Contents

[**1)** **Description of datasets** 4](#_Toc116068104)

[a) Employee.csv 4](#_Toc116068105)

[b) Fields.csv 4](#_Toc116068106)

[c) WorkCode.csv 5](#_Toc116068107)

[d) WorkDetails.csv 5](#_Toc116068108)

[**2)** **Reading Data in python environment** 6](#_Toc116068109)

[**3)** **Data Cleaning** 6](#_Toc116068110)

[a) Removing Null/Nan values 6](#_Toc116068111)

[b) Removing Duplicated rows 7](#_Toc116068112)

[c) Removing outliers 7](#_Toc116068113)

[**4)** **Merging of Datasets** 8](#_Toc116068114)

[**5)** **Insights/Patterns identified from merged dataset** 8](#_Toc116068115)

[**6)** **Feature Selection** 10](#_Toc116068116)

[**7)** **Train-Test Split** 11](#_Toc116068117)

[a) What is Train-Test Split? 11](#_Toc116068118)

[b) When to Use the Train-Test Split 12](#_Toc116068119)

[c) How to Configure the Train-Test Split 12](#_Toc116068120)

[**8)** **Feature Scaling** 13](#_Toc116068121)

[**9)** **Machine Learning approaches** 13](#_Toc116068122)

[a) LightGBM Regression 14](#_Toc116068123)

[b) Multiple Linear Regression 14](#_Toc116068124)

[c) Elastic Net Regression 14](#_Toc116068125)

[d) Bayesian Ridge Regression 15](#_Toc116068126)

[**10)** **Accuracy of all applied ML techniques** 15](#_Toc116068127)

[**11)** **Alternate ways to increase performance of the models** 16](#_Toc116068128)

[a) Different feature scaling technique 16](#_Toc116068129)

[b) Hyperparameters tuning 16](#_Toc116068130)

[c) Ensemble methods 16](#_Toc116068131)

[d) Cross Validation 17](#_Toc116068132)

[e) Feature transformation 17](#_Toc116068133)

[f) Feature Creation 17](#_Toc116068134)

[**12)** **Comparative Analysis** 17](#_Toc116068135)

[a) Accuracy 17](#_Toc116068136)

[b) Mean Squared Error 18](#_Toc116068137)

[c) Mean Absolute Error 18](#_Toc116068138)

[d) Limitations Of each model 19](#_Toc116068139)

[13) Libraries Used 19](#_Toc116068140)

[14) Knowledge gained 20](#_Toc116068141)

[References 21](#_Toc116068142)

# **Description of datasets**

## Employee.csv

The employee.csv file contains 2,015 observations in 6 variables of worker related data. Out of 6 variables, 5 were string variables(including a datetime variable i.e., “DateofJoin”) and 1 was numeric variable. Variables description is given as follows.

1. Estate (Estate of the employee)
2. Division (Division from which employee belongs)
3. EmpCode (EmpCode of employee, to uniquely identify him/her)
4. Gender (Gender of the employee, either male or female)
5. DateofJoin (The employee joining date)
6. PAMACT (Pamact of the employee)

“DateofJoin” and “PAMACT” had some null values which are cleaned through the process of data wrangling.

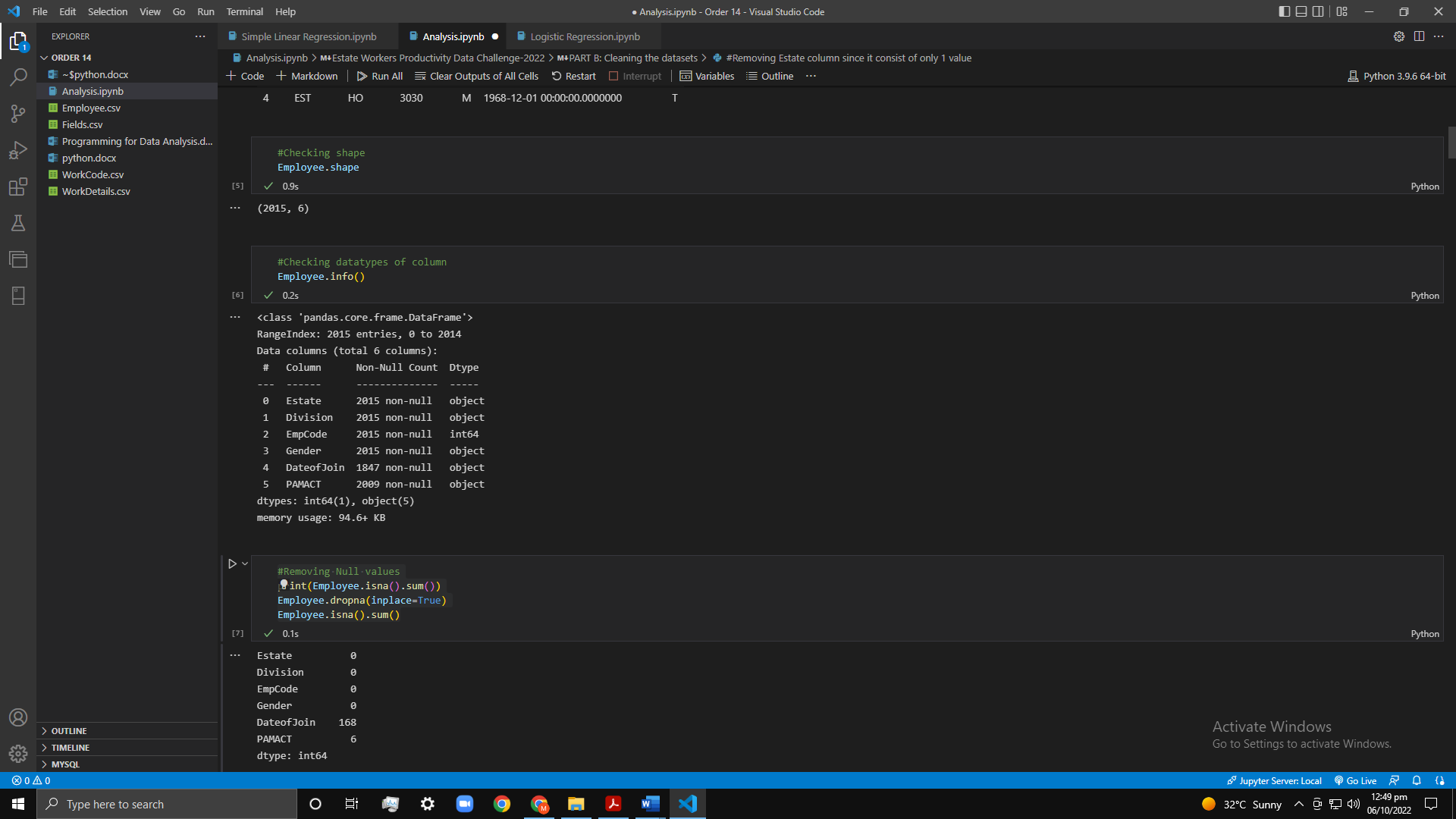


Figure 1.1 Employee.csv description

## Fields.csv

The fields.csv file contains 63 observations in 7 variables of estate, its division & crop cultivated related data. Out of 7 variables, 5 were string variables and 2 were numeric variable. Variables description is given as follows.

1. EState (Estate where the crop is planted)
2. Division (Division where the crop is planted)
3. CropType (Type of crop)
4. Field (Crop planted in which field section)
5. Area (Crop planted in which area)
6. NumberOfTree (Number of trees grown in the field)
7. Type (Crop condition i.e., either R(ripped),O(ongoing),C(cultivated))

“NumberOfTree” had some null values which are cleaned through the process of data wrangling.

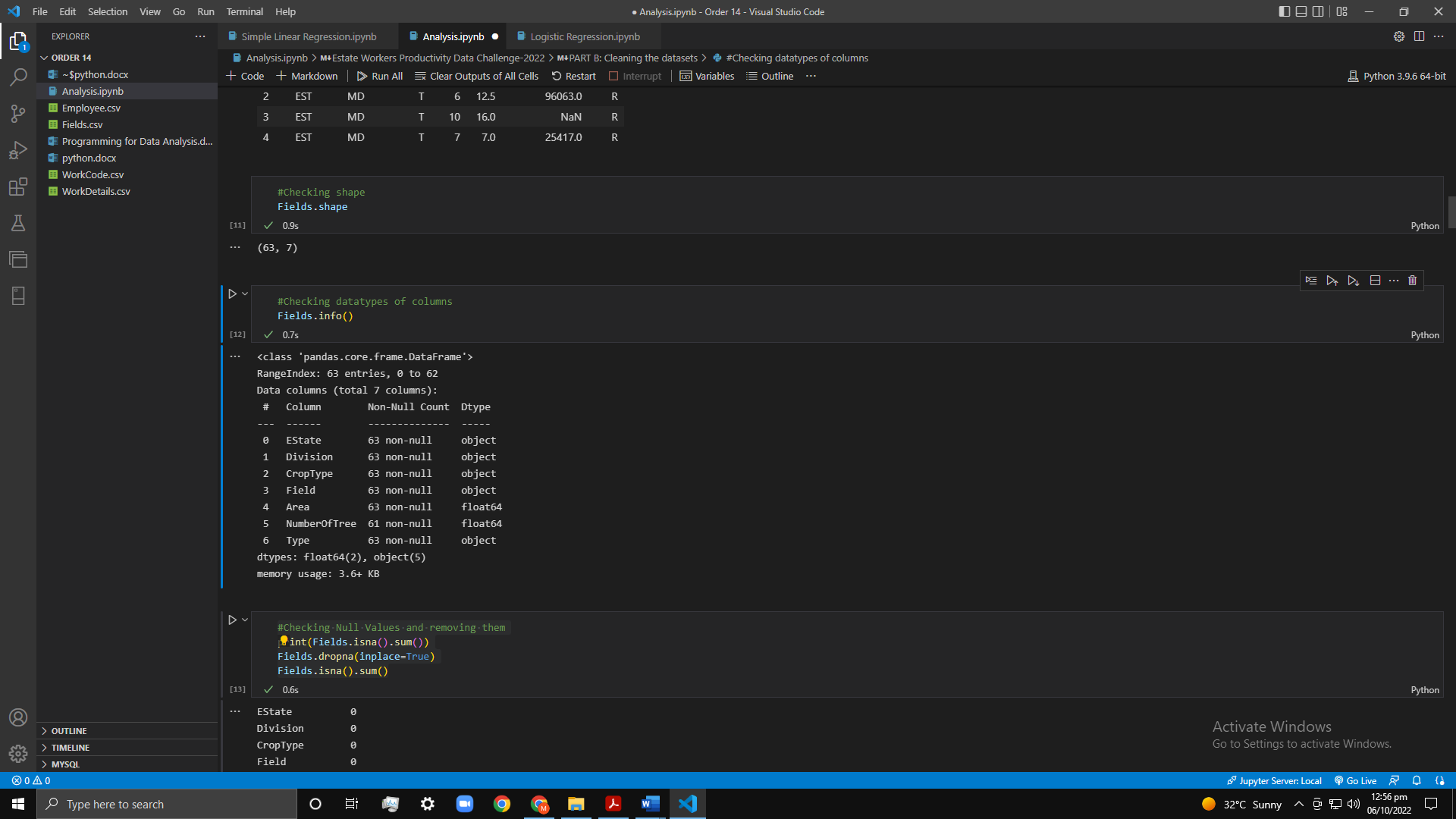


Figure 1.2 Field.csv description

## WorkCode.csv

The workcode.csv file contains 483 observations in 3 variables of work code information. It consisted of 3 string variables. Variables description is given as follows.

1. WorCode (Working category code)
2. Crop (Type of crop)
3. Type (Crop condition i.e., either R(ripped),O(ongoing),C(cultivated))

“Type” had some null values which are cleaned through the process of data wrangling.

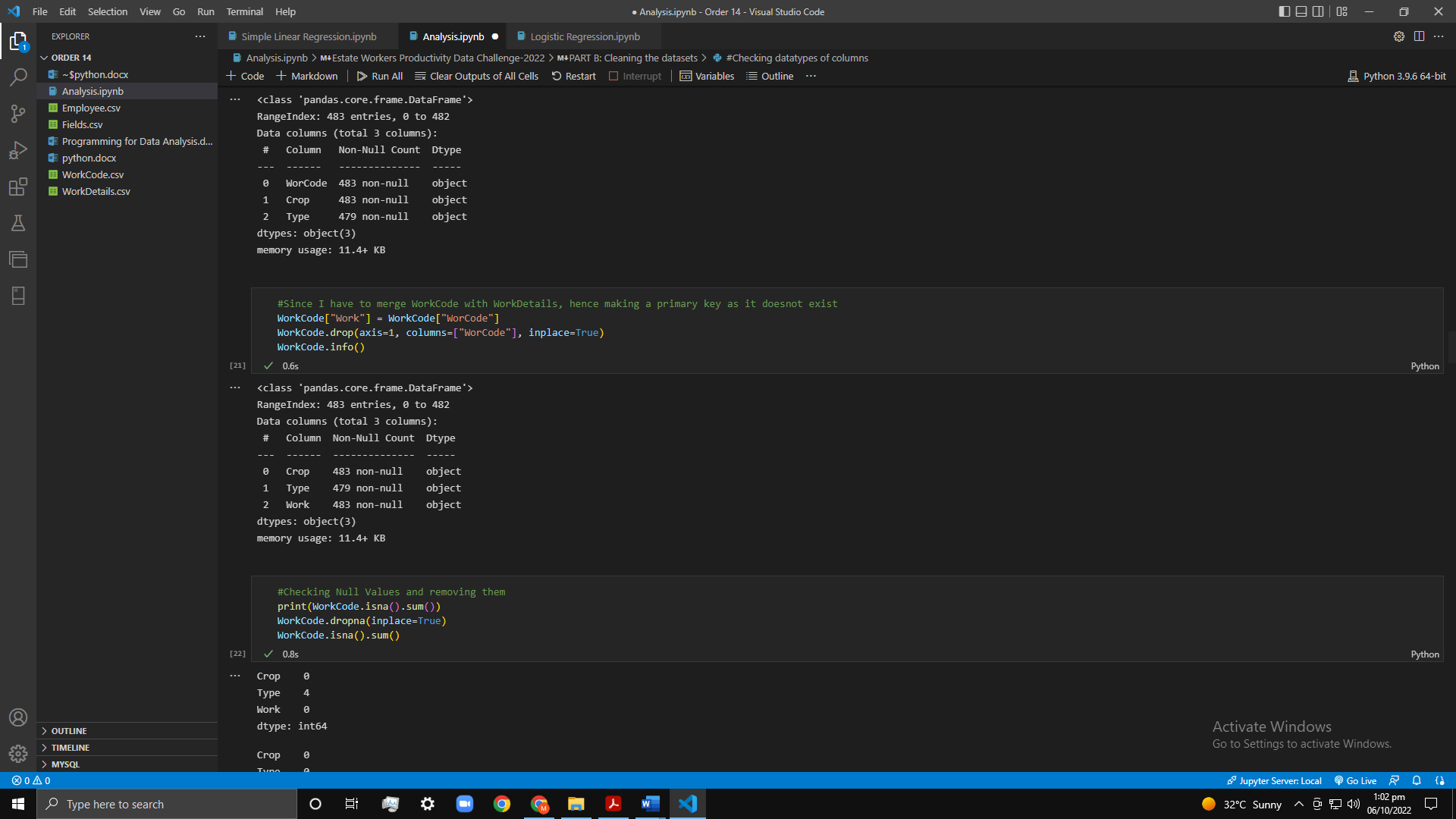


Figure 1.3 WorkCode.csv description

## WorkDetails.csv

The workdetails.csv file contains 1,110,902 observations in 10 variables of estate worker’s productivity related data during the period from 2012 to 2015. Out of 10 variables, 3 were string variables and 7 were numeric variable. Variables description is given as follows.

1. Year (name of year)
2. Month (month number)
3. Day (Day number)
4. Estate (Estate where the crop is planted)
5. Division (Division where the crop is planted)
6. EmpCode (EmpCode of employee, to uniquely identify him/her)
7. Work (Working category code)
8. NumberofDays (self-explanatory)
9. Qty (Quantity of crops)
10. ExtraKilos (Extra work done by employee in terms of kilos)

“Qty” and “ExtraKilos” had some null values which are cleaned through the process of data wrangling.

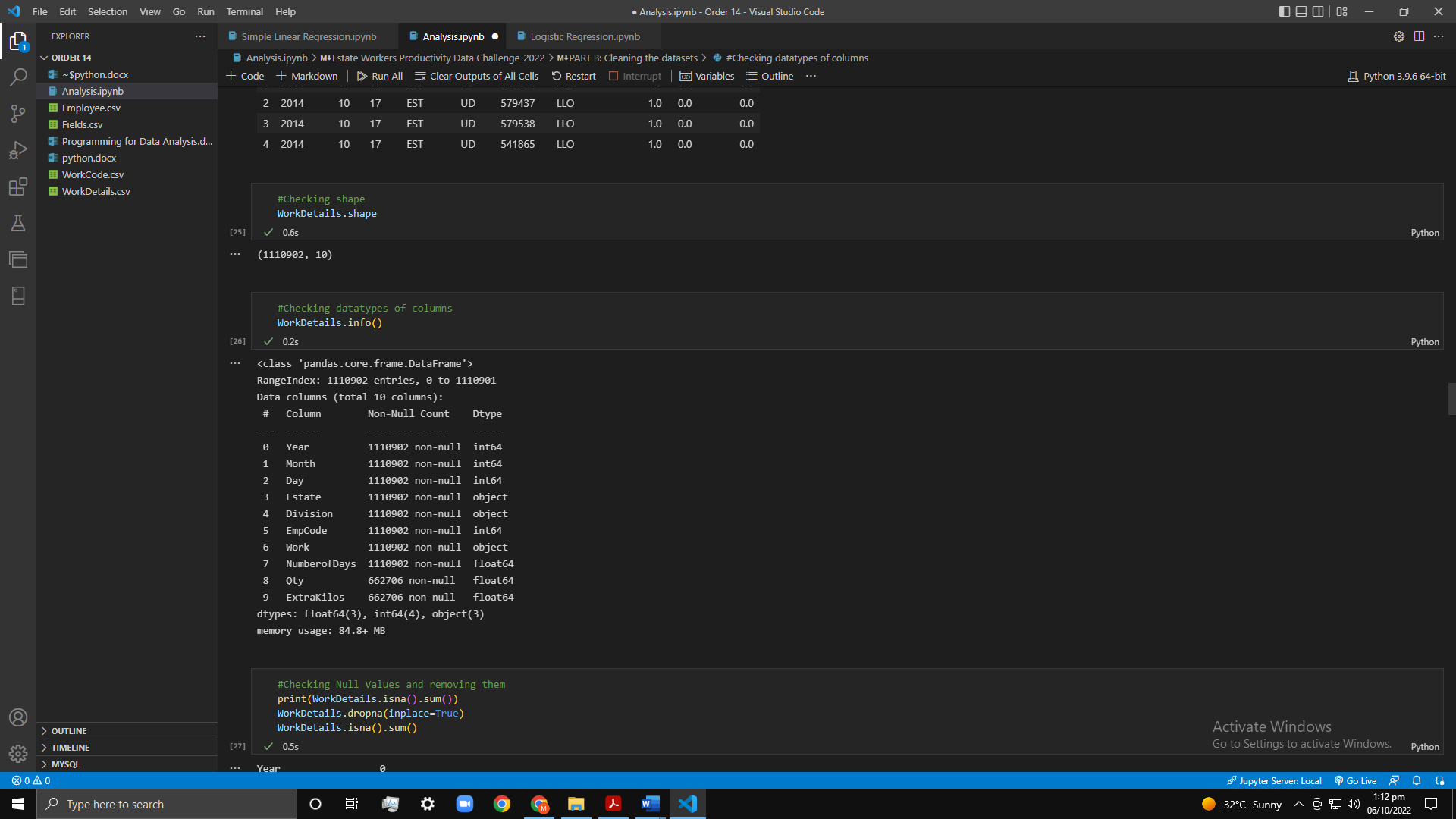


Figure 1.4 WorkDetails.csv description

# **Reading Data in python environment**

Following image shows that how the data is read in the python Jupyter notebook.



Figure 2.1 Reading datafiles in python

# **Data Cleaning**

Data cleaning is the process of preparing raw data for analysis by removing bad data, organizing the raw data, and filling in the null values. Ultimately, cleaning data prepares the data for the process of data mining when the most valuable information can be pulled from the data set. I have cleaned all the above-mentioned datasets using some steps which are as follows:

## Removing Null/Nan values

I first detected null values in each dataset using pandas library “isna()” method which gives the Boolean result that whether the value is null or not, then I used “sum()” method to count the number of null values. I then used the “dropna()” method to drop all those rows which have null/nan values.

Graphical user interface, text

Description automatically generated

Figure 3.1 Removing Null values from WorkDetails

## Removing Duplicated rows

I first calculated duplicated rows using “duplicates()” method then used “sum()” method to calculate the number of duplicated rows. After that I used “drop\_duplicates()” method to remove the duplicated row and only saving one row from them.

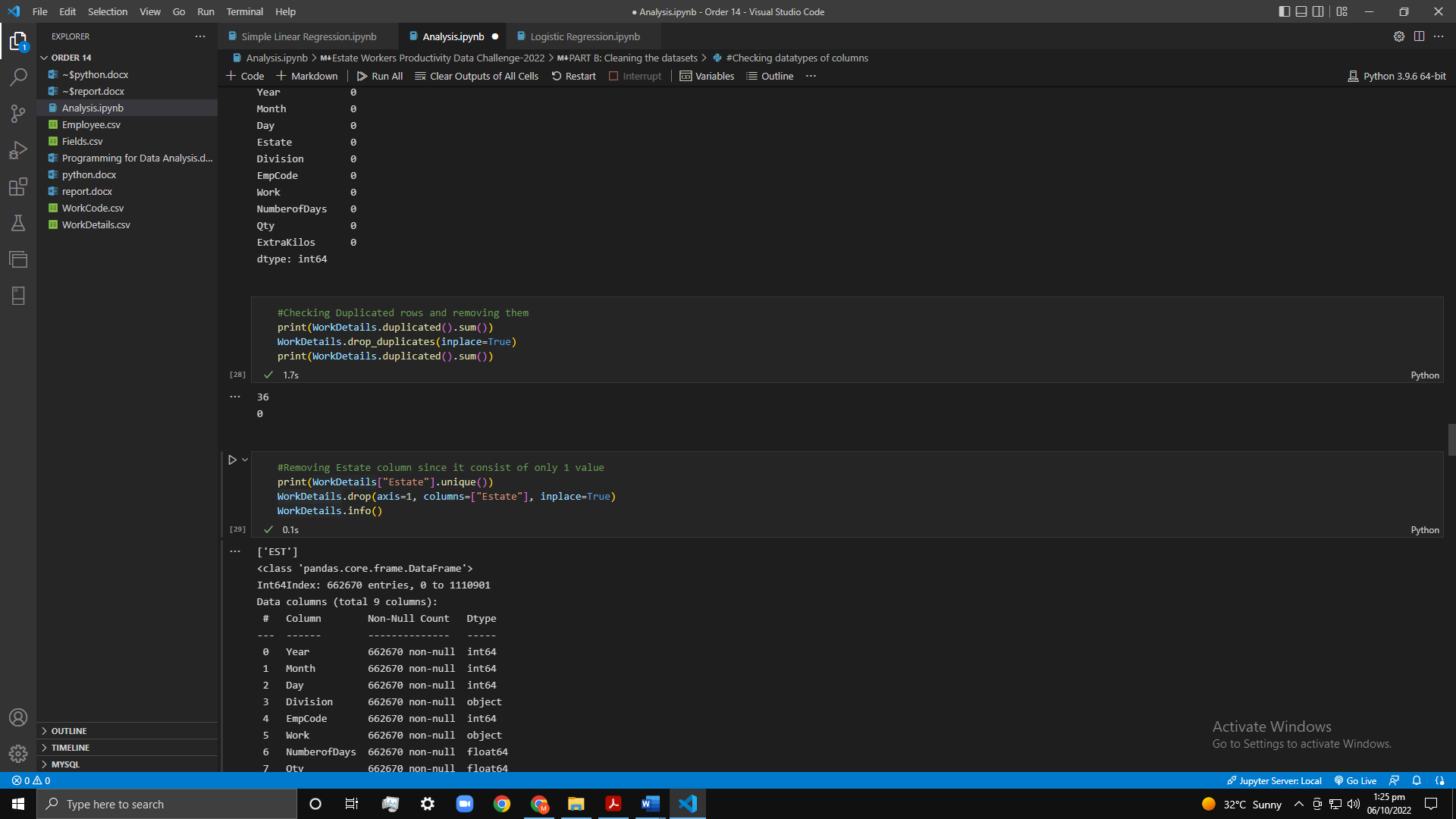


Figure 3.2 Removing duplicates from WorkDetails

## Removing outliers

An outlier is an observation that lies an abnormal distance from other values in a random sample from a population. I have removed some extreme outliers, by calculating 99% percentile and taking values till 99%, where the difference was great between normal values and extreme values. You can see I have not completely removed outliers, because they are not corrupting the data and they are required for the analysis to be valid.

Graphical user interface

Description automatically generated Graphical user interface, text

Description automatically generated

Figure 3.3(a) Detecting outliers Figure 3.3(b) After removing outliers

# **Merging of Datasets**

I merge all the datasets using “merge()” method. I have used “Full Outer-Join” to merge the datasets. A full outer join is a technique for joining tables that includes the unmatched rows from both tables in the final output. The join condition forms the basis for the matching. The null value is present in the result table for any column that doesn't have a value.

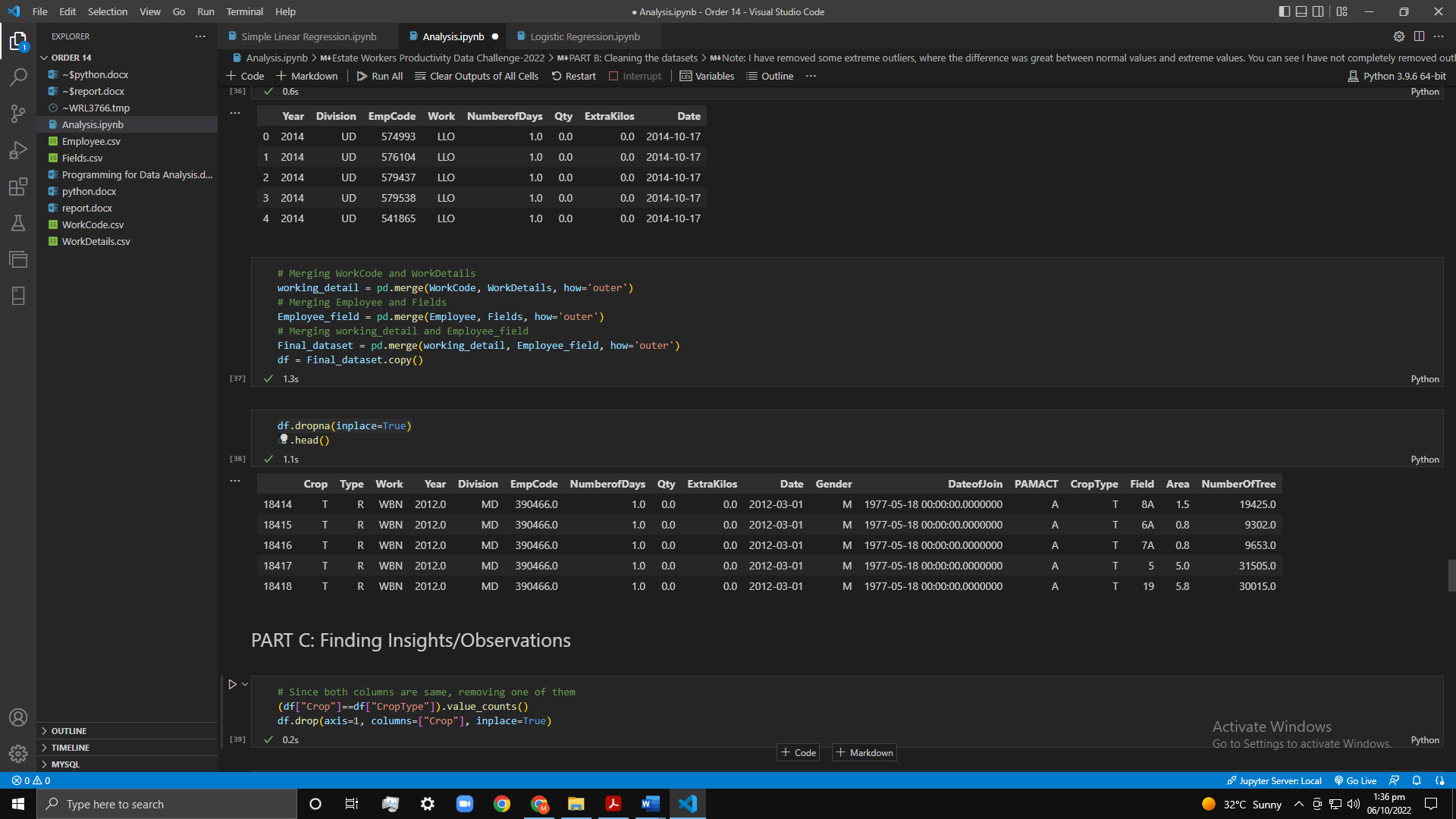


Figure 3.1 Merged final dataset

# **Insights/Patterns identified from merged dataset**

Data insights are the comprehensive understandings that come from information analysis on a specific topic for a person or organization. Organizations are able to make better judgments thanks to this in-depth insight than they would otherwise. Following insights were taken from the data.

1. Most crops are harvested in area 1.0. This can be easily shown through the bar graph.

Graphical user interface, application

Description automatically generated

Figure 4.1 Number of crops grown per area

1. Females worked more than the males. This can easily be shown through the bar graph.

Graphical user interface

Description automatically generated

Figure 4.2 Amount of work done per gender

1. “Extra Kilos” and “Qty” are highly positive correlated. Two variables that move together, or in the same direction, are said to have a positive correlation. When one variable rises while the other rises or when one variable fall while the other falls, there is a positive correlation. Theoretically, the same external forces can affect both of these separate variables because they travel in the same direction.

Chart

Description automatically generated

Figure 4.3(a) Using Spearman's Correlation

Chart

Description automatically generated

Figure 4.3(b) Using Pearson's Correlation

## **Feature Selection**

With the use of just pertinent data and the elimination of irrelevant data, feature selection is a technique for lowering the input variable for your model. It involves automatically selecting features for your machine learning model that are pertinent to the problem you are attempting to solve. We accomplish this by adding or removing significant features without altering them. It assists in minimizing the amount of noise in our data and the quantity of our input data.

What goes in, comes out is a basic tenet of machine learning models. We can anticipate that the outcome will be garbage if we input junk into our model. Garbage in this context refers to noise in our data. We gather vast amounts of data to train a model in order to improve machine learning. Typically, a sizable amount of the data obtained is noise, and some of the columns in our dataset may not have a major impact on how well our model performs. In addition, training a model can take longer when there is a lot of data available. The model can become erroneous as a result of learning from this unimportant data.

I have used Filter Method here, in this method, features are eliminated in accordance with how they relate to or correlate with the output. We use correlation to determine whether a feature's connection to the output labels is positive or negative, and we delete features as necessary. For instance, Fisher's Score, the Chi-Square Test, and information gain.

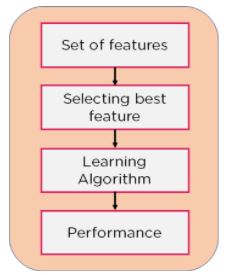


Figure 5.1 Filter Method

The relevant features after feature selection were the following:

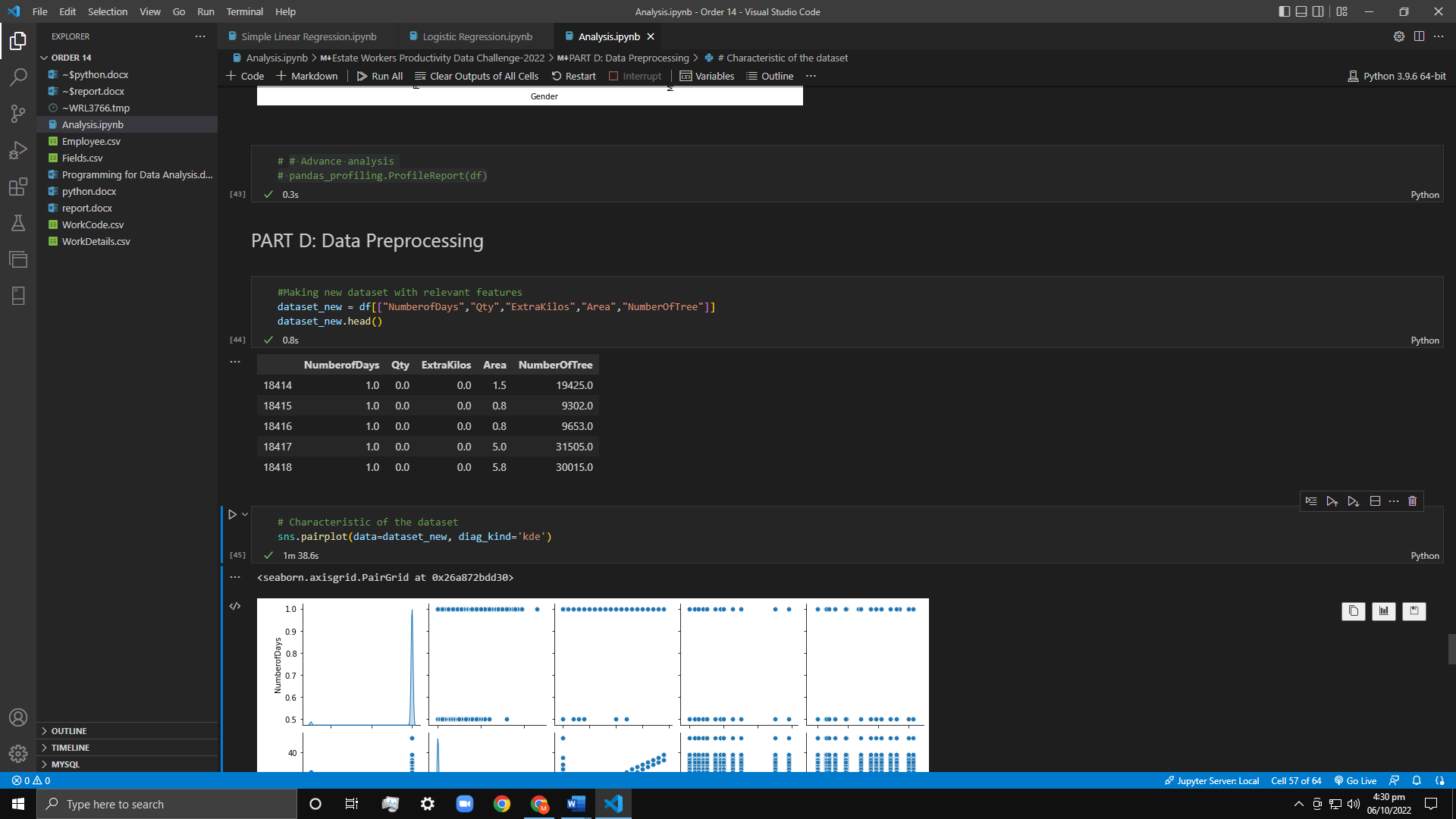


Figure 5.2 Relevant Features

# **Train-Test Split**

## What is Train-Test Split?

The train-test split is a method for assessing how well a machine learning algorithm is working. It works with any supervised learning method and may be applied to classification or regression issues. A dataset is split into two subsets as part of the method. The first subset, often known as the training dataset, is utilized to fit the model. Instead of using the second subset to train the model, the input element of the dataset is given to it, and predictions are then made and compared to the expected values. The test dataset is this second dataset's official name.

* Train Dataset: Used to fit the machine learning model.
* Test Dataset: Used to evaluate the fit machine learning model.

The goal is to gauge the machine learning model's performance on fresh data—data that were not used to train the model. We anticipate applying the model in this way. In other words, to fit it to the data that is currently available with inputs and outputs that are known, then to forecast on fresh examples in the future where we do not have the anticipated output or target values. When a suitable size dataset is provided, the train-test procedure is appropriate. (Guido, 2016)

## When to Use the Train-Test Split

Each challenge involving predictive modelling has its own definition of what "sufficiently large" means. It indicates that there is sufficient information to divide the dataset into train and test datasets, and that each of the train and test datasets is an appropriate representation of the problem domain. For this to work, the initial dataset must also be an accurate reflection of the problem domain. There should be enough entries to cover all common cases and the majority of exceptional cases in the problem domain, according to a reasonable representation of the problem domain. This could refer to combinations of practice-observed input variables. Thousands, hundreds of thousands, or even millions of examples might be necessary. On the other hand, a limited dataset makes the train-test method inappropriate. The issue is that there won't be enough data in the training dataset for the model to successfully develop an input-to-output mapping when the dataset is divided into train and test sets. Additionally, there won't be sufficient data in the test set to accurately assess the model's performance. The anticipated performance could be excessively positive (excellent) or negative (poor) (bad).

The k-fold cross-validation approach is a good alternative model evaluation method if you don't have enough data. Computational efficiency is another justification for using the train-test split assessment method in addition to dataset size. When a model requires a lot of money to train, repeated evaluation employed in other processes becomes impossible. Deep neural network models are an example. In this situation, the train-test method is frequently applied.

Alternatively, a project can have an effective model and a sizable dataset, but it might also need a rapid estimate of model performance. In this instance, the train-test split process is used. Random selection is used to divide the original training dataset's samples into the two subsets. To make sure that the train and test datasets are accurate representations of the original dataset, this is done.

## How to Configure the Train-Test Split

The size of the train and test sets serves as the procedure's key configurable parameter. For either the train or test datasets, this is most frequently given as a percentage that ranges from 0 to 1. For instance, if the size of the training set is 0.67 (67 percent), the test set will receive the leftover percentage of 0.33 (33 percent). There is no ideal split ratio. You must select a split percentage that satisfies the goals of your project considering factors like:

* computation fees incurred during model training.
* Cost of computing when analyzing the model
* representativeness of the training set.
* representativeness of the test set

However, typical split percentages consist of:

* Train: 80%, Test: 20%
* Train: 67%, Test: 33%
* Train: 50%, Test: 50%

In my program, I have used 80:20 ratio i.e., 80% training data and 20% testing data. I have used scikit learn library and from sklearn.model\_selection, I have used train\_test\_split() method which takes other features and target variable as X, Y, gives the result in the form of 1-D series.

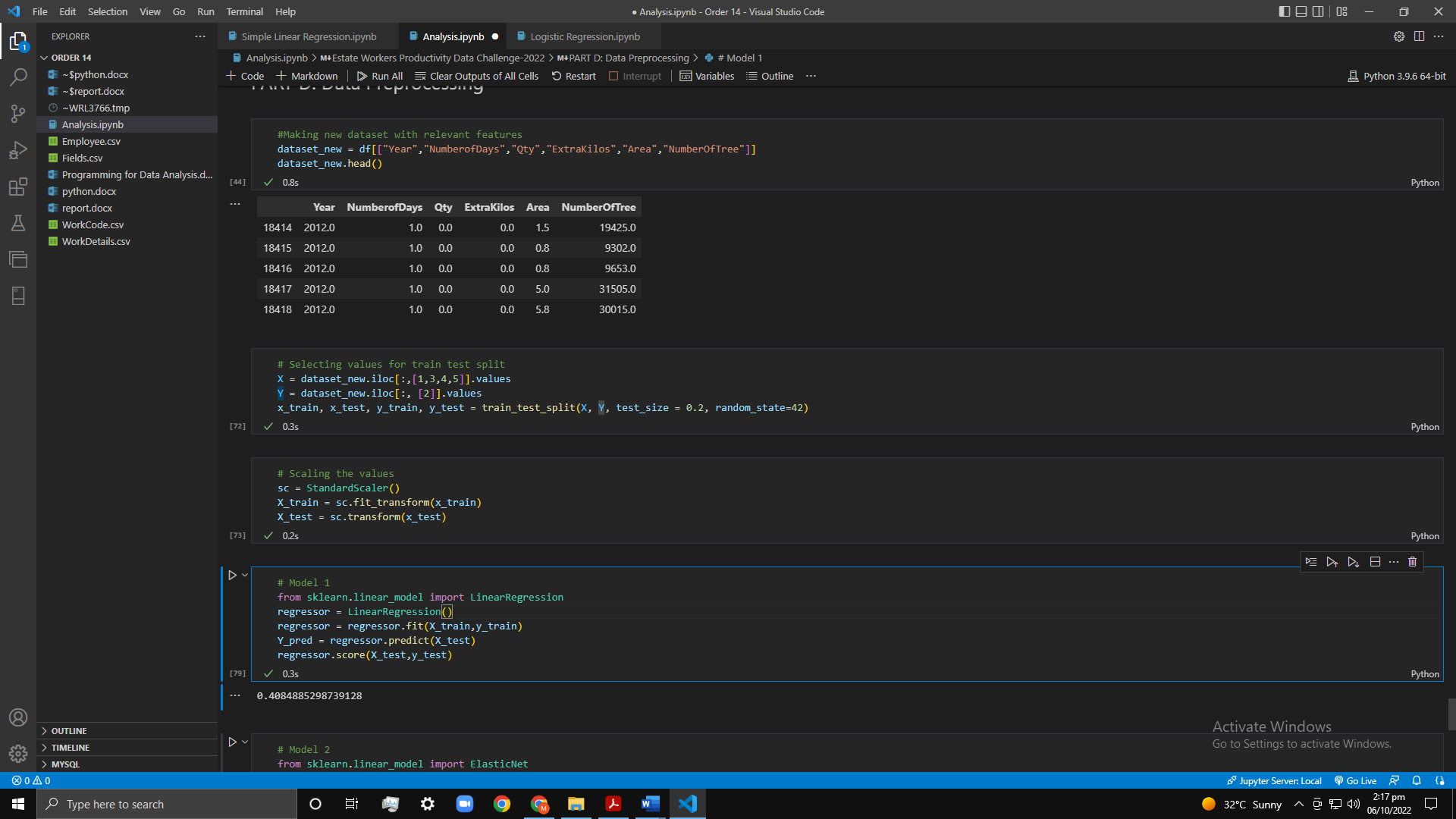


Figure 6.1 Train Test Split

# **Feature Scaling**

Feature scaling is a method for uniformly distributing the independent features in the data over a predetermined range. It is done as part of the pre-processing of the data to deal with extremely variable magnitudes, values, or units. In the absence of feature scaling, a machine learning algorithm would often priorities larger values over smaller ones, regardless of the unit of measurement. I have used Standard Scaling to normalize the features. Standardization is another scaling technique where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation. In this case, the values are not restricted to a particular range. Here’s the formula for standardization:



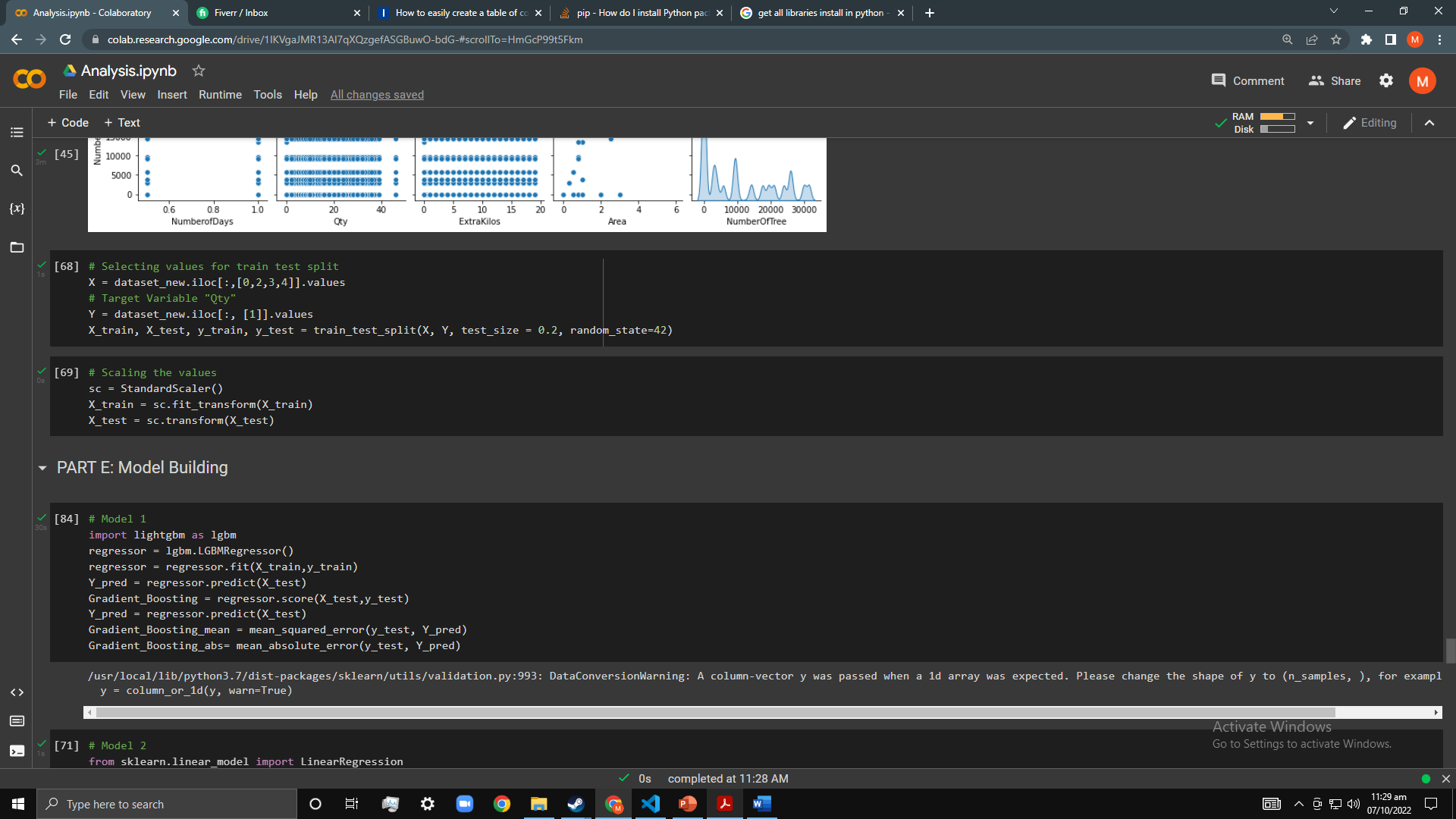


Figure 7.1 Scaling the features

# **Machine Learning approaches**

I have assumed that the problem of crop yield prediction is a regression problem. Regression is a method for determining how independent features or variables relate to a dependent feature or result. It is a technique for machine learning predictive modelling, where an algorithm is used to forecast continuous outcomes.

Any forecasting or predictive model must include regression analysis, making it a common technique in machine learning-powered predictive analytics. Regression is a typical application for supervised machine learning models in addition to classification. The input and output training data for this method of training models have to be labelled. Accurately labelled training data is essential because machine learning regression models need to grasp the link between features and outcome variables.

Here the target variable is “Qty”. Following machine learning techniques are used:

## LightGBM Regression

It stands for “Light Gradient Boosting Machine”. It is a gradient boosting framework that uses tree-based learning algorithms and is regarded as one of the most powerful computation-based algorithms. It is regarded as a processing algorithm with quick speeds. LightGBM algorithm grows vertically, meaning it grows leaf-wise and other algorithms grow level-wise, whereas other algorithms' trees expand horizontally. LightGBM selects the leaf with a significant loss for growth. When expanding the same leaf, it can reduce loss more than a level-wise strategy. I have used “GOSS” as boosting operation which stands for “Gradient-Based One Side Sampling”.

When calculating the gain, GOSS gives higher weight to data points with larger gradients. Instances that haven't been used effectively for training more in this method. To preserve accuracy, some data points are randomly deleted from the analysis, and some are kept. Given the same sampling rate, this strategy is often superior to random sampling.

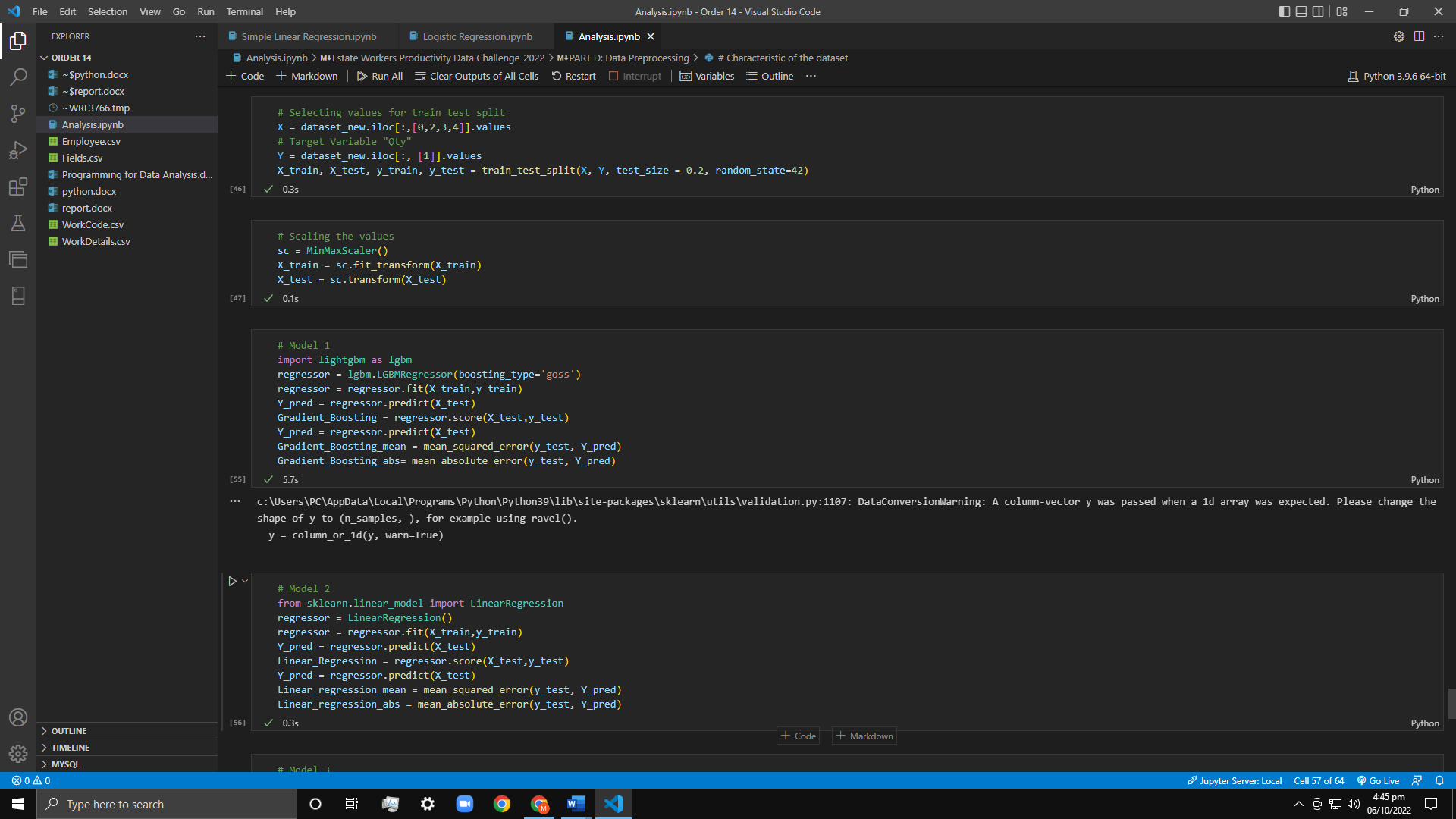


Figure 8.1 LightGBM Regression Model

## Multiple Linear Regression

A fundamental and widely used form of predictive analysis is linear regression. Regression analysis' main goal is to look at two things: (1) Is it possible to accurately forecast an outcome (dependent) variable using a set of predictor variables? (2) Which individual variables—as shown by the size and sign of the beta estimates—are highly important predictors of the outcome variable, and how do they affect the outcome variable? The link between one dependent variable and one or more independent variables is explained using these regression estimations.

Text

Description automatically generated

Figure 8.2 Linear Regression Model

## Elastic Net Regression

Elastic net regression regularizes regression models by using the penalties from the lasso and ridge procedures. In order to improve the regularization of statistical models, the strategy combines the lasso and ridge regression approaches. The lasso approach's drawbacks—namely, the fact that it only needs a few samples for highly dimensional data—are improved by the elastic net method. The inclusion of "n" numbers of variables up until saturation is possible with the elastic net approach. When the variables are grouped into highly correlated groups, lasso usually selects one variable from each group while completely ignoring the others.

Text

Description automatically generated

Figure 8.3 Elastic Net Regression Model

## Bayesian Ridge Regression

[Bayesian](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.BayesianRidge.html#sklearn.linear_model.BayesianRidge) Ridge estimates a probabilistic model of the regression problem as described above. The prior for the coefficient w is given by a spherical Gaussian:

The priors over α and λ are chosen to be gamma distributions, the conjugate prior for the precision of the Gaussian. The resulting model is called Bayesian Ridge Regression and is like the classical Ridge.

The parameters w, α and λ are estimated jointly during the fit of the model, the regularization parameters α and λ being estimated by maximizing the log marginal likelihood.

A screenshot of a computer

Description automatically generated

Figure 8.4 Bayesian Ridge Regression Model

# **Accuracy of all applied ML techniques**

Accuracy is one metric for evaluating classification models. Informally, **accuracy** is the fraction of predictions our model got right.

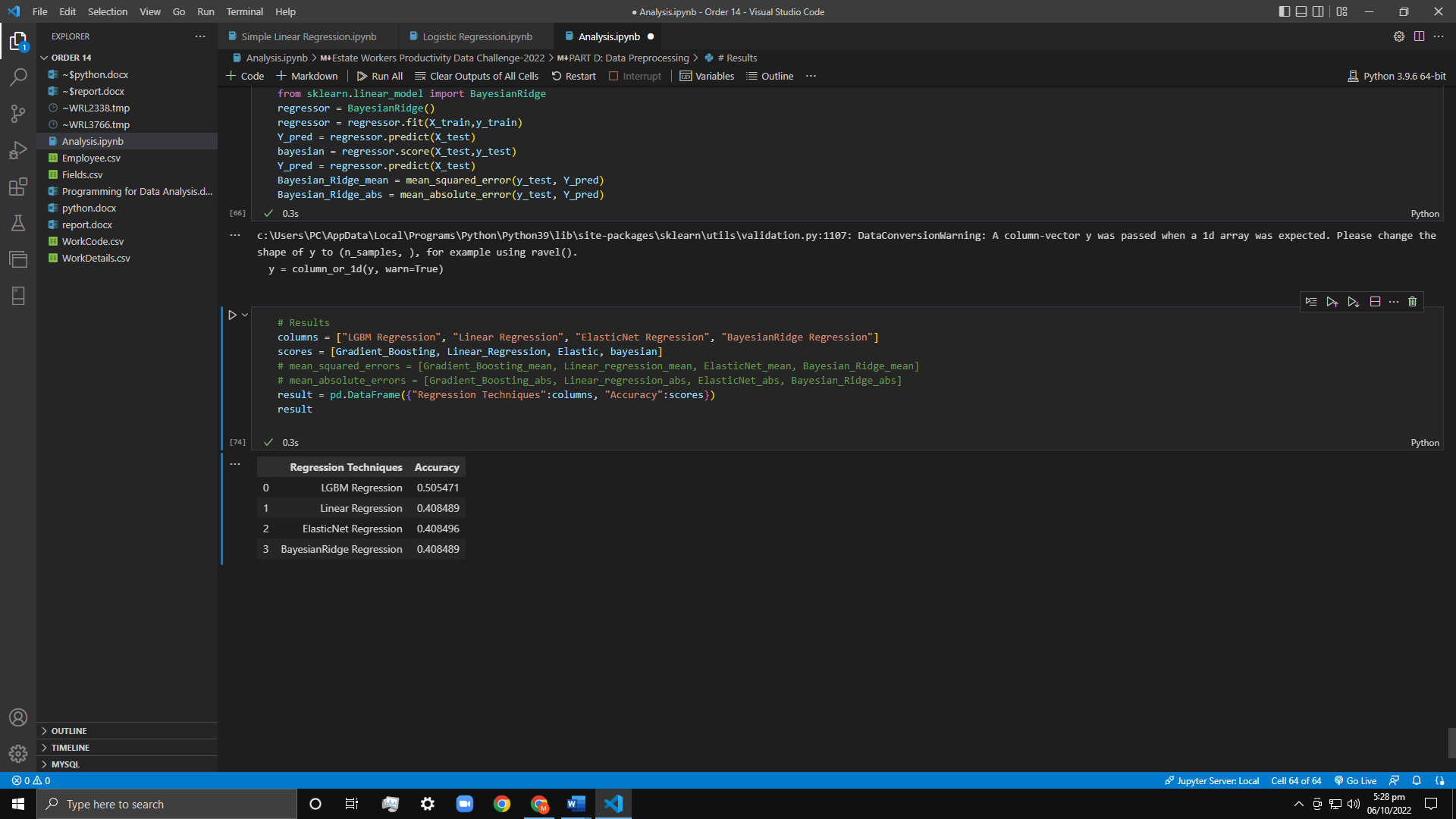


Figure 9.1 Accuracy of all models

# **Alternate ways to increase performance of the models**

Increasing a model's performance can occasionally be difficult. Following are the ways in which we can do this.

## Different feature scaling technique

We can apply different feature scaling technique such as Normalization. Normalization is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1. It is also known as Min-Max scaling. Here, Xmax and Xmin are the maximum and the minimum values of the feature respectively.



* When the value of X is the minimum value in the column, the numerator will be 0, and hence X’ is 0
* On the other hand, when the value of X is the maximum value in the column, the numerator is equal to the denominator and thus the value of X’ is 1
* If the value of X is between the minimum and the maximum value, then the value of X’ is between 0 and 1

## Hyperparameters tuning

In most of the ML models you’ll have multiple hyperparameters. Choosing the best combination requires an understanding of the models parameters and the business problem you’re trying to tackle. So before doing anything you have to know the hyperparameters of the models and their importance. To get the best hyperparameter the two-step procedure is followed .

* For each combination of hyperparameters the model is evaluated
* The combination that gives the best performing model are selected as optimal

Finding the best value for each parameter is the goal of parameter tuning in order to increase the model's accuracy. You must have a solid grasp of the significance of each parameter and how it affects the model individually in order to fine-tune it. This procedure can be repeated with numerous effective models. We can used *GridSearchCV* or *RandomSearchCV* to find the best hyperparameters in Sklearn. (Géron, 2019)

## Ensemble methods

This is the most typical strategy that dominates winning data science competition answers. This method merely aggregates the output of several weak models to yield superior results. This can be done in a variety of ways:

* Boosting
* Bagging (Bootstrap Aggregating)

It is always a better idea to apply ensemble methods to improve the accuracy of your model. There are two good reasons for this:

* They are generally more complex than traditional methods.
* The traditional methods give you a good base level from which you can improve and draw from to create your ensembles.

## Cross Validation

Cross Validation is one of the most important concepts in data modeling. It says, try to leave a sample on which you do not train the model and test the model on this sample before finalizing the model.

## Feature transformation

Feature transformation is needed in a number of situations, including the following:

A) Converting a variable's scale from its initial scale to a scale between zero and one. The normalization of data is what this is. For instance: Before using any algorithm, it is necessary to normalize the variables in a data set if the first variable is measured in meters, the second in centimeters, and the third in kilograms.

B) Some algorithms function well with data that is typically distributed. As a result, we must eliminate the variable's skewness (s). Skewness can be eliminated using techniques like log, square root, or inverse of the data.

## Feature Creation

Feature creation is the process of creating new variables from preexisting variables. It aids in revealing a data set's hidden relationships. Let's imagine that based on transaction dates, we want to estimate the volume of transactions in a given store. Transaction dates may not directly correlate with transaction volume in this case, but if we look at the day of the week, there may be a stronger correlation. The day of the week is concealed in this instance. In order to improve the model, we must extract it.

# **Comparative Analysis**

Now we will perform comparative analysis on the mentioned Machine Learning Model

## Accuracy

The accuracy of LGBM Regression is highest since it is using GOSS as its optimizer. The other models accuracies are almost equal.

Graphical user interface, application, website

Description automatically generated

Figure 11.1 Accuracies of different models

## Mean Squared Error

You may determine how closely a regression line resembles a set of points using the mean squared error (MSE). This is accomplished by squaring the distances between the points and the regression line (also known as the "errors"). The squaring is required to eliminate any unfavorable indications. Additionally, it emphasizes bigger discrepancies. Since you're averaging a collection of errors, this error type is known as the mean squared error. The forecast is more accurate the lower the MSE. Here the MSE of LGBM is lowest

Graphical user interface, application

Description automatically generated

Figure 11.2 MSE of different models

## Mean Absolute Error

A model evaluation statistic used with regression models is mean absolute error. The average of the absolute values of each prediction error over all test set instances is the mean absolute error of a model with respect to the test set. The difference between the instance's true value and the expected value represents each prediction error. Here the MAE of LGBM is lowest

Graphical user interface

Description automatically generated

Figure 11.3 MSE of different models

## Limitations Of each model

1. LGBM Regression

* Light GBM split the tree leaf-wise which can lead to overfitting as it produces much complex trees.
* Light GBM is sensitive to overfitting and thus can easily overfit small data.

1. Linear Regression

* Since linear regression assumes a linear relationship between the input and output variables, it fails to fit complex datasets properly. In most real-life scenarios, the relationship between the variables of the dataset isn't linear and hence a straight line doesn't fit the data properly.
* Outliers can have a very big impact on linear regression's performance and hence they must be dealt with appropriately before linear regression is applied on the dataset.

1. Elastic Net Regression

* One disadvantage is the computational cost. You need to cross-validate the relative weight of L1 vs. L2 penalty, α, and that increases the computational cost by the number of values in the α grid.

1. Bayesian Ridge Regression

* The inference of the model can be time-consuming.
* If there is a large amount of data available for our dataset, the Bayesian approach is not worth it and the regular frequentist approach does a more efficient job.

# Libraries Used

Following python libraries are used in this assignment.

* Pandas
* NumPy
* Matplotlib
* Seaborn
* Scikit-Learn
* Pandas-Profiling
* Light GBM

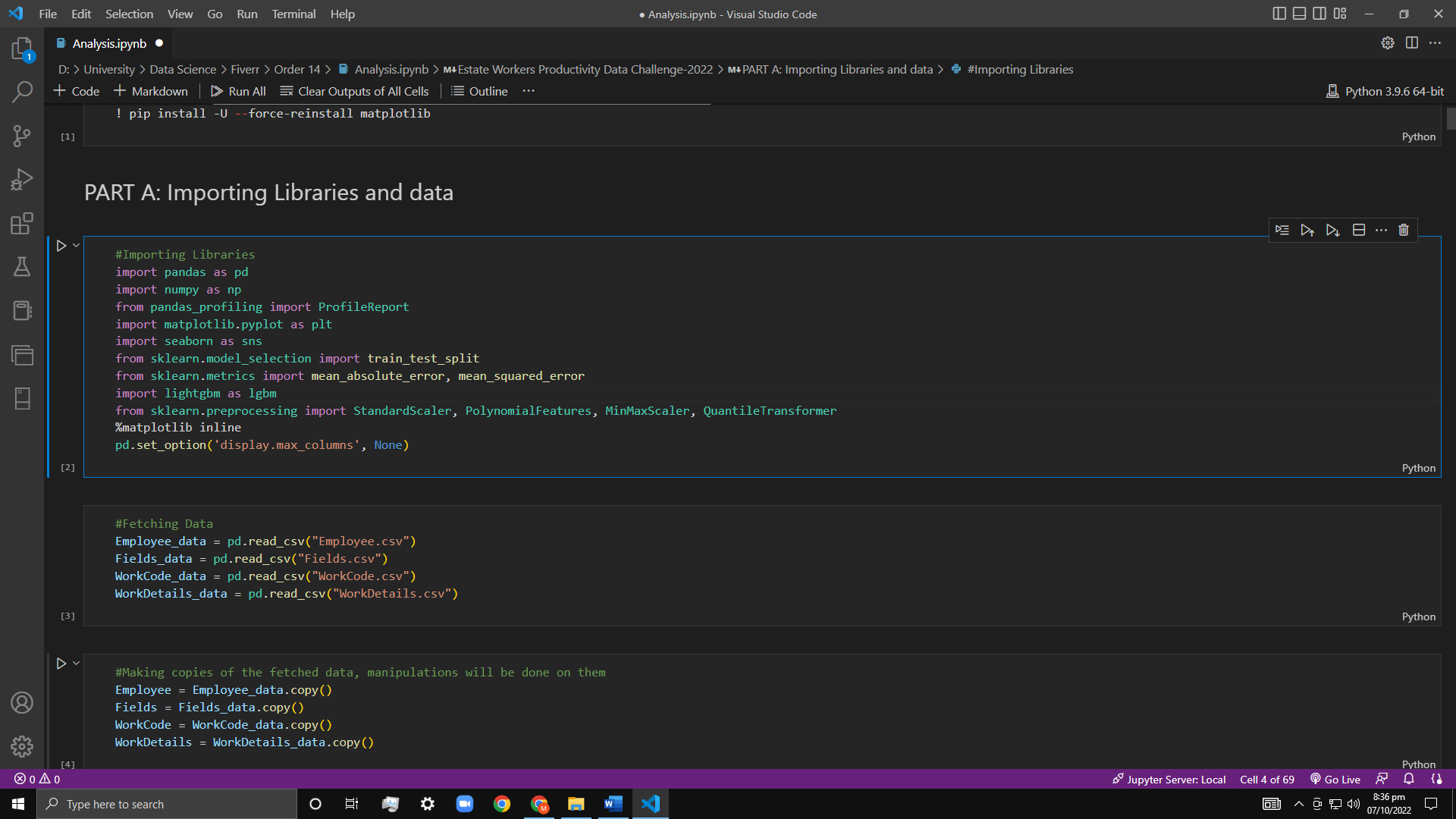


Figure 13.1 Libraries Used

# Knowledge gained

I gained knowledge about various domains of data science. I now know how to properly perform data cleaning and data wrangling on the vast datasets. I now can properly remove outliers, null values, and duplicate rows from the vast datasets. I observe how the data can spill the meaningful insights just by teasing it a little bit. One of the main and intriguing things I learnt is that how a Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. I learnt how Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments. I learnt how to properly develop a machine learning pipeline, how to perform Exploratory data analysis, feature engineering, feature scaling and model selection. I learnt python’s different libraries such as scikit-learn.

# References

Géron, A., 2019. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow.* SECOND EDITION ed. Chicago: O’Reilly.

Guido, A. C. M. a. S., 2016. *Introduction to Machine Learning with Python.* First Edition ed. Berlin: O’Reilly.