Supervised Machine Learning: Regression

Honors Peer-graded Assignment

Objectives

- The main objective of the analysis specifies whether your model will be focused on prediction or interpretation.

Trying to understand the factors that influence the sale price of mobile phones.

1. Specifically, which factors drive mobile phone prices up? (**Interpretation power**)
2. how accurately can you predict the sale price based on the mobile phone's features? (**Prediction power**)

About Dataset

- Brief description of the data set you chose and a summary of its attributes.

1. Phone Name is the name of the phone that is extracted.
2. Rating is what it gets out of 5 stars.
3. The number of Ratings is the Total number of people those rates this product.
4. Ram size
5. Rom is the Storage that the product has.
6. Front and Rare Camera in the Mega Pixels.
7. Battery size and processor of different types.
8. Lastly, the price of the phone is in Indian Rupees.

All this data is gathered from the Flipkart Indian Online shopping website.Source: Data

Exploratory Data Analysis

- Brief summary of data exploration and actions are taken for data cleaning and feature engineering.

Here are the steps taken to clean the data:

1. Explored the current state of the `Mobile Phone data for 2032` and change the Column names for better readability
2. Transformed data types for each column to appropriate ones
3. Reformatted column values or fields for further analysis using regular expressions
4. Handled missing values using fillna() method using backfill methods
5. Dropped unwanted columns that have more count of non-null values
6. Checked Outliers - using boxplots and analyzed the impact on the dataset
7. Exploratory Data Analysis using Visualization:
    a. Used `describe()` to reveal statistical information about the numeric attributes

    b. Used `value_counts()` function to reveal some information about our categorical (object) attributes
    c. Graphical representations to understand relation between target (Annual salary) and independent variables using pairplot()
8. Established a correlation between the Price and other predictor variables
9. Test Assumptions for Linear Regression:
    a. Linearity Assumption - Tested this assumption with some scatter plots and regression lines
    b. Homoscedasticity: a situation in which the error term / the "noise" in the relationship between the independent variables and the target variable(Price) is the same across all values of the independent variable. Error variance across the true line is dispersed somewhat not uniformly, and the *homoscedasticity* is more likely not met.
    c. Normality: checked, 'price', variable to be not normally distributed. Hence, performed Log and boxcox transformations to make the 'price' distribution into a more symmetrical bell curve
    d. Multicollinearity = linear regression requires independent variables to have little or no similar features. Check with the heatmap and we can solve the *multicollinearity* issue by using regularization methods
10. Performed feature engineering methods to come up with a good set of features to train on.
    - Feature transformation:
        - reduced categorical values to minimum categories to perform one-hot coding
        - one hot encoding categorical column values creating dummy_data
    - Feature selection
        - selected attributes that best explain the relationship of the independent variables with respect to the target variable, **Price**
11. Saved and exported the cleaned_phone_data for future use.

Regression Models & Results

- Summary of training at least three linear regression models which should be variations that cover using a simple linear regression as a baseline, adding polynomial effects, and using a regularization regression. Preferably, all use the same training and test splits, or the same cross-validation method.

Divided the dataset into a 30% test set and a 70% train set to understand the model performance on the new data.

1. Linear regression

Applied linear regression model to fit the train set and predict values on the test set
Model evaluation Results :

Model	Evaluation Metric	Scores
Linear regression	R-squared on training data	0.844

Linear regression	R-squared on testing data	0.778
Linear regression	R2_score	0.778

The closer R squared to 1, the better the fit of the model.

2. **Linear regression with Scaling:**

Scaled data suing StandardScaler methods and applied linear regression model to fit the train set and predict values on the test set
Model evaluation Results :

Model	Evaluation Metric	Scores
Linear regression + scaling	R-squared on training data	0.844
Linear regression + scaling	R-squared on testing data	-1.0531760673887472e+21
Linear regression + scaling	R2_score	-1.0531760673887472e+21
Linear regression + scaling	MSE	1.868269029710405e+29
Linear regression + scaling	RMSE	432234777604764.8

If the R squared is negative, it suggests overfitting, when a statistical model fits exactly against its training data.

The closer R squared to 1, the better the fit of the model, and the better we did in regards to explaining the overall variance. for example, We can always add more features. Even if those features don't have any predictive power, they will never bring down the R-squared score. If you were to add on another feature and it didn't have any predictive power, we can just set that coefficient to 0 and it wouldn't bring down our R-squared.

3. **Polynomial + linear regression**

Polynomial transform is a simple way to increase the complexity of the model, but we must be mindful of overfilling. Below, we will perform a second-degree (degree=2) polynomial transformation.

Model	Evaluation Metric	Scores
Polynomial Regression	R-squared on training data	0.9934023971683771
Polynomial Regression	R-squared on testing data	-1.085068443290085e+18
Polynomial Regression	R2_score	-1.085068443290085e+18

We see the model has a negative $R2$ on the test data set, this is a sign of overfitting when a statistical model fits exactly against its training data.

4. Polynomial regression + regularization methods using GridSearchCV and pipeline

a. Linear regression with polynomial features

Model	Cross-validation Metric	Scores
Polynomial Regression	R-squared on testing data	0.8955756261874462
Best model	Pipeline(steps=[('polynomial', PolynomialFeatures(degree=1, include_bias=False)), ('model', LinearRegression(normalize=False))])	

b. Linear regression with polynomial features with StandardScaler()

Model	Cross-validation Metric	Scores
Polynomial + scaling	R-squared on testing data	0.8955756261874462
Polynomial + scaling	best_score_	-9.405368643131627e+23
Polynomial + scaling	best_params_	{'model__normalize': False, 'polynomial__degree': 2}
Best model	Pipeline(steps=[('polynomial', PolynomialFeatures(degree=1, include_bias=False)), ('model', LinearRegression(normalize=False))])	
Best model	MSE	248.36880264278986
Best model	RMSE	61687.0621262131

Both the MAE and RMSE can range from 0 to ∞. They are negatively-oriented scores: Lower values are better

c. Ridge Regression

Model	Cross-validation Metric	Scores
Ridge Regression	R_score on testing data	0.9996522591957278
Ridge Regression	R-squared on testing data	0.9996522591957278

Ridge Regression	best_score_	0.7710198620150134
Ridge Regression	best_params_	{'model__alpha': 10, 'polynomial__degree': 1}
Best model	Pipeline(steps=[('polynomial', PolynomialFeatures(degree=1, include_bias=False)), ('ss', StandardScaler()), ('model', Ridge(alpha=10))])	

d. Ridge Regression without polynomial features

Model	Evaluation Metric	Scores
Ridge Regression	R-squared on training data	0.8446516073750969
Ridge Regression	R-squared on testing data	0.7814404739036939
Ridge Regression	R2_score	0.7814404739036939

without using `SatndardScaler()`, `PolynomialFeatures()` give us low r-squared and R2_score values. hence, not recommended.

e. Lasso Regression:

Model	Cross-validation Metric	Scores
Lasso Regression	R_score on testing data	0.7902183470329915
Lasso Regression	R-squared on testing data	0.7902183470329915
Lasso Regression	best_score_	0.7710198620150134
Lasso Regression	best_params_	{'model__alpha': 10, 'polynomial__degree': 1}
Best model	Pipeline(steps=[('polynomial', PolynomialFeatures(degree=1, include_bias=False)), ('ss', StandardScaler()), ('model', Lasso(alpha=10, tol=0.2))])	

Lasso regression doesn't do a great job. Model performance dropped as R-squared and R2_score dropped from 0.99 to 0.79

f. ElasticNet Regression

Model	Cross-validation Metric	Scores
ElasticNet	R_score on testing data	-6.102162266503638
ElasticNet	R-squared on testing data	-6.102162266503638
ElasticNet	best_score_	0.8414734265997355
ElasticNet	best_params_	{'model__alpha': 10, 'model__l1_ratio': 0.9, 'polynomial__degree': 2}
Best model	Pipeline(steps=[('polynomial', PolynomialFeatures(include_bias=False)), ('ss', StandardScaler()), ('model', ElasticNet(alpha=10, l1_ratio=0.9))])	

Best Model and Insights

- A paragraph explaining which of your regressions you recommend as a final model that best fits your needs regarding accuracy and explainability.

R-squared is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable in a regression model. An R2 of 1.0 indicates that the data perfectly fit the linear model. Any R2 value less than 1.0 indicates that at least some variability in the data cannot be accounted for by the model (e.g., an R2 of 0.5 indicates that 50% of the variability in the outcome data cannot be explained by the model)

- R-squared for ElasticNet is the lowest = -6.102 than
- R-squared for Lasso Regression = 0.79
- Here we can see that the Ridge regression model has the highest R-squared values
  - R-squared = 0.999 using GridSearchCV() **AND**
  - best model = Pipeline(steps=[('polynomial', PolynomialFeatures(degree=1, include_bias=False)), ('ss', StandardScaler()), ('model', Ridge(alpha=10))])

- Summary Key Findings and Insights, which walks your reader through the main drivers of your model and insights from your data derived from your linear regression model.

The value R2 quantifies goodness of fit and a higher R-squared indicates the model is a good fit, Hence, the Best model is Ridge Regression to fit the overall data with perfect bias-variance trade-off.

which factors drive mobile phone prices up?

Below are the features:

ROM_GB	= 0.745777
RAM_GB	0.718304
Battery_mAh	0.118468
Rating	0.001147
Number_of_Ratings	-0.197043

how accurately can you predict the sale price based on the mobile phone's features?

R-score provide the confidence level = 0.99, 99.9% we can predict model price values of phones

Future steps

- Suggestions for the next steps in analyzing this data, which include revisiting this model and adding specific data features to achieve a better explanation or a better prediction.

We would explore Principal Component Analysis(PCA) to reduce the dimensionality of our data. We will do so by creating a Pipeline object first, then applying standard scaling and performing PCA, and then applying Elastic Net Regularization. We can reduce the complexity and noise of the data, and highlight the most important features and relationships.

Adding more features related to mobile phone users, demographic, and geographical attributes. Models with too many features will have less model prediction accuracy with high variance and low bias. While a model with few features or less number of features will have a high bias leading to reduce model prediction accuracy on new/test data. This dataset has limited features that represent the prediction of phone prices. Hence, we have better prediction scores but less confidence in the general price of phones.