Data Analysis Report on Cars24 Dataset

1. Objective

The goal of this analysis is to clean, preprocess, explore, and model the data provided in the Cars24 dataset. Key tasks include handling missing values, outlier detection, univariate and bivariate analysis, feature engineering, and building a regression model to predict the rating_engineTransmission target variable.

2. Data Overview

- **Dataset**: The data was loaded from data (1).xlsx containing two sheets: data (primary dataset) and fields summary (metadata).
- **Rows and Columns**:
  - Initial dataset shape: (number_of_rows, number_of_columns)--> (26307, 73)
  - 52 columns have more than 40% missing values.

- **Summary**:
  - General data statistics and metadata description were checked using .info() and .describe() functions.

```
df.describe()
```

	inspectionStartTime	year	month	engineTransmission_engineOil_cc_value_9	engineTransmission_engine_cc_value_10	odometer_reading	rating_engineTransmission
count	26307	26307.000000	26307.000000	0.0	0.0	26307.000000	26307.000000
mean	2019-02-24 04:17:07.174554624	2010.856578	5.462006	NaN	NaN	76460.143764	3.624663
min	2019-01-02 10:02:34	1989.000000	1.000000	NaN	NaN	1.000000	0.500000
25%	2019-01-29 13:54:15.500000	2008.000000	2.000000	NaN	NaN	46396.000000	3.500000
50%	2019-02-24 12:57:51	2011.000000	5.000000	NaN	NaN	72013.000000	4.000000
75%	2019-03-23 10:11:12.500000	2014.000000	9.000000	NaN	NaN	98289.500000	4.000000
max	2019-04-15 12:47:00	2019.000000	12.000000	NaN	NaN	999999.000000	5.000000
std	NaN	3.766234	3.583866	NaN	NaN	46762.524489	0.847645

```
[9] df.info()
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26307 entries, 0 to 26306
Data columns (total 73 columns):
 #   Column                                               Non-Null Count  Dtype
---  ------                                               --------------  -----
 0   appointmentId                                        26307 non-null  object
 1   inspectionStartTime                                  26307 non-null  datetime64[ns]
 2   year                                                 26307 non-null  int64
 3   month                                                26307 non-null  int64
 4   engineTransmission_battery_value                     26307 non-null  object
 5   engineTransmission_battery_cc_value_0                3438 non-null   object
 6   engineTransmission_battery_cc_value_1                430 non-null    object
 7   engineTransmission_battery_cc_value_2                72 non-null     object
 8   engineTransmission_battery_cc_value_3                16 non-null     object
 9   engineTransmission_battery_cc_value_4                4 non-null      object
 10  engineTransmission_engineoilLevelDipstick_value      26307 non-null  object
 11  engineTransmission_engineOilLevelDipstick_cc_value_0 411 non-null    object
 12  engineTransmission_engineOil                         26307 non-null  object
 13  engineTransmission_engineOil_cc_value_0              18557 non-null  object
 14  engineTransmission_engineOil_cc_value_1              11004 non-null  object
 15  engineTransmission_engineOil_cc_value_2              6593 non-null   object
 16  engineTransmission_engineOil_cc_value_3              3742 non-null   object
 17  engineTransmission_engineOil_cc_value_4              1772 non-null   object
 18  engineTransmission_engineOil_cc_value_5              609 non-null    object
 19  engineTransmission_engineOil_cc_value_6              121 non-null    object
 20  engineTransmission_engineOil_cc_value_7              11 non-null     object
 21  engineTransmission_engineOil_cc_value_8              2 non-null      object
 22  engineTransmission_engineOil_cc_value_9              0 non-null      float64
 23  engineTransmission_engine_value                      26307 non-null  object
 24  engineTransmission_engine_cc_value_0                 9070 non-null   object
 25  engineTransmission_engine_cc_value_1                 5084 non-null   object
 26  engineTransmission_engine_cc_value_2                 2374 non-null   object
 27  engineTransmission_engine_cc_value_3                 904 non-null    object
 28  engineTransmission_engine_cc_value_4                 296 non-null    object
 29  engineTransmission_engine_cc_value_5                 92 non-null     object
 30  engineTransmission_engine_cc_value_6                 37 non-null     object
 31  engineTransmission_engine_cc_value_7                 8 non-null      object
 32  engineTransmission_engine_cc_value_8                 4 non-null      object
 33  engineTransmission_engine_cc_value_9                 3 non-null      object
```

3. Data Preprocessing

3.1 Handling Missing Values

- Columns with missing values were identified.
- Columns described as "current condition if not yes" were imputed with 'yes' in empty rows.
- For other columns with missing values:
  - Unique values and counts were explored.
  - Missing values were examined for further imputation or processing.

3.2 Removing Unnecessary Columns

- Columns described as "comments" in the metadata were dropped.

3.3 DateTime Feature Engineering

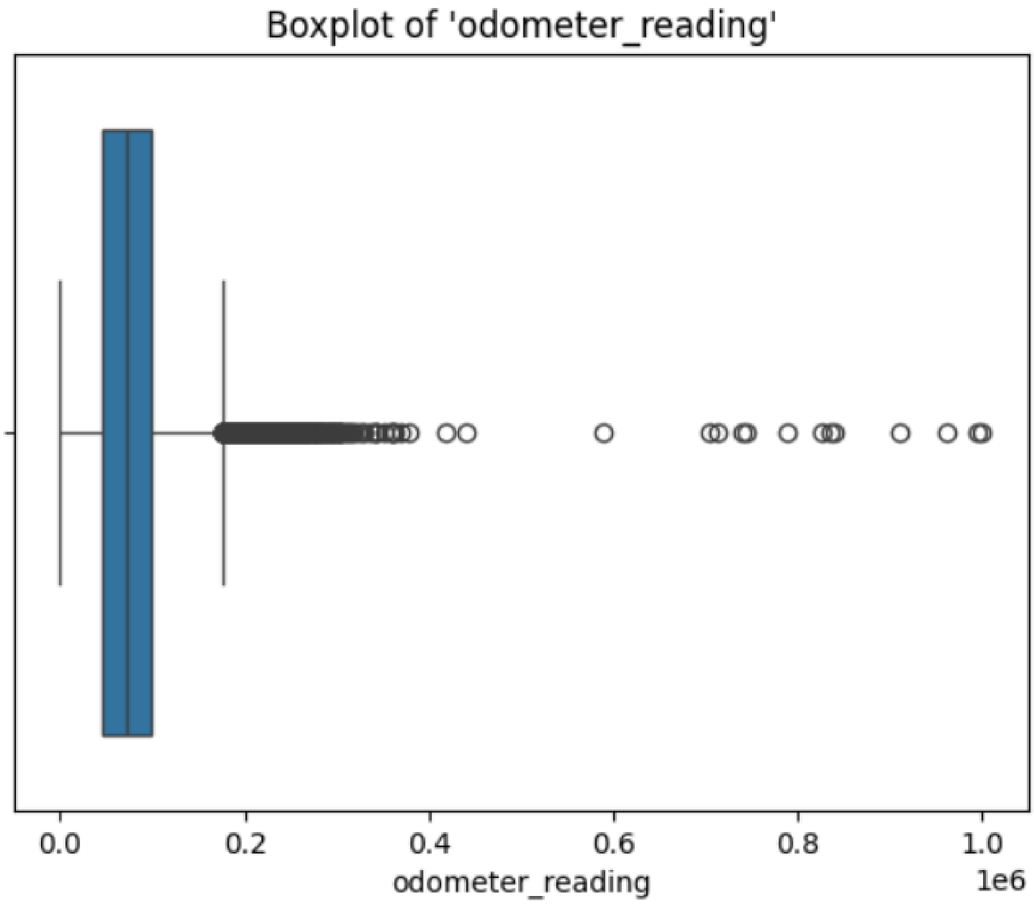
- From the inspectionStartTime column, new time-based features were extracted:
    - inspection_day, inspection_month, inspection_year, and inspection_hour.
- A Vehicle_age column was created to determine the vehicle's age using the difference between inspection_year and year.

4. Exploratory Data Analysis (EDA)

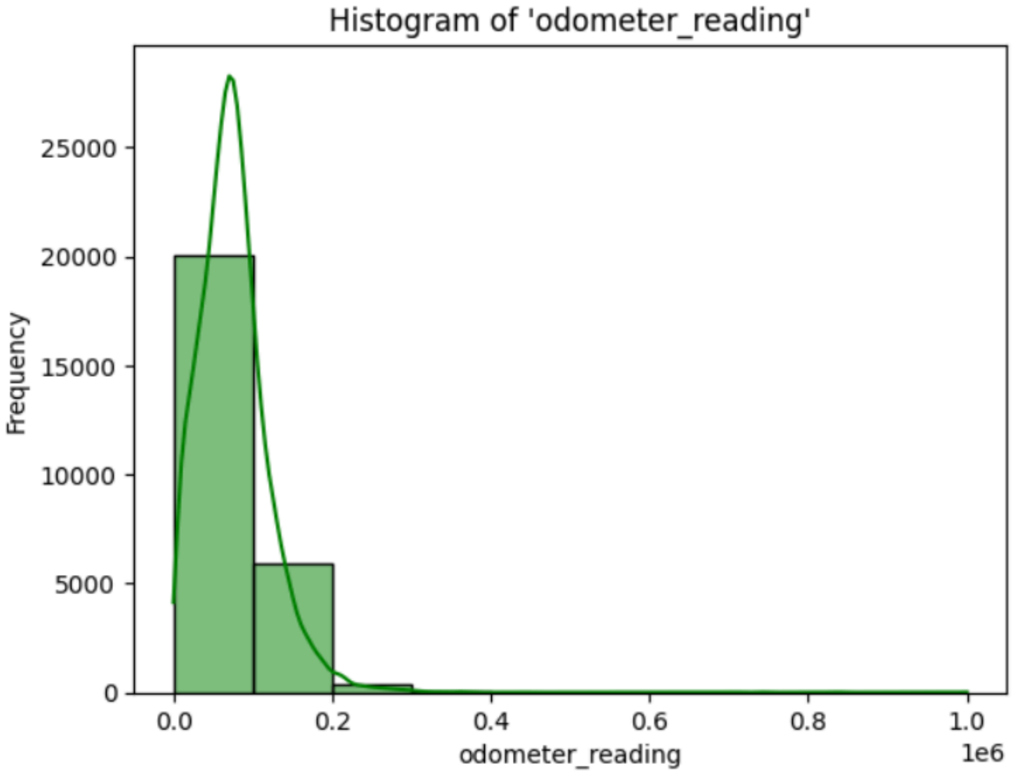
4.1 Univariate Analysis

A detailed univariate analysis was performed for key features such as:

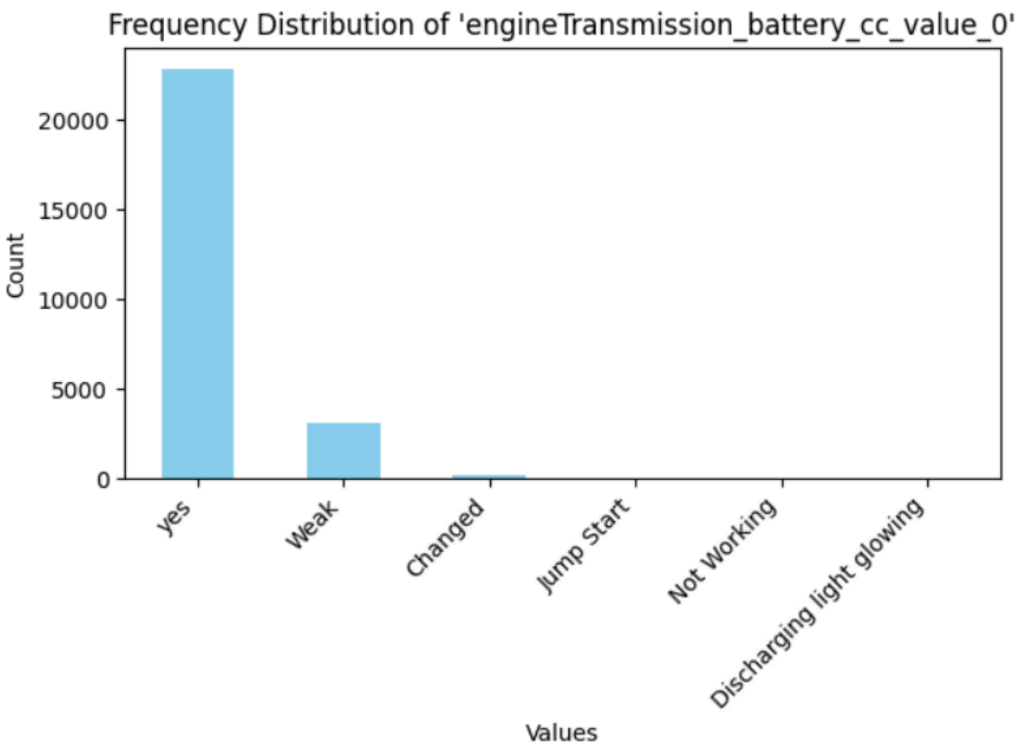
1. **odometer_reading**
    - Boxplots, histograms, value counts, and missing value distributions were visualized.



    -



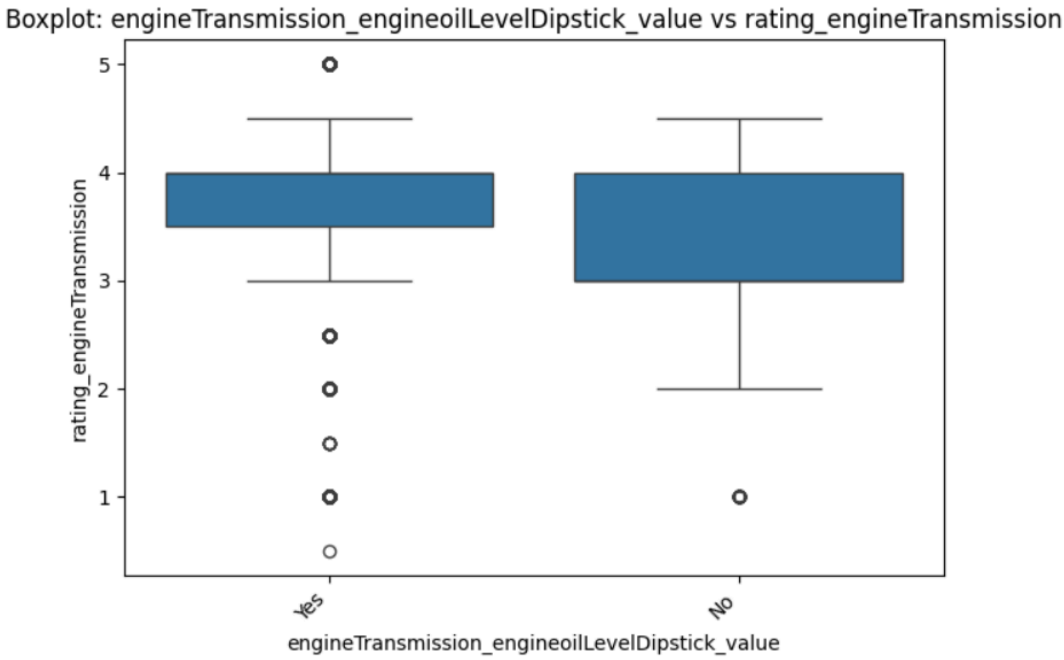
2. **engineTransmission_battery_cc_value_0**
   - Similar univariate metrics and visualizations were generated.

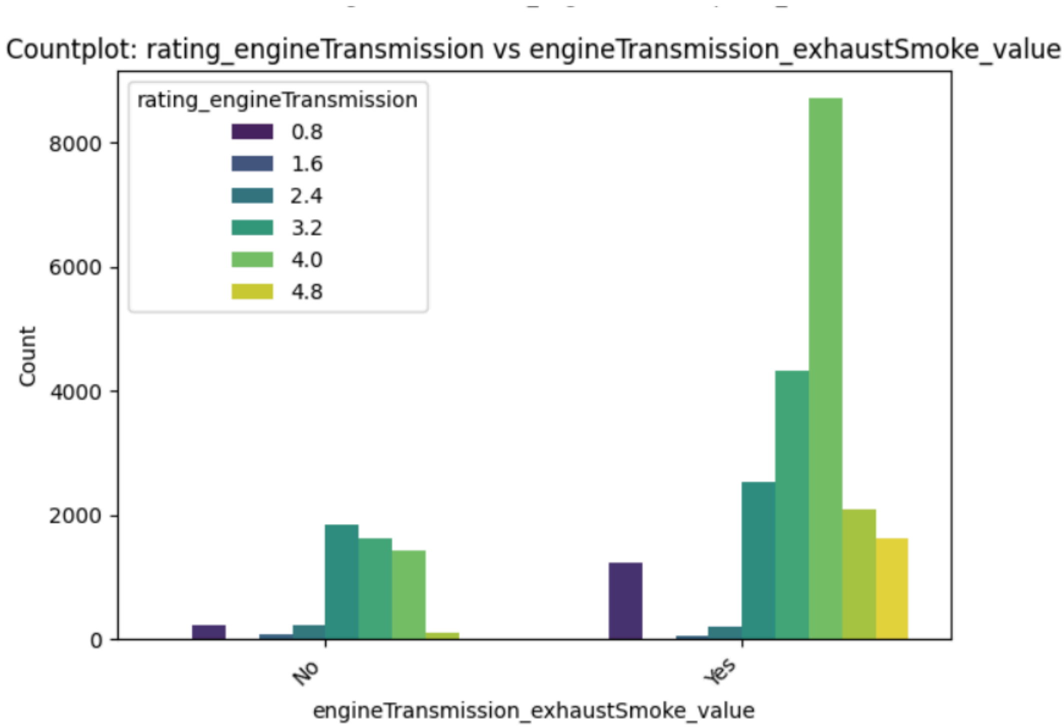


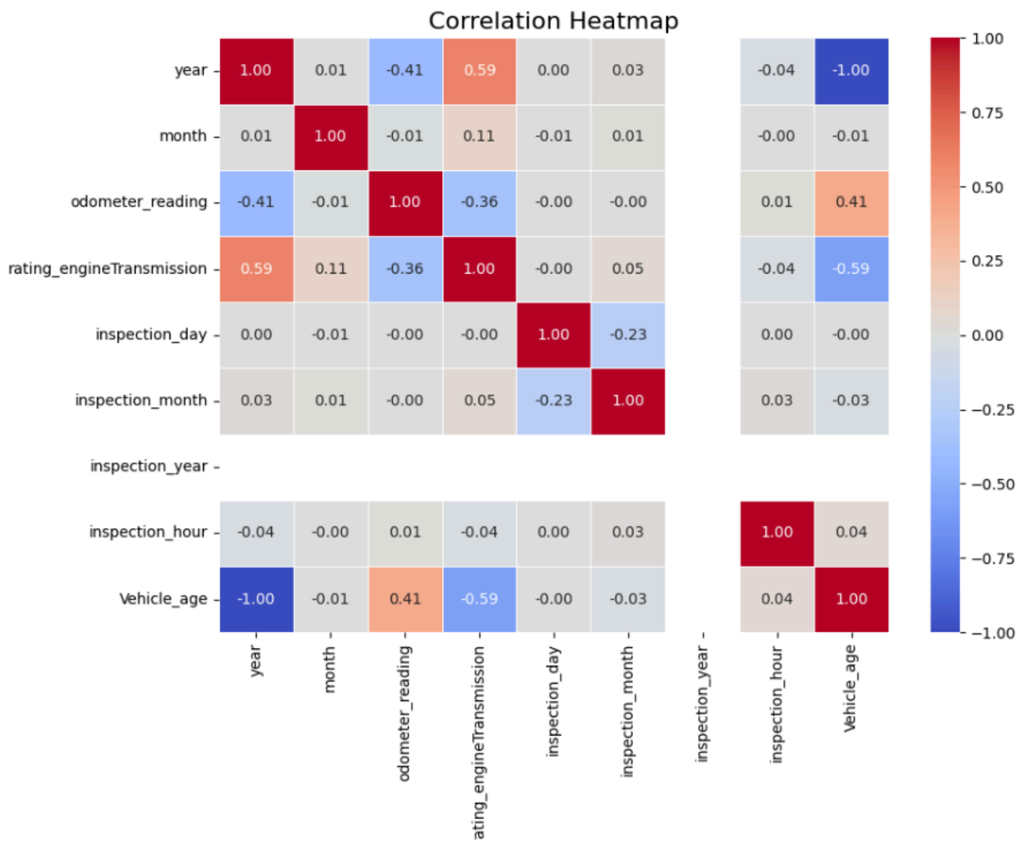
4.2 Bivariate Analysis

- Relationships between the dependent variable rating_engineTransmission and independent variables were explored:
    - **Categorical Variables**: Boxplots were used to examine distribution.
    - **Numerical Variables**: Scatterplots highlighted relationships with Vehicle_age and odometer_reading.

engineTransmission_coolant_value







4.3 Correlation Analysis

- Heatmaps and correlation matrices were used to analyze relationships among numerical variables.

4.4 Outlier Analysis

- Outliers in Vehicle_age and odometer_reading were identified using the IQR method.
- Boxplots were visualized to observe extreme values.

5. Outlier Treatment

- Outliers in Vehicle_age and odometer_reading were removed using the IQR method:
  - Upper and lower limits were calculated for both variables.
  - Observations outside these bounds were dropped.

6. Feature Engineering

- Unnecessary columns (appointmentId, inspectionStartTime, etc.) were dropped.
- Columns were rearranged to place the target variable rating_engineTransmission at the end.
- Categorical variables were one-hot encoded using pd.get_dummies() with drop_first=True.
- Features were scaled using MinMaxScaler.

7. Data Preparation

- Independent variables (X) and target variable (y) were separated:
  - Independent features were normalized.
  - Final dataset shape after preprocessing:
    - **X**: (number_of_samples, number_of_features)
    - **y**: (number_of_samples,)

8. Model Development

- A baseline **Ordinary Least Squares (OLS)** regression model was trained as a preliminary step to understand feature significance.

  Result of OLS

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     rating_engineTransmission   R-squared:                       0.446
Model:                                   OLS   Adj. R-squared:                  0.441
Method:                        Least Squares   F-statistic:                     92.78
Date:                       Wed, 18 Dec 2024   Prob (F-statistic):               0.00
Time:                               05:00:41   Log-Likelihood:                -25218.
No. Observations:                      26307   AIC:                         5.089e+04
Df Residuals:                          26080   BIC:                         5.275e+04
Df Model:                                226
Covariance Type:                   nonrobust
==================================================================================
```

Key Learnings and Next Steps

- The data contains a lot of columns . There is a linear relationship between the independent and dependent variables however, there is a problem of multicollinearity, thus, non-linear models like SVM and DTs are used for training.
- Some models were overfitting.
- Upon tuning hyperparameters, XG Boost trained well showing the highest Score as well as least difference between train and test data.
- This is choses as the final model.

Different Models and Score

Model	Train R2 Score	Test R2 Score
KNN	1	0.6639
SVM	0.5901	0.5718
Decision Trees	0.7428	0.6972
XGBoost	0.8334	0.734
XGBoost (max_depth=4)	0.7574	0.7254

5.XGBoost feature importance plot

```
Top Features by Importance:
                                   Feature  Importance
155  engineTransmission_engineSound_value_Yes    0.187309
235                        fuel_type_Petrol    0.066029
0                                      year    0.058731
24          engineTransmission_engineOil_Yes    0.055902
195        engineTransmission_clutch_value_Yes    0.042151
```

The final model which I used is XGBoost.

The model performance is shown below

XGB Train R2 Score: 0.7574038875760878

XGB Test R2 Score: 0.7254113930831192

The mean absolute percentage error between the final predictions and the actual values: 11.49%

Here there are around 1090, which are outliers in the predicted value.

```
threshold = 2 * comparison_df['Difference'].std()

outliers_df = comparison_df[comparison_df['Difference'] > threshold]
print(f"Number of Outliers: {len(outliers_df)}")
print(outliers_df)
```

```
Number of Outliers: 1090
       Actual  Predicted  Difference
8373      1.0   4.177699    3.177699
16022     0.5   3.592936    3.092936
2112      1.0   4.055227    3.055227
345       1.0   3.906495    2.906495
5548      1.0   3.772796    2.772796
...       ...        ...         ...
547       4.5   3.905965    0.594035
8025      3.0   3.593769    0.593769
25691     4.0   3.407547    0.592453
11403     3.0   3.592215    0.592215
11817     4.0   3.407814    0.592186

[1090 rows x 3 columns]
```

Next Steps:

If time would have been more, I would have dwelled more upon the data cleaning and feature Engineering part.
I think for reducing the number of columns for faster training and solving the issue of overfitting, combining multiple columns to single one to capture most of the variance in them can further reduce the model complexity and decrease overfitting.

The hyperparameter tuning of models like SVM , DT and Random Forests could further increase the model accuracy.