less sales in 2005 and more sales between 2007 and 2008

Convert strings to categories :

- One way to help turn all of our data into numbers is to convert the columns with the string datatype into a category datatype.
- To do this we can use the pandas types API which allows us to interact and manipulate the types of data.

● Once data is converted to categories turn them into codes and Filled missing values with Median of that column. Once our data is in numeric format and there are no missing values, we should be able to build a machine learning model.

Splitting data into train/valid sets :

As this is a time series problem, we will split our data into training, validation and test sets using the dates.

● Training = all samples up until 2011

● Valid = all samples form January 1, 2012 - April 30, 2012

● Test = all samples from May 1, 2012 - November 2012

Modeling:

We have used RandomForestRegressor model. Fit the train data to the model.
The evaluation function used here is root mean squared log error (RMSLE).

Make predictions on test data :

● Our model has been trained on data formatted in the same way as the training data.
● This means in order to make predictions on the test data, we need to take the same steps we used to preprocess the training data.
● Also check if test and training data has same columns so that there will not be any further hassle.
● Once preprocessing is done on test data, fit the test data to the model with best parameters.Now we've built a model which is able to make predictions.

RESULT

```
In [75]: # Make predictions on the test dataset using the best model
         test_preds = ideal_model.predict(df_test)
```

```
In [76]: # Create DataFrame compatible with Kaggle submission requirements
         df_preds = pd.DataFrame()
         df_preds["SalesID"] = df_test["SalesID"]
         df_preds["SalePrice"] = test_preds
         df_preds
```

Out[76]:

	SalesID	SalePrice
0	1227829	20300.076629
1	1227844	19005.918869
2	1227847	50037.759022
3	1227848	59533.179053
4	1227863	43240.593575
...	...	...
12452	6643171	45010.580330
12453	6643173	15243.111547
12454	6643184	15370.414444
12455	6643186	19424.227913
12456	6643196	28185.808608

12457 rows × 2 columns

CONCLUSION:

So, we got the sale price of the bulldozers using the random forest regressor algorithm and our model is working. The accuracy of model is also high so we can conclude that we have used accurate parameters and model to predict the outcome which is sale price.

Future enhancements:

The future enhancements for the project can be

1. using new models on the dataset like decision tree,support vector regression,lasso

regression and knn model

2.using the same model on different vehicles to predict the sale price using different evaluation metrics on the model