Auto Scout 24

Data Scientist - Skills Test

Is it possible to predict the product tier from the information given in the other columns?

```python
data_case_study[data_case_study['ctr'].str.count(r'\.') != 1]\
    [['search_views', 'detail_views', 'ctr']].reset_index(drop=True).tail()
```

	search_views	detail_views	ctr
100	2848.0	67.0	23.525.280.898.876.400
101	1075.0	31.0	2.883.720.930.232.550
102	1381.0	58.0	4.199.855.177.407.670
103	829.0	29.0	34.981.905.910.735.800
104	0.0	0.0	NaN

```python
modified_data[modified_data['first_registration_year']>2020]
```

	article_id	product_tier	make_name	price	first_zip_digit	first_registration_year	created_date	deleted_date
36302	358877131	Basic	Opel	9250	7	2106	24.09.18	26.09.18

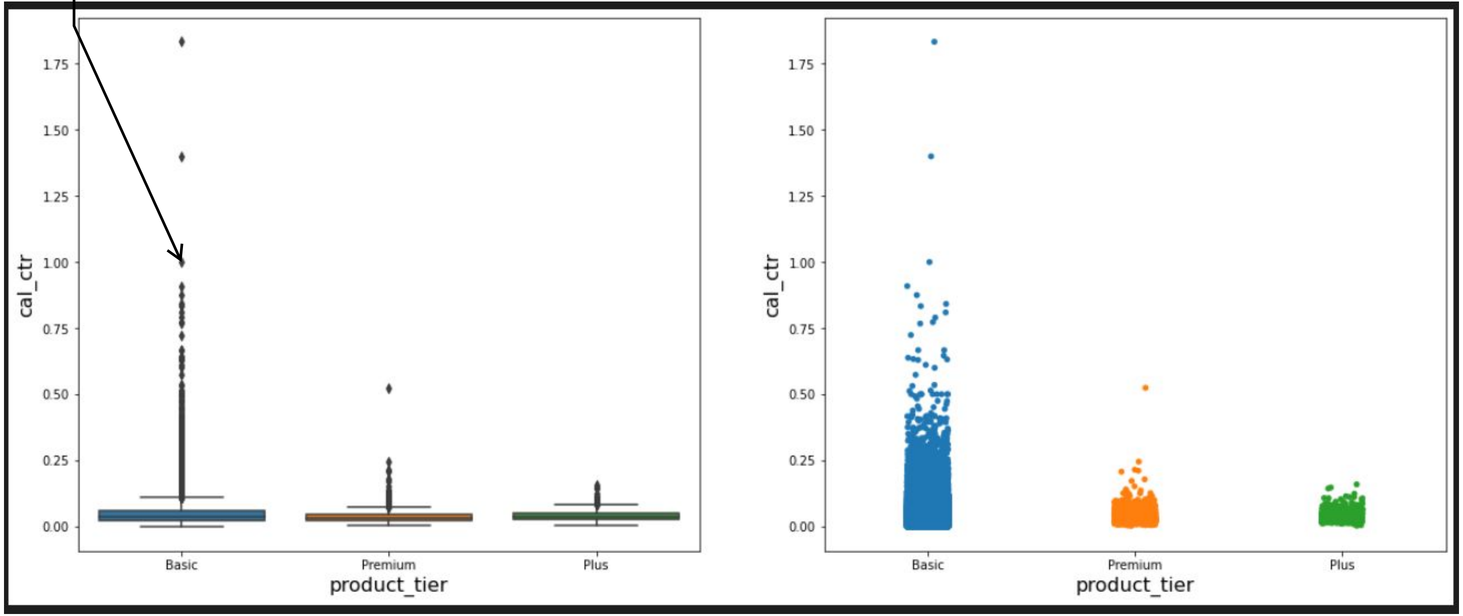
```python
modified_data.loc[modified_data['article_id'] == 358877131, 'first_registration_year'] = \
    modified_data.groupby('make_name', as_index=False)['first_registration_year'].mean()\
    .loc[modified_data['make_name'] == 'Opel', 'first_registration_year'].iloc[0].astype(int)
```

```python
modified_data[modified_data['article_id'] == 358877131]
```

	article_id	product_tier	make_name	price	first_zip_digit	first_registration_year	created_date	deleted_date
36302	358877131	Basic	Opel	9250	7	1991	24.09.18	26.09.18

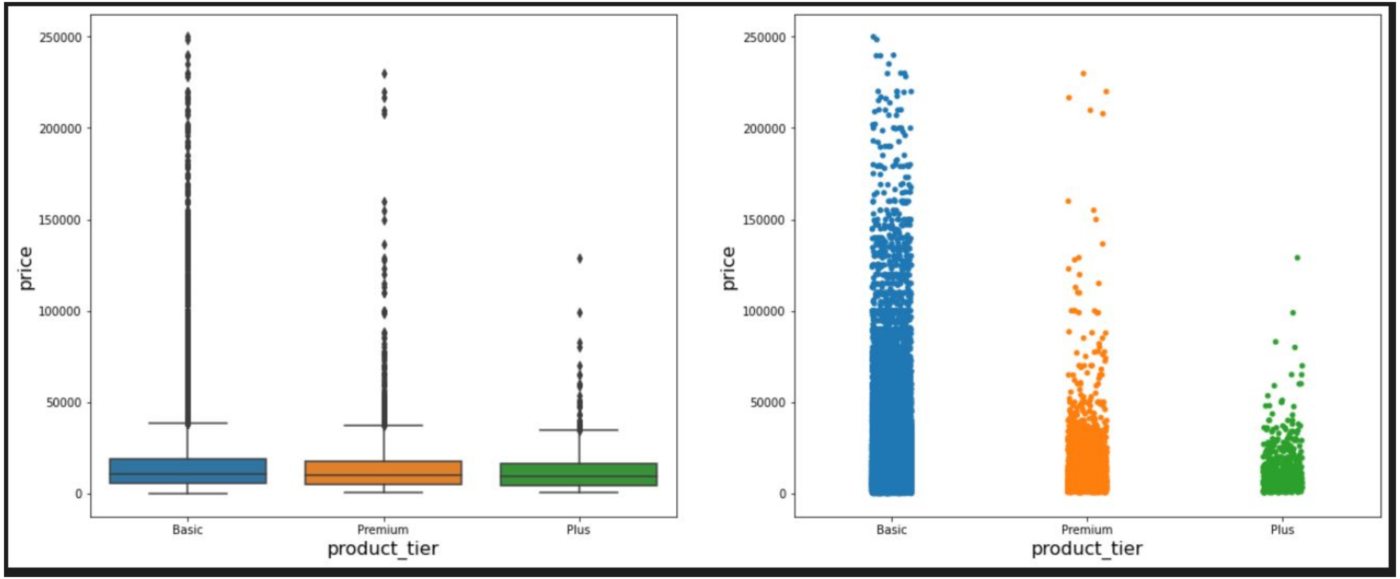
product_tier = basic, has the higher CTR

Product_Tier Vs CTR



From Q1, Q2 and Q3 (ctr) of the box plot for the Basic category we can observe that it subsumes the ctr from other categories and therefore the feature ctr independently cannot differentiate the articles in different product_tiers.

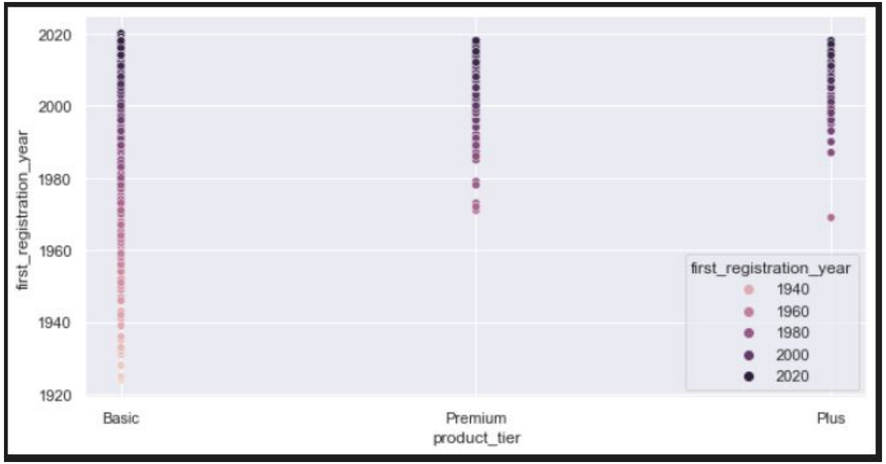
Product_Tier Vs Price



- From the plot below we can directly reject a common perception that articles in the product_tier Basic need not necessarily be cheaper compared to other categories.

- The vertical spread of the strip plot, particularly along the product_tier Basic shows that price of the article is not the ONLY factor or reason for the low ctr of other categories.

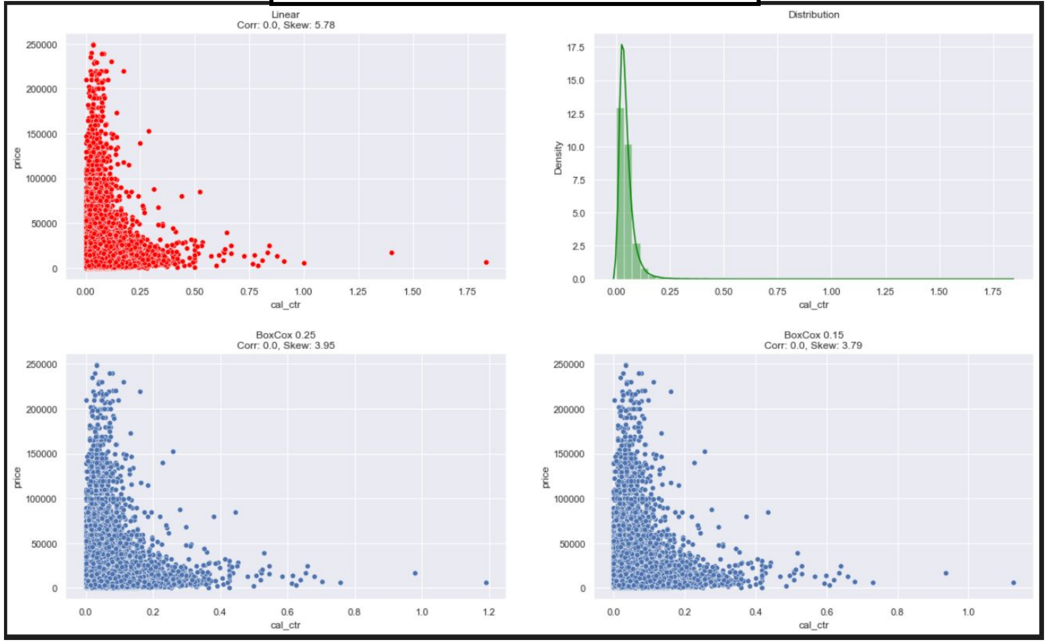
Product_Tier Vs Registration_year



- The scatter plot below indicates that the category Basic has a very wide collection of articles w.r.t the first_registration_year.

- This variance for the Basic category attracts more visitors yielding higher CTR and understandbly the opposite for other categories.

Price Vs CTR



- Highly skewed but distribution indicates price does not influence the CTR as we can observe cluster of data points on along both axes.

- Although the skewness decreases but has no effect on the correlation (preserved). So price and ctr have no dependency between them.

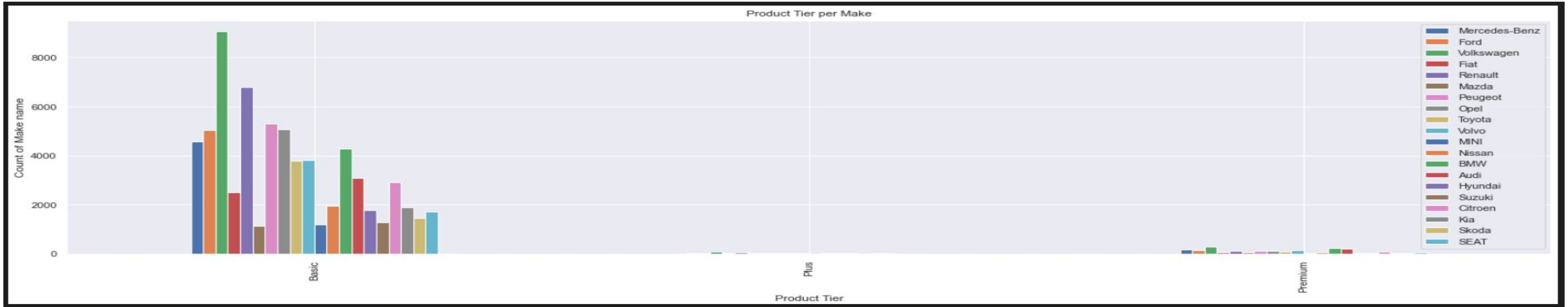
ANOVA - Product_Tier Vs Price

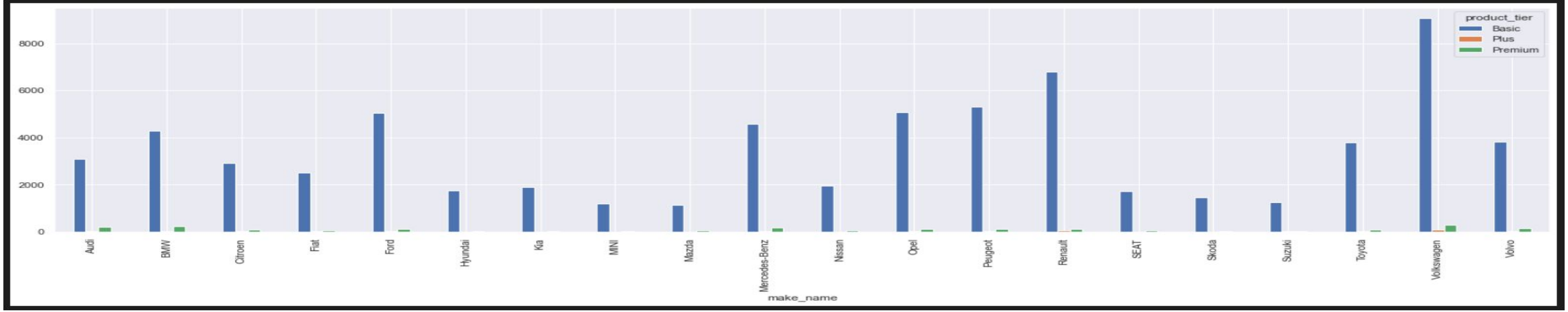
```python
anova_test_data = modified_data.copy()
anova_test_data = anova_test_data.groupby('product_tier').\
                  head(Counter(modified_data['product_tier'])['Plus'])
anova_test_data.product_tier = anova_test_data.product_tier.astype('category').cat.codes

fvalue, pvalue = stats.f_oneway(anova_test_data['product_tier'], anova_test_data['price'])
print(f'Computed F statistic of the test = {fvalue}')
print(f'Associated p-value from the F distribution = {pvalue}')
```

Python

```
Computed F statistic of the test = 1346.0679535706752
Associated p-value from the F distribution = 3.775289854824415e-249
```

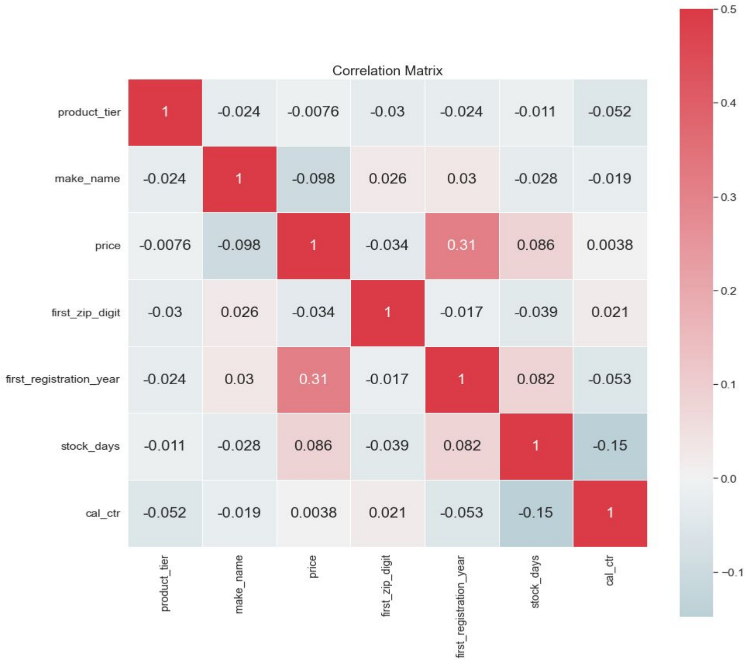
- H0 - Subgroups of product_tier has an effect on the price. H1 - Subgroups of product_tier does not have an effect on the price.
- As there are more than one groups in the product_tier and price being a continuous variable we sought to apply ANOVA test to verify if there is any deviation between the groups w.r.t price.
- Due to unbalanced (unequal sample size for each group) data, we will perform one-way ANOVA with balanced design (equal sample size for each group).
- For the given dataset the ANOVA analysis is significant (p value < 0.05) therefore we reject H0 and accept H1. We conclude that product_tier does not have an effect on the price.





- The first bar graph shows the count of product_tier per make from which it is clear that articles from Basic product_tier cover and dominate all the make_name available.

- From the second bar graph we can comprehend the imbalance in the data in terms of different make_name per product_tier. Such an imbalance already indicates the bias towards one class while performing classification.



- All the selected features against the product_tier, individually show minimal correlation (both positive and negative) with each other.
- One crucial observation we can see from the matrix is the correlation between the price and the first_year_registration. This also showcases the obvious fact that articles with older registration have lower price and vice-versa.
- From the coefficients we can see that make_name is most negatively correlated with price and hence has an influence on it.

Model training and Hyper-parameter optimization

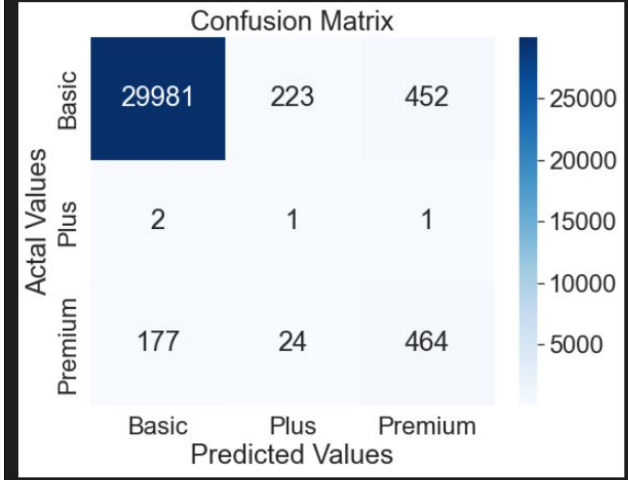
- Features - ['make_name', 'first_registration_year', 'price', 'search_views', 'detail_views', 'stock_days', 'cal_ctr']
- Target - ['product_tier']
- Train:Test ratio - 0.6 : 0.4
- Grid search parameters

```
param_grid = {
    'objective':['multiclass'],
    'num_class':[3],
    'n_estimators': list(range(200, 800, 200)),
    'boosting_type': ['gbdt', 'rf'],
    'num_leaves': list(range(10, 100, 20)),
    'learning_rate': [0.01, 0.1],
    'subsample_for_bin': [20000, 30000],
    'min_child_samples': [20, 50],
    'colsample_bytree': [0.6, 0.8],
    "max_depth": [5, 10],
    "metric":['softmax']
}
```

Best parameters

```
lgbm_tuned = LGBMClassifier(boosting_type = 'gbdt',
                            n_estimators = 400,
                            num_class = 3,
                            colsample_bytree = 0.6,
                            learning_rate = 0.1,
                            max_depth = 5,
                            metric = 'softmax',
                            min_child_samples = 50,
                            num_leaves = 10,
                            subsample_for_bin = 30000)
```

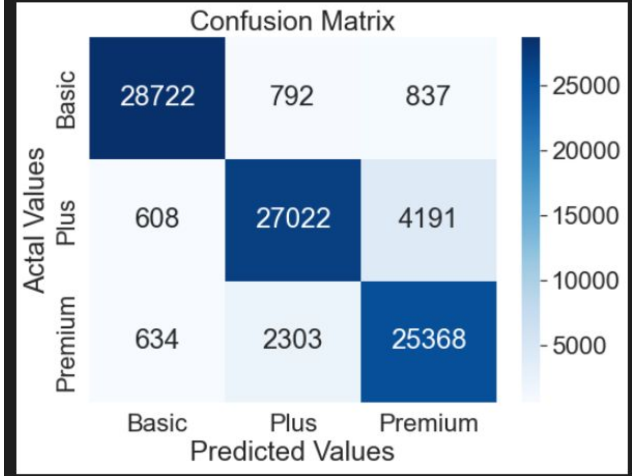
```
Accuracy of the model = 97.2
```



	precision	recall	f1-score	support
Basic	0.99	0.98	0.99	30656
Plus	0.00	0.25	0.01	4
Premium	0.51	0.70	0.59	665
accuracy	0.97	0.97	0.97	0
macro avg	0.50	0.64	0.53	31325
weighted avg	0.98	0.97	0.98	31325

- Use SMOTE method to oversample the minority class and balance the dataset.
- Develop 2 models i.e model-A as a binary classifier that initially splits the prediction into the product_tier Basic and Other (Plus and Premium). Then model-B as another binary classifier for the other group.

```
Accuracy of the model = 89.6
```

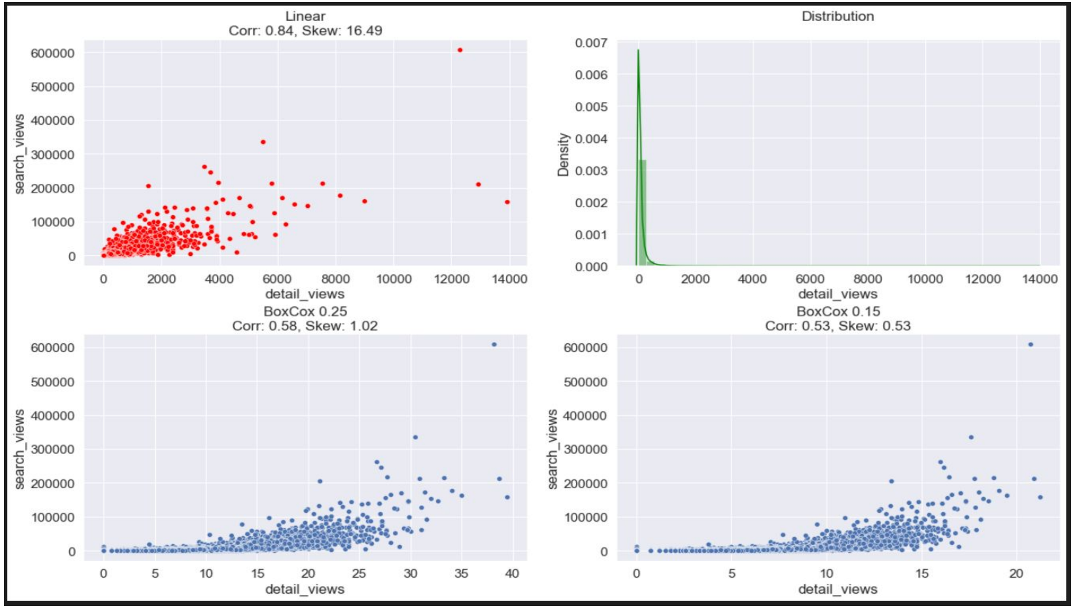


	precision	recall	f1-score	support
Basic	0.96	0.95	0.95	30351
Plus	0.90	0.85	0.87	31821
Premium	0.83	0.90	0.86	28305
accuracy	0.90	0.90	0.90	0
macro avg	0.90	0.90	0.90	90477
weighted avg	0.90	0.90	0.90	90477

Is it possible to predict detail views from the information given in the other columns?

Preprocessing and Cleaning

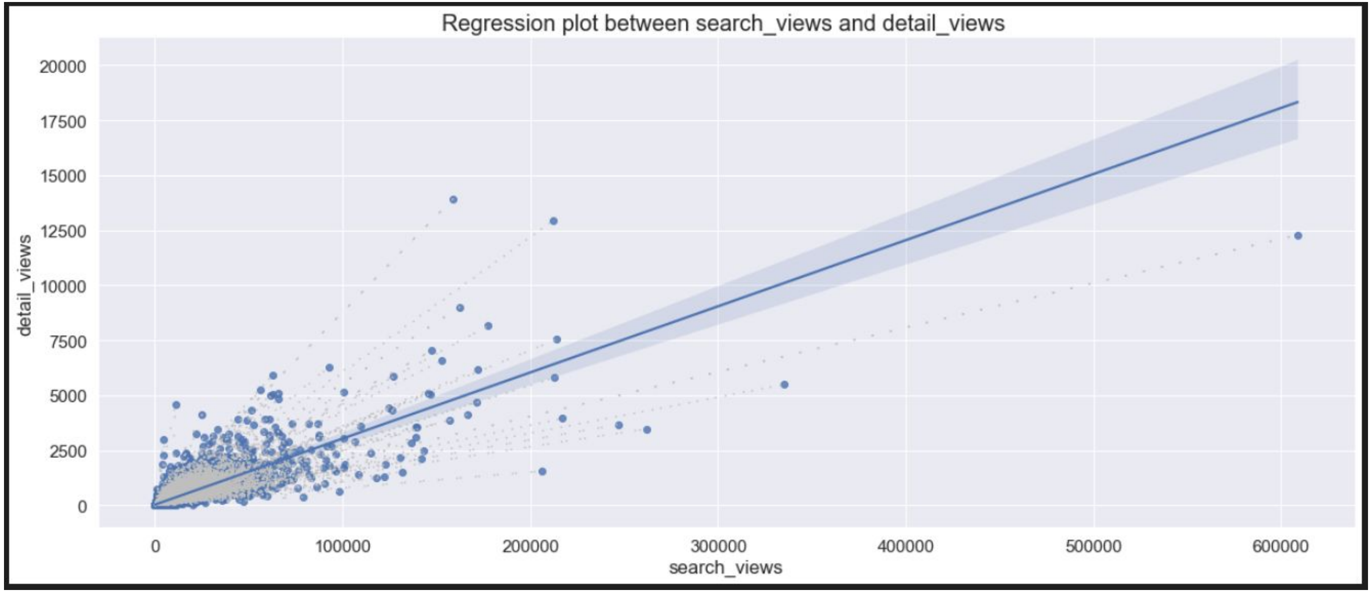
- We can witness some values along the column stock days are negative (i.e deleted_date > created_date) and therefore they are discarded. If we can determine the reason then we could tweak the data and minimize the loss of significant amount of samples.
- Additionally we can notice that some articles have search_views = 0 and stock_days = 0 which means the article was posted and taken down immediately. These data points are trivial and hence can be eliminated.
- For the purpose of analysis and explainability we incremented the stock_days by 1 from the existing values so as to include the contribution of the articles having stock_days = 0 (i.e created_date == deleted_date) and search_views > 0.



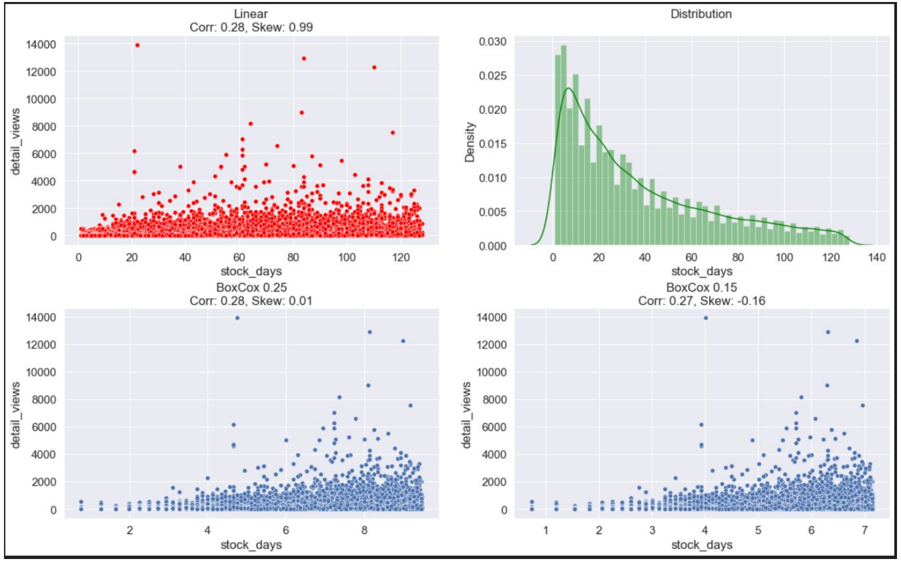
- A very high correlation is observed between the search_views and detail_views. Therefore we can postulate as24 is providing significantly good suggestions based on the filters selected by the visitors.

- In this scenario there is higher probability of a search_view being converted to detail_view eventually boosting the ctr. A more sophisticated and organic approach would be to include the position of product as a correction factor for the CTR calculation.

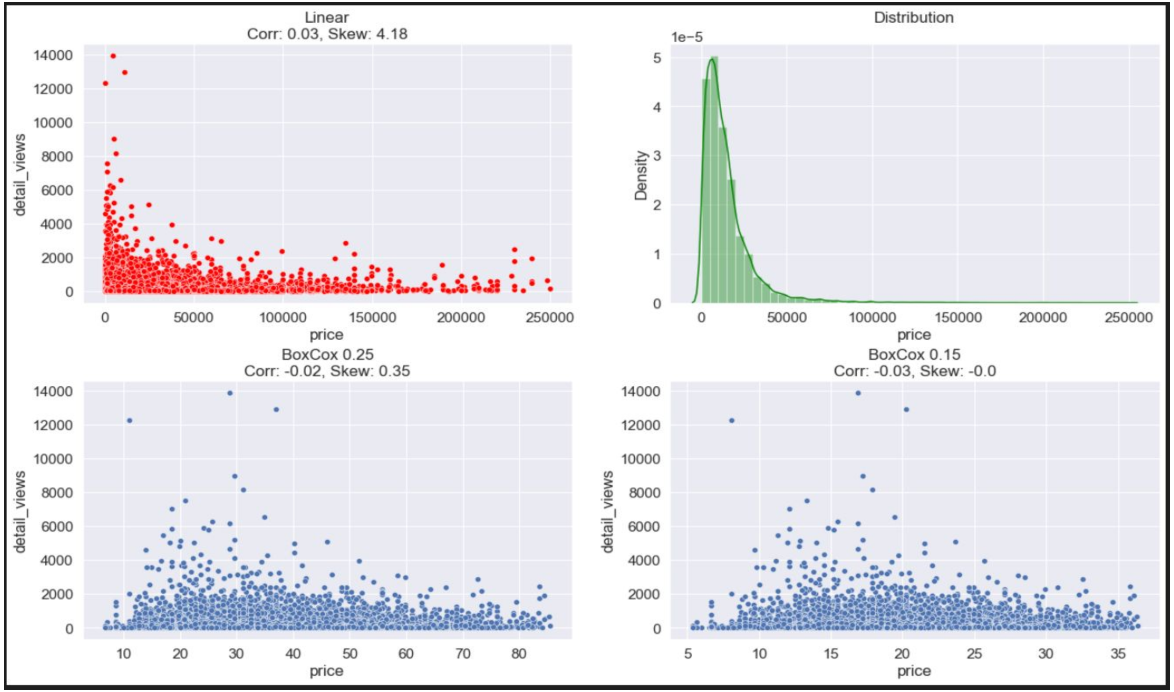
- Correlation changes with transformation (Positive aspect)



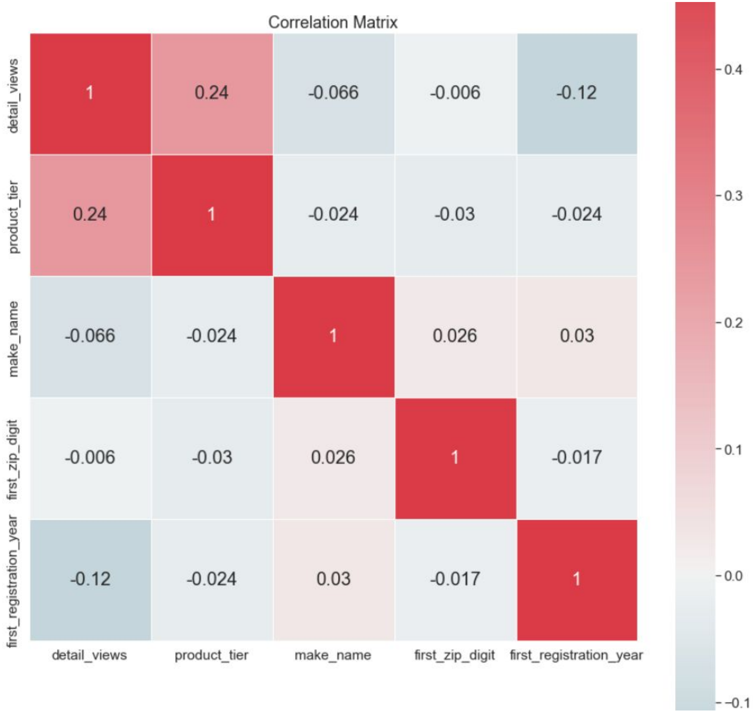
From the regression plot and the confidence region between the search_views and detail_views we can see that the uncertainty around the best fit line is very narrow indicating the interdependence between the variables.



- From the scatter plot we can distinctly see some significant red spots above the 4000 mark as the stock_days increases over 40. This indicates that the articles listed for more than a month are still relevant to the visitors and it will be interesting to know the conversion of these articles to being purchased (or leased).
- The distribution of detail_views between stock_days > 40 and stock_day < 60 will act as a significant feature to predict the detail_views of the articles. In contrast the density of detail_views tails out in the end and it can aid it distinguishing the group of articles at a node in the decision tree.



- Very less correlation between price and detail_views shows that price is not a limiting factor for the visitors to choose their desired product.
- The distribution from the pdf plot for the price range 0 to 50000 shows the diversity of visitors on as24.



- Very high negative correlation between first_year_registration adn detail_views shows the orientation of the visitors. The visitors are more likely looking for recently registered cars in spite of higher price.
- The product_tier is playing a vital role in influencing the detail_view clicks per article, but this assumption cannot be accepted to the fullest because of imbalanced data samples in the product_tier class.
- There should have been non-trivial level of correlation between the make_name and detail_views but as we noticed in the above analysis the product_tier Basic covers most of the make_names when compared to Plus and Premium class groups.
- Using ctr will lead to data leakage because of the its formula. Therefore we will not include it in the feature set.

Model training and Hyper-parameter optimization

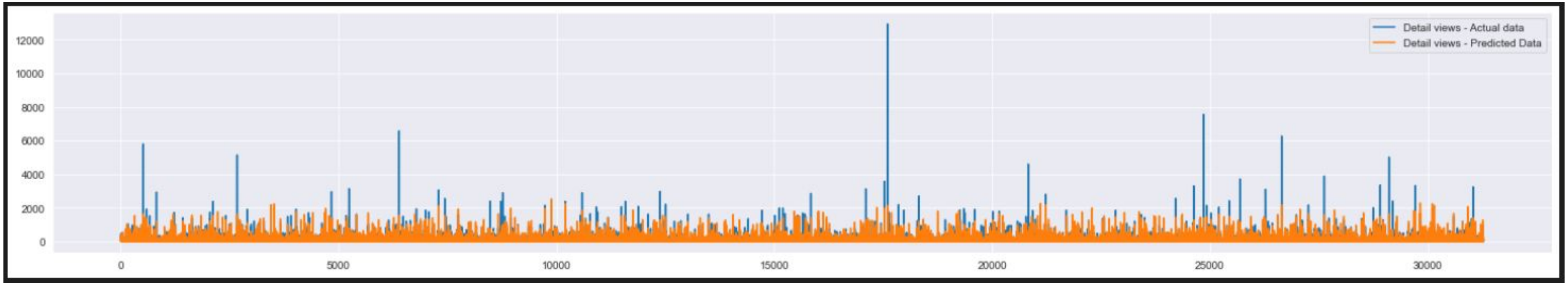
- Features - ['product_tier', 'make_name', 'price', 'first_registration_year', 'created_date', 'search_views', 'stock_days']
- Additional calendar features - The creation_date of the advertisement can at times be captured under certain period or season of the calendar and lead to increased detail_views.
  - created_month_of_year - month from the date (1-12)
  - created_week_of_year - week in the year (1-52)
  - created_day_of_week - day no. of the week (1-7)
- Target - ['detail_views']
- Train:Test ratio - 0.6 : 0.4
- Grid search parameters

```
param_grid = {
    'objective':["regression"],
    'n_estimators': list(range(200, 1000, 200)),
    'boosting_type': ['gbdt', 'dart'],
    'num_leaves': list(range(10, 50, 10)),
    'learning_rate': [0.01, 0.1],
    'min_child_samples': [20, 50],
    'colsample_bytree': [0.6, 0.8],
    "max_depth": [5, 10],
    "metric":["mape"]
}
```

Best parameters

```
lightgbm.LGBMRegressor(
                    n_estimators=400,
                    objective="regression",
                    num_leaves=30,
                    max_depth=5,
                    min_child_samples=50,
                    learning_rate=0.1,
                    colsample_bytree=0.8,
                    verbosity=-1,
                    extra_trees=True,
                    metric="mape"
                    )
```

```
Results of regression model to predict detail views:
MSE: 18971.012977880066
RMSE: 137.73530040581488
R-Squared: 0.5897513505247178
```



What could be done

- We can observe that the model performs appreciably better at detail_views less than 4000 but the errors increase for the anomalies where the detail_views are greater than 5000.
- Understandably the model has overfitted which is also due to less variance in provided data. This could be avoided by inducing artificial samples to model the anomalies.
- Replace the anomalies with the average values over a fixed window size and over a group of articles.

THANK YOU

ANY QUESTIONS