CS 5805 – Machine Learning 1
Final Term Project

AT&T Internet Speeds and Prices Analysis

Presented by
Mihir Rathod

Introduction

- The dataset is about the AT&T – Internet Speeds, Prices, Geographical location, and Socioeconomic factors across multiple regions in the United States.
- It explores the relationship between internet services and regional characteristic for analysis.
- It includes data on Download/Upload Speeds, Pricing of Internet services, and Types of Internet Packages offered.
- The dataset also contains Geographical Information such as customer addresses, and details like People per Square Mile and Median Household Income.

Overview of Dataset - Characteristics

- The Dataset originally consists of 432,303 observations and 26 features (11 categorical and 15 numerical).
- Numerical Features: It includes download speeds, upload speeds, latitude, longitude, number of providers, etc.
- Categorical Features: It includes state, technology, major city, package type, etc.
- Target Variables –
    - Phase I & II (Regression): 'speed_up' – Upload speeds.
    - Phase III (Classification): 'package' – Type of internet package with 4 classes.

Objectives

Phase I: Perform feature engineering and exploratory data analysis (EDA) to understand the dataset, implement different models for feature selection, and choose the best feature selection model for further implementation.

Phase II: Implement regression on a continuous feature ('speed_up') to predict and analyze the upload speed using other predictors in the dataset.

Phase III: Implement seven different classifiers on the categorical target variable ('package'), analyze the results, and pick the best classifier based on their performances and other metrics.

Phase IV: Implement clustering using two clustering models, perform association rule mining using Apriori algorithm model, and analyze the results.

PHASE I:
FEATURE ENGINEERING & EDA

PHASE I - Head of dataset

Head of the dataset:

	address_full	incorporated_place
0	2406 Country Club Ave NW, Huntsville AL 35816	Huntsville city
1	1902 Oglesby Dr NW, Huntsville AL 35816	Huntsville city
2	2312 Cardinal Ave NW, Huntsville AL 35816	Huntsville city
3	1903 Oglesby Dr NW, Huntsville AL 35816	Huntsville city
4	1905 Canterbury Cir NW, Huntsville AL 35816	Huntsville city

	major_city	state	lat	lon	block_group	collection_datetime	provider
0	huntsville	AL	34.745	-86.607	10890007022	1650310200	AT&T
1	huntsville	AL	34.748	-86.607	10890007022	1650310229	AT&T
2	huntsville	AL	34.747	-86.606	10890007022	1650310203	AT&T
3	huntsville	AL	34.748	-86.608	10890007022	1650310195	AT&T
4	huntsville	AL	34.749	-86.605	10890007022	1650310196	AT&T

	speed_down	speed_up	speed_unit	price	technology	package
0	0.768	0.384	Mbps	55.000	Not Fiber	Internet Basic 768kbps
1	5.000	1.000	Mbps	55.000	Not Fiber	Internet Basic 5
2	0.768	0.384	Mbps	55.000	Not Fiber	Internet Basic 768kbps
3	5.000	1.000	Mbps	55.000	Not Fiber	Internet Basic 5
4	300.000	300.000	Mbps	55.000	Fiber	AT&T FIBER—INTERNET 300

	fastest_speed_down	fastest_speed_price
0	0.768	55.000
1	5.000	55.000
2	0.768	55.000
3	5.000	55.000
4	5,000.000	180.000

	fn
0	../data/intermediary/isp/att/huntsville/010890007022.geo...
1	../data/intermediary/isp/att/huntsville/010890007022.geo...
2	../data/intermediary/isp/att/huntsville/010890007022.geo...
3	../data/intermediary/isp/att/huntsville/010890007022.geo...
4	../data/intermediary/isp/att/huntsville/010890007022.geo...

	redlining_grade	race_perc_non_white	income_lmi	ppl_per_sq_mile
0	NaN	0.475	0.382	512.090
1	NaN	0.475	0.382	512.090
2	NaN	0.475	0.382	512.090
3	NaN	0.475	0.382	512.090
4	NaN	0.475	0.382	512.090

	n_providers	income_dollars_below_median	internet_perc_broadband
0	4.000	35,091.000	0.528
1	4.000	35,091.000	0.528
2	4.000	35,091.000	0.528
3	4.000	35,091.000	0.528
4	4.000	35,091.000	0.528

	median_household_income
0	21667
1	21667
2	21667
3	21667
4	21667

PHASE I - Data cleaning

- **Dropped** the unnecessary features such as - 'collection_datetime', 'fn', 'address_full', 'incorporated_place', 'major_city', 'provider', 'speed_unit', 'income_lmi', 'income_dollars_below_median'.
- **Dropped** nan values for 'price', 'technology', 'package'.
- **Merged** similar package names in 'package' for simplicity.
- **Filled** the nan values using mode for 'redlining_grade' and 'n_providers'.
- **Filled** the nan values using median for 'ppl_per_sq_mile' and 'internet_perc_broadband'.

These are the number of missing observations in my dataset.

```
The no. of missing observations in the dataset before cleaning:
address_full                      0
incorporated_place                0
major_city                        0
state                             0
lat                               0
lon                               0
block_group                       0
collection_datetime               0
provider                          0
speed_down                        0
speed_up                          0
speed_unit                    82600
price                         82600
technology                    82600
package                       82600
fastest_speed_down                0
fastest_speed_price               0
fn                                0
redlining_grade              246027
race_perc_non_white               0
income_lmi                    17661
ppl_per_sq_mile                7965
n_providers                    7969
income_dollars_below_median   17661
internet_perc_broadband          849
median_household_income            0
dtype: int64
```

PHASE I - Cleaned dataset

Displaying the cleaned dataset:

	state	lat	lon	block_group	speed_down	speed_up	price	technology
0	AL	34.745	-86.607	10890007022	0.768	0.384	55.000	Not Fiber
1	AL	34.748	-86.607	10890007022	5.000	1.000	55.000	Not Fiber
2	AL	34.747	-86.606	10890007022	0.768	0.384	55.000	Not Fiber
3	AL	34.748	-86.608	10890007022	5.000	1.000	55.000	Not Fiber
4	AL	34.749	-86.605	10890007022	300.000	300.000	55.000	Fiber

	package	fastest_speed_down	fastest_speed_price	redlining_grade
0	Basic Internet	0.768	55.000	C
1	Basic Internet	5.000	55.000	C
2	Basic Internet	0.768	55.000	C
3	Basic Internet	5.000	55.000	C
4	Fiber Internet	5,000.000	180.000	C

	race_perc_non_white	ppl_per_sq_mile	n_providers	internet_perc_broadband
0	0.475	512.090	4.000	0.528
1	0.475	512.090	4.000	0.528
2	0.475	512.090	4.000	0.528
3	0.475	512.090	4.000	0.528
4	0.475	512.090	4.000	0.528

	median_household_income
0	21667
1	21667
2	21667
3	21667
4	21667

The number of missing observations in the dataset after cleaning:

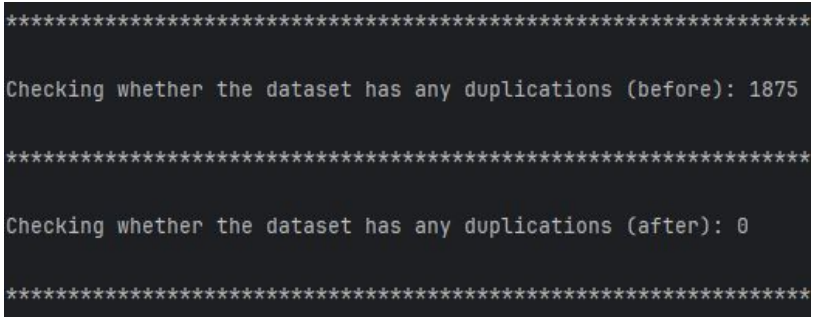
state	0
lat	0
lon	0
block_group	0
speed_down	0
speed_up	0
price	0
technology	0
package	0
fastest_speed_down	0
fastest_speed_price	0
redlining_grade	0
race_perc_non_white	0
ppl_per_sq_mile	0
n_providers	0
internet_perc_broadband	0
median_household_income	0

dtype: int64

PHASE I - Checking duplicates

There were 1875 duplicates in the dataset.

I dropped all the duplicates from the existing dataset.



PHASE I - Aggregation

Grouping by 'block_group'

- The subset of dataset is grouped by 'block_group' – it represents a census block group based on the latitude and longitude.

Aggregation

Using mean:

- 'price', 'speed_down', 'speed_up', 'ppl_per_sq_mile', 'internet_perc_broadband', 'lat', 'lon', 'race_perc_non_white' and 'median_household_income'.

Using mode:

- 'package'.

Using sum:

- 'n_providers'.

I am going to use this aggregated_att in phase 2 and 3:

	block_group	price	speed_down	speed_up	n_providers
0	10890002021	55.000	289.464	289.321	112.000
1	10890002022	55.000	300.000	300.000	145.000
2	10890003011	54.554	294.821	294.661	224.000
3	10890003012	55.000	283.838	283.838	148.000
4	10890003013	55.000	288.031	288.015	125.000
...	...	...	...	...	...
11455	550791870001	55.000	195.200	183.200	30.000
11456	550791870002	55.000	300.000	300.000	4.000
11457	550791870003	55.000	109.286	73.286	42.000
11458	550791874001	55.000	54.077	36.923	52.000
11459	550799800001	55.000	23.000	3.714	14.000

	ppl_per_sq_mile	internet_perc_broadband	lat	lon
0	696.213	0.586	34.769	-86.580
1	619.672	0.618	34.766	-86.581
2	990.162	0.808	34.781	-86.588
3	907.839	0.799	34.788	-86.586
4	1,495.346	0.752	34.790	-86.595
...	...	...	...	...
11455	7,232.242	0.906	43.058	-87.883
11456	12,189.077	0.859	43.053	-87.892
11457	15,357.599	0.969	43.055	-87.889
11458	2,230.873	0.936	43.033	-87.908
11459	0.000	0.825	43.050	-87.887

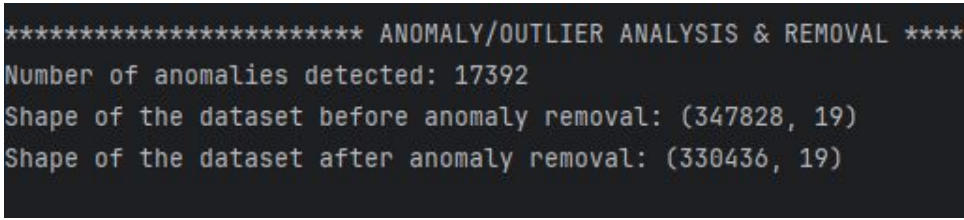
	race_perc_non_white	median_household_income	package
0	0.890	15,992.000	Fiber Internet
1	0.740	26,094.000	Fiber Internet
2	0.719	36,964.000	Fiber Internet
3	0.888	24,850.000	Fiber Internet
4	0.955	34,167.000	Fiber Internet
...	...	...	...
11455	0.088	85,476.000	Fiber Internet
11456	0.188	41,060.000	Fiber Internet
11457	0.294	62,368.000	High-Speed Internet
11458	0.158	88,144.000	Basic Internet

PHASE I - Anomaly Detection and Removal

Applied K-Means clustering with 3 clusters (distance-based anomaly detection) to identify outliers in the dataset.

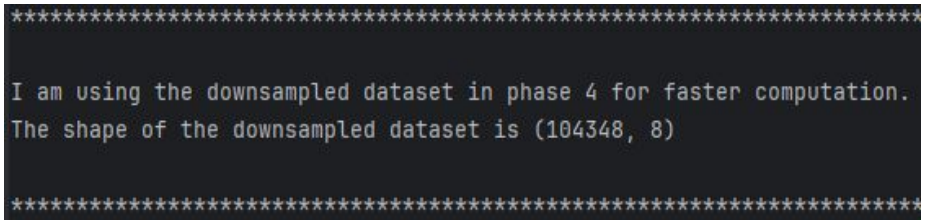
Anomalies were detected based on their distances from the cluster centers and removed to ensure cleaner data and more accurate model performance.

Number of anomalies detected are 17,392.



PHASE I - Downsampling

Downsampling my dataset to 30% for faster computation in phase 4 for clustering.



PHASE I - One-Hot Encoding

Performing One-Hot Encoding avoiding the dummy trap by using get_dummies().

Implementing it on the categorical features -
'state', 'package', 'technology', 'redlining_grade'.

```
Displaying the encoded dataset-
      lat     lon  block_group  speed_down  speed_up  price  \
0 34.745 -86.607  10890007022       0.768     0.384 55.000
1 34.748 -86.607  10890007022       5.000     1.000 55.000
2 34.747 -86.606  10890007022       0.768     0.384 55.000
3 34.748 -86.608  10890007022       5.000     1.000 55.000
4 34.749 -86.605  10890007022     300.000   300.000 55.000

   fastest_speed_down  fastest_speed_price  race_perc_non_white  \
0               0.768               55.000                0.475
1               5.000               55.000                0.475
2               0.768               55.000                0.475
3               5.000               55.000                0.475
4           5,000.000              180.000                0.475

   ppl_per_sq_mile  n_providers  internet_perc_broadband  \
0          512.090        4.000                    0.528
1          512.090        4.000                    0.528
2          512.090        4.000                    0.528
3          512.090        4.000                    0.528
4          512.090        4.000                    0.528

   median_household_income  anomaly  state_AR  state_CA  state_FL  state_GA  \
0                    21667        0     False     False     False     False
1                    21667        0     False     False     False     False
2                    21667        0     False     False     False     False
3                    21667        0     False     False     False     False
4                    21667        0     False     False     False     False

   state_IL  state_IN  state_KS  state_KY  state_LA  state_MI  state_MO  \
0     False     False     False     False     False     False     False
1     False     False     False     False     False     False     False
2     False     False     False     False     False     False     False
3     False     False     False     False     False     False     False
4     False     False     False     False     False     False     False

   state_IL  state_IN  state_KS  state_KY  state_LA  state_MI  state_MO  \
0     False     False     False     False     False     False     False
1     False     False     False     False     False     False     False
2     False     False     False     False     False     False     False
3     False     False     False     False     False     False     False
4     False     False     False     False     False     False     False

   state_MS  state_NC  state_OH  state_OK  state_SC  state_TN  state_TX  \
0     False     False     False     False     False     False     False
1     False     False     False     False     False     False     False
2     False     False     False     False     False     False     False
3     False     False     False     False     False     False     False
4     False     False     False     False     False     False     False

   state_WI  package_Fiber Internet  package_High-Speed Internet  \
0     False                   False                        False
1     False                   False                        False
2     False                   False                        False
3     False                   False                        False
4     False                    True                        False

   technology_Not Fiber  redlining_grade_B  redlining_grade_C  \
0                  True              False               True
1                  True              False               True
2                  True              False               True
3                  True              False               True
4                 False              False               True

   redlining_grade_D
0              False
1              False
2              False
3              False
4              False
```

PHASE I - Splitting the dataset

1. Target Variable:
    - Selected ‘speed_up’ (upload speeds) as the target variable for regression.
2. Feature Matrix and Target Variable:
    - X: Features (all columns except ‘speed_up’).
    - y: Target variable (only ‘speed_up’).
3. Train-Test Split:
    - Split the dataset into training (80%) and testing (20%) subsets.
    - Performed with shuffle=True to ensure randomness in the split.

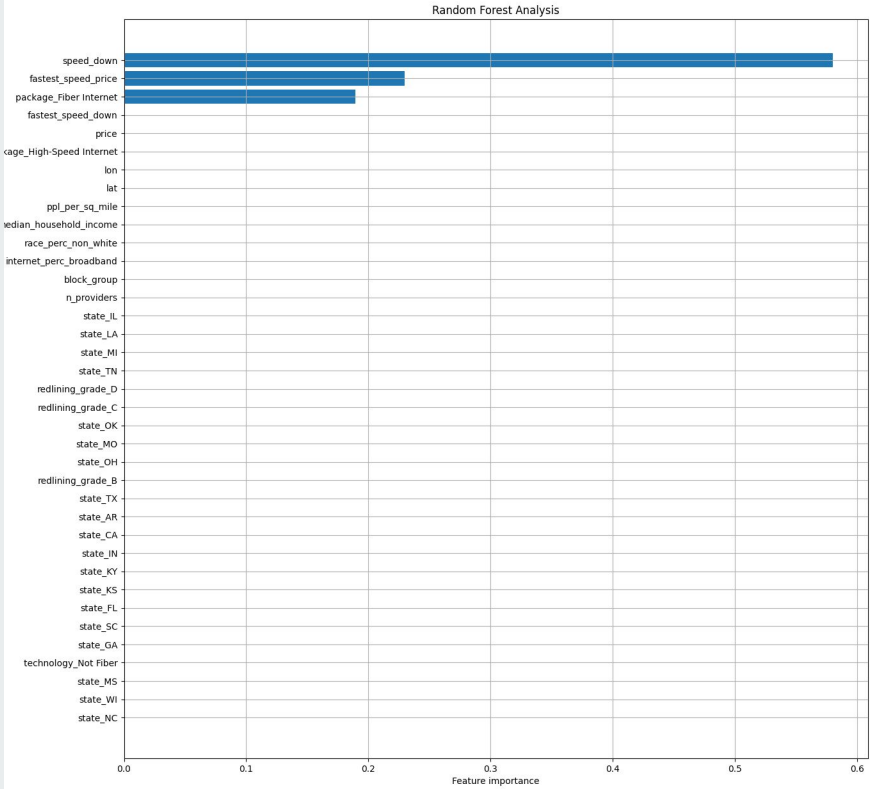
PHASE I:
FEATURE SELECTION & DIMENSIONALITY REDUCTION ANALYSIS

PHASE I - Random Forest

Random Forest helps to determine the most important features according to the feature importances.

This graph highlights the relative importance of all the features for the target.

It is evident that ‘speed_down’, ‘fastest_speed_price’ and 'package_Fiber Internet' have shown the highest importance in predicting the target.



PHASE I - Random Forest

Using a threshold of 0.01 for segregating the selected features to retain from the eliminated features according to their importances to the target variable.

Selected Features are ‘speed_down’, ‘fastest_speed_price’ and 'package_Fiber Internet'.

Other features are eliminated such as ‘fastest_speed_down’, 'package_AT&T FIBER—INTERNET 2000', 'price', etc.

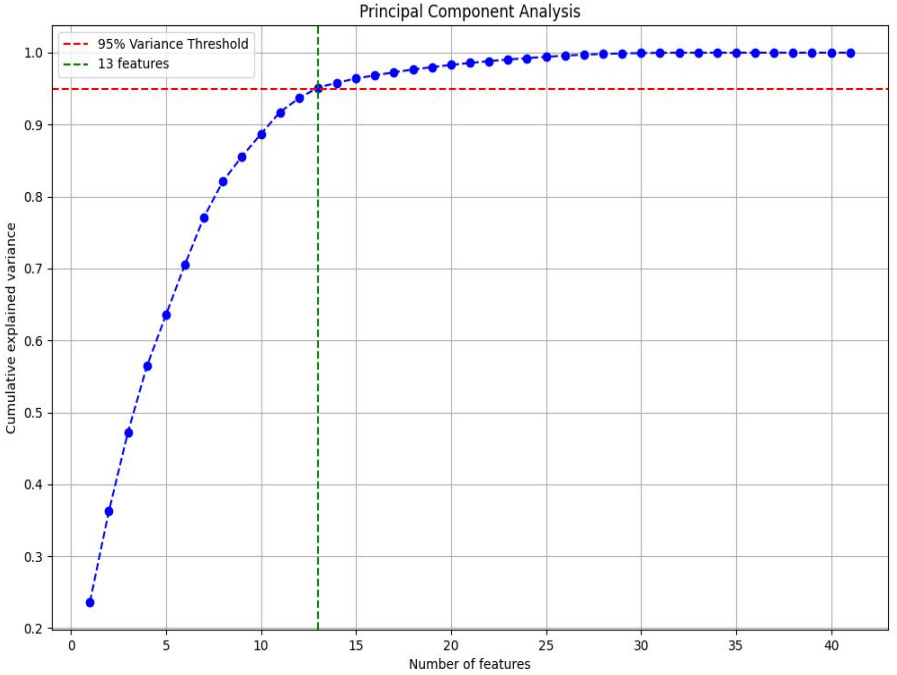
```
Selected Features (Random Forest): Index(['speed_down', 'fastest_speed_price', 'package_Fiber Internet'], dtype='object')

Eliminated Features (Random Forest): Index(['fastest_speed_down', 'price', 'package_High-Speed Internet', 'lon',
       'lat', 'ppl_per_sq_mile', 'median_household_income',
       'race_perc_non_white', 'internet_perc_broadband', 'block_group',
       'n_providers', 'state_IL', 'state_LA', 'state_MI', 'state_TN',
       'redlining_grade_D', 'redlining_grade_C', 'state_OK', 'state_MO',
       'state_OH', 'redlining_grade_B', 'state_TX', 'state_AR', 'state_CA',
       'state_IN', 'state_KY', 'state_KS', 'state_FL', 'state_SC', 'state_GA',
       'technology_Not Fiber', 'state_MS', 'state_WI', 'state_NC'],
      dtype='object')
```

PHASE I - Principal Component Analysis

PCA is applied to reduce the dataset's dimensionality while preserving maximum variance.

PCA needs standardized dataset to ensure all features contribute equally to the transformation.

After implementing PCA, the number of features retained is 13.




```
********** PRINCIPLE COMPONENT ANALYSIS **********

Number of features needed to explain more than 95% of the variance: 13
```

PHASE I - Singular Value Decomposition

SVD is very similar to PCA, it reduces the dimensionality of the dataset while retaining the most important variance.

It also requires standardized dataset to ensure all features contribute equally to the transformation.

It allows you to select the number of top components you want to retain.

The results have top 12 components, meaning the dataset was reduced to 12 components retaining the key information of the dataset.

```
********************** SINGULAR VALUE DECOMPOSITION **********************

Selected Features:Index(['n_providers', 'ppl_per_sq_mile', 'block_group', 'lon',
       'internet_perc_broadband', 'race_perc_non_white', 'lat',
       'fastest_speed_down', 'fastest_speed_price', 'median_household_income',
       'speed_down', 'price'],
      dtype='object')
```

PHASE I - Variance Inflation Factor

VIF is used to detect multicollinearity in the dataset, to verify if the variables are highly correlated.

The features with very high VIF values (generally above 5), indicate high correlation and is recommended for removal to avoid redundancy.

In my case, ‘fastest_speed_down’ and ‘fastest_speed_price’ have very high VIF values indicating multicollinearity.

```
********** VARIANCE INFLATION FACTOR **********

Variance Inflation Factor Table:
                  Variable       VIF
0                    const     0.000
1                      lat     1.329
2                      lon     1.468
3             block_group     1.195
4              speed_down     2.338
5                   price     1.001
6       fastest_speed_down 4,774.451
7      fastest_speed_price 4,811.750
8      race_perc_non_white     1.516
9          ppl_per_sq_mile     1.280
10           n_providers     1.197
11  internet_perc_broadband     1.478
12  median_household_income     1.019

Here the the vif scores for fastest_speed_down and fastest_speed_price are very high.
```

PHASE I - Conclusion on feature selection

After performing various feature selection methods, such as Random Forest, PCA, SVD, and VIF, the most effective approach for each phase is VIF for all the phases.

- **Phase II:** For Regression, VIF performs exceptionally well as it identifies and eliminates the multicollinear features, resulting in a more stable model.
- **Phase III:** For Classification, VIF worked well across all classifiers which helps to improve the performances of each model.
- **Phase IV:** For Clustering, VIF played a crucial role in ensuring that the features used for clustering were independent, enhancing the quality of clusters formed.

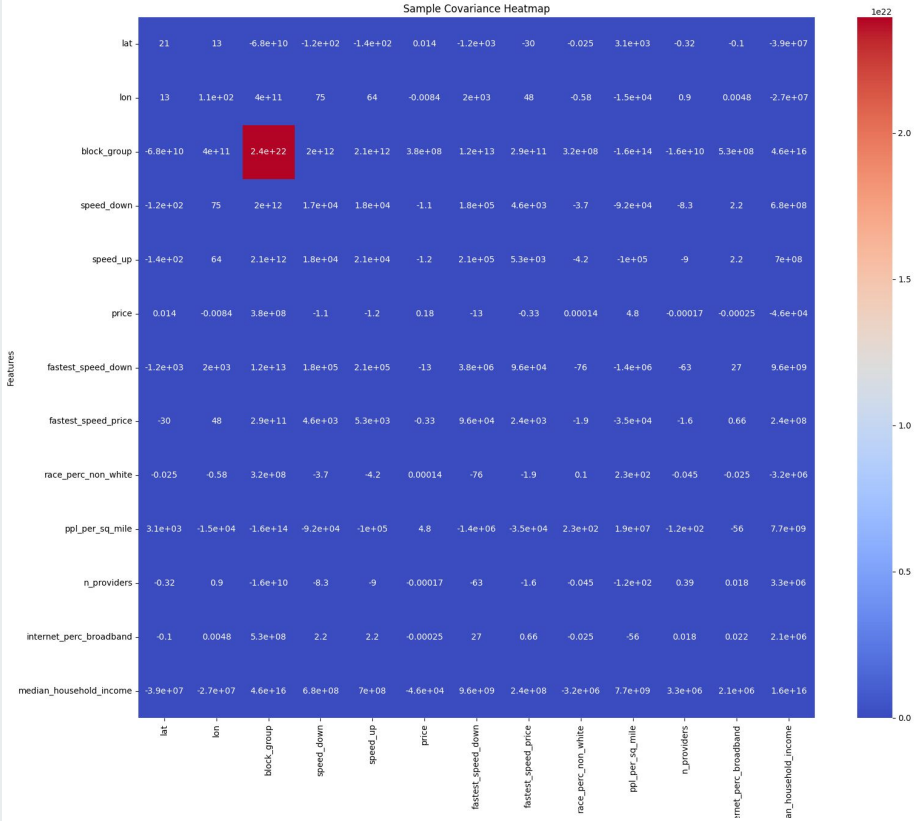
Overall, VIF seems the most reliable feature selection method for my dataset across all phases, providing consistency and effectiveness in improving model accuracy and reducing complexity.

PHASE I - Sample Covariance Matrix

Covariance matrix is to understand the relationship between numerical features in the dataset.

It ranges from 0 to 1.

Heatmap visualization shows the covariance values between the features, with most values close to zero, indicating weak correlations.



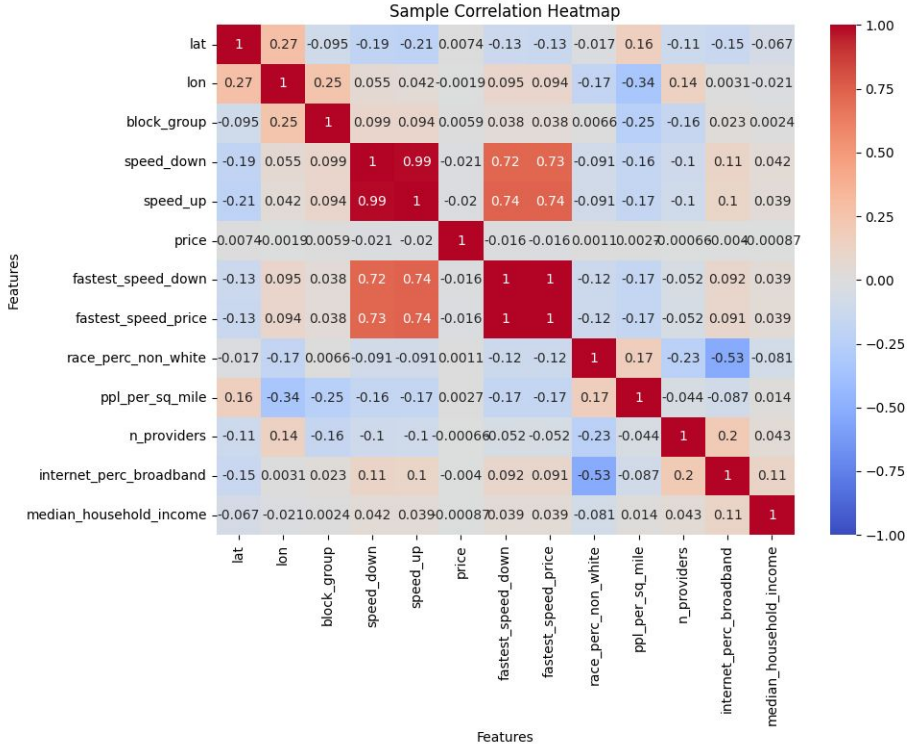
PHASE I - Sample Correlation Coefficient Matrix

Pearson's sample correlation coefficient matrix is used to understand the linear relationships between numerical features in the dataset.

The correlation between features is mostly weak, with values closer to 0, indicating low linear relationships between most variables.

The heatmap visualizes the coefficients ranging from -1 (negative correlation) to 1 (positive correlation).

Some features show moderate correlations, while the rest have weak or no correlation.



PHASE I - Checking if target is balanced or not

My target variable is 'package'.

Value counts of 'package':

- Fiber Internet: 147,703 instances.
- High-Speed Internet: 104,093 instances.
- Basic Internet: 78,640 instances.

Since, all the classes in the target variable have relatively comparable counts, the target variable is balanced.

```
Internet Package class value counts for comparision:
package
Fiber Internet         147703
High-Speed Internet    104093
Basic Internet          78640
Name: count, dtype: int64
```

PHASE II:
REGRESSION ANALYSIS

Data preparation for PHASE II

Feature selection - based on VIF to eliminate multicollinearity and focus on the most important variables.

Data cleaning - dropped unnecessary columns and handled the missing values by mode and median.

Data aggregation - grouped by block_group with all the VIF selected features.

Outlier removal - detected and removed all the outliers using K-means.

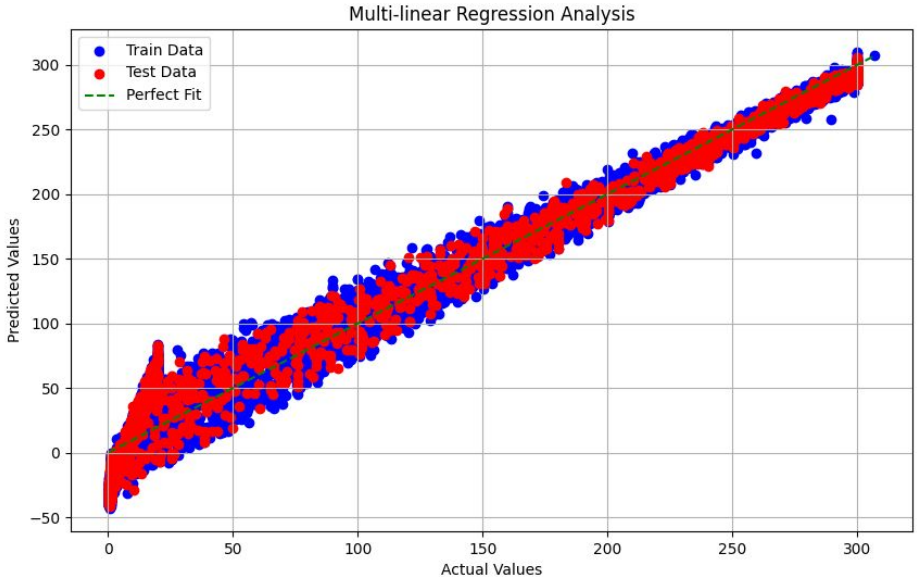
Target variable - ‘speed_up’

Splitting the dataset into 80% training and 20% testing for the regression model.

PHASE II - Multi-linear Regression

Applied regression to predict ‘speed_up’ (upload speeds) based on selected features.

This is the scatter plot which was created comparing the predicted values for the training and testing datasets which helps to visualize the model’s performance better.



PHASE II - Multi-linear Regression

These are the performance metrics of the model.

The model demonstrates a strong fit to the data based on the high R-squared and Adjusted R-squared scores,.

AIC and BIC scores measure the model's complexity. High AIC and BIC suggests that model has complexity.

MSE of around 300 indicates good predictive accuracy.

Multiple Linear Regression Metrics table:

Metric	Value
R-squared	0.977
Adjusted R-squared	0.977
AIC	74,558.373
BIC	74,636.166
MSE	300.839

PHASE II - T-test Analysis

The T-test evaluates the significance of each predictor variable in the regression model.

Features with higher values are more significant contributors to the model.

According to the values, it is evident that ‘speed_down’ has the most influence on the target variable ‘speed_up’.

```
T-test results:
const                          0.733
block_group                   -1.365
price                         -0.697
speed_down                   571.482
n_providers                   -6.147
ppl_per_sq_mile               -4.164
internet_perc_broadband      -12.170
lat                          -14.312
lon                           -5.076
race_perc_non_white          -11.132
median_household_income       -2.913
dtype: float64
```

PHASE II - F-test Analysis & Confidence Interval Analysis

F-test evaluates the overall significance of the regression model whether the model as a whole provides a better fit to the data compared to a model with no predictors.

The F-test value of 36,887 which is significantly high indicates that the model is statistically significant and the predictors collectively explain a substantial portion of the variance.

Confidence Interval helps identify whether the coefficients are statistically significant.

It is evident that 'speed_down' has a strong positive relationship.

```
F-test result:
36886.81452587509

95% Confidence Intervals for Coefficients:
                                0       1
const                    -175.095 384.380
block_group                -0.000   0.000
price                      -6.898   3.278
speed_down                  1.085   1.093
n_providers                -0.022  -0.011
ppl_per_sq_mile            -0.000  -0.000
internet_perc_broadband   -20.474 -14.794
lat                        -0.855  -0.649
lon                        -0.153  -0.068
race_perc_non_white        -9.362  -6.558
median_household_income    -0.000  -0.000
```

PHASE II - Stepwise Regression

Stepwise regression is performed to iteratively remove predictors with high p-values, ensuring the final model includes only statistically significant features.

In the first iteration, ‘price’ was removed with a p-value of 0.48.

‘block_group’ was eliminated next with a p-value of 0.16.

The model will be refitted after removal of these features.

```
Eliminating price with a p-value of 0.4855
Eliminating block_group with a p-value of 0.1608
```

PHASE II - Stepwise Regression

The final model results –

R-squared and Adjusted R-squared is 97.7%, F-statistic 46,110 with a p-value of 0.00, indicating that the model as a whole is highly significant.

Stepwise regression ensures retaining only the most impactful predictors while maintaining high accuracy and interpretability.

Final Model:

OLS Regression Results

Dep. Variable:	speed_up	R-squared:	0.977
Model:	OLS	Adj. R-squared:	0.977
Method:	Least Squares	F-statistic:	4.611e+04
Date:	Sat, 07 Dec 2024	Prob (F-statistic):	0.00
Time:	21:25:00	Log-Likelihood:	-37269.
No. Observations:	8709	AIC:	7.456e+04
Df Residuals:	8700	BIC:	7.462e+04
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	2.3766	3.301	0.720	0.472	-4.094	8.847
speed_down	1.0893	0.002	572.871	0.000	1.086	1.093
n_providers	-0.0162	0.003	-6.049	0.000	-0.021	-0.011
ppl_per_sq_mile	-0.0002	3.81e-05	-4.022	0.000	-0.000	-7.86e-05
internet_perc_broadband	-17.7117	1.447	-12.237	0.000	-20.549	-14.874
lat	-0.7233	0.048	-15.054	0.000	-0.817	-0.629
lon	-0.1226	0.020	-6.236	0.000	-0.161	-0.084
race_perc_non_white	-7.9930	0.715	-11.186	0.000	-9.394	-6.592
median_household_income	-3.852e-09	1.32e-09	-2.917	0.004	-6.44e-09	-1.26e-09

Omnibus:	198.988	Durbin-Watson:	1.977
Prob(Omnibus):	0.000	Jarque-Bera (JB):	212.493
Skew:	-0.382	Prob(JB):	7.21e-47
Kurtosis:	3.030	Cond. No.	2.63e+09

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The condition number is large, 2.63e+09. This might indicate that there are strong multicollinearity or other numerical problems.

PHASE III:
CLASSIFICATION ANALYSIS

Data preparation for PHASE III

Feature selection - based on VIF to eliminate multicollinearity and focus on the most important variables.

Data cleaning - dropped unnecessary columns and handled the missing values by mode and median.

Data aggregation - grouped by 'block_group' with all the VIF selected features.

Outlier removal - detected and removed all the outliers using K-means.

Target variable - 'package' – with three internet plans.

Splitting the dataset into 80% training and 20% testing (with stratify=y) to maintain class distribution in the training and testing sets for classification models.

PHASE III - Pre-Pruned Decision Tree Classifier

Applied the Pre-Pruned Decision Tree Classifier to classify the target variable 'package' (internet plans).

Using grid search to tune the hyperparameters to get the most optimized and best model.

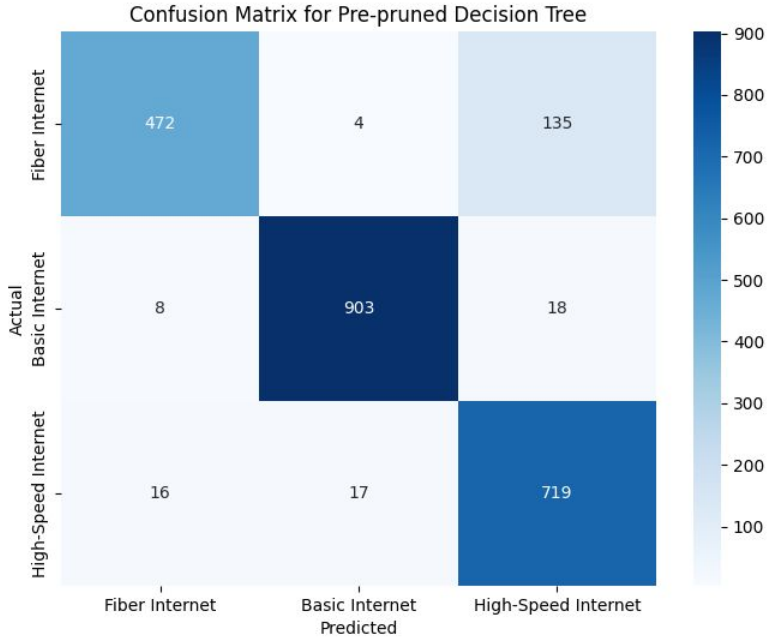
Generated the best parameters and the performance metrics for this classifier.

According to the metrics, it shows strong performance with a high AUC i.e., 0.98 and excellent metrics for both training and testing data.

```
Starting Grid Search for Pre-pruned Decision Tree...

Best Parameters: {'criterion': 'gini', 'max_depth': 5,
 'max_features': 'sqrt', 'min_samples_leaf': 30,
 'min_samples_split': 20, 'splitter': 'best'}

Confusion Matrix:
[[472   4 135]
 [  8 903  18]
 [ 16  17 719]]
Train Accuracy: 0.9142
Test Accuracy: 0.9136
Precision: 0.9203
Recall: 0.9136
Specificity: 0.9575
F1-score: 0.9129
AUC: 0.981674001126671
```

PHASE III - Pre-Pruned Decision Tree Classifier

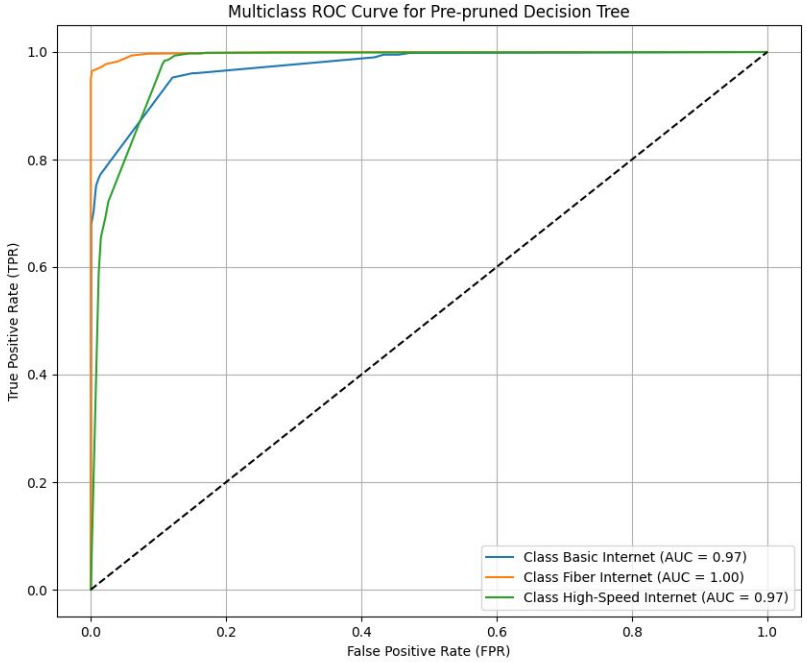
Confusion Matrix Heatmap represents the model's performance in predicting the target classes.



PHASE III - Pre-Pruned Decision Tree Classifier

AUC Scores:

- Basic Internet: 0.97
- Fiber Internet: 1.00
- High-Speed Internet: 0.97



PHASE III - Post-Pruned Decision Tree Classifier

Implemented Post-pruned decision tree classifier to predict the target the target variable.

Using grid Search to optimize the model's hyperparameters, to select the best ccp_alpha (cost complexity pruning alpha) which helps to prune the decision tree.

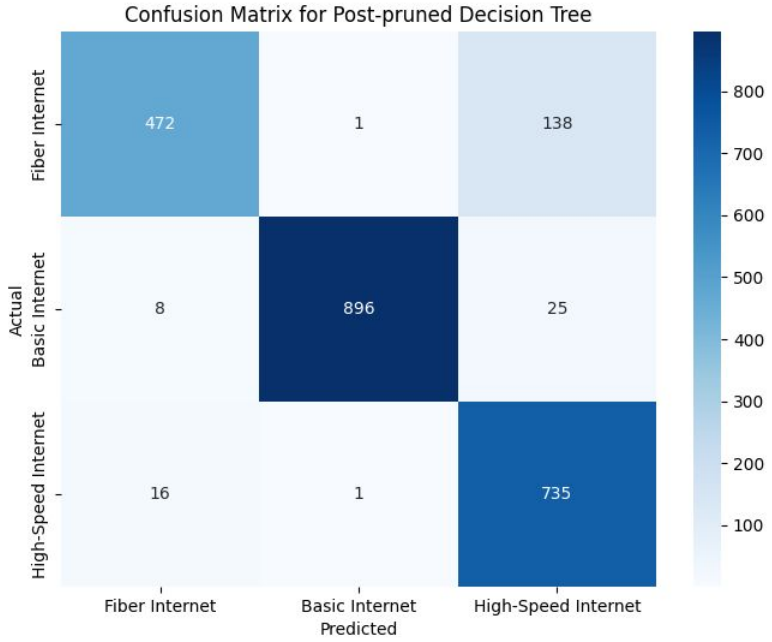
The best ccp_alpha value was chosen based on the cost-complexity pruning path to balance model performance and complexity.

The test accuracy is 91.75 indicating model's strong generalization ability.

```
Starting Grid Search for Post-pruned Decision Tree...

Best Parameters: {'criterion': 'gini', 'max_depth': 5,
 'max_features': 'sqrt', 'min_samples_leaf': 30,
 'min_samples_split': 20, 'splitter': 'best'}

Confusion Matrix:
[[472   1 138]
 [  8 896  25]
 [ 16   1 735]]
Train Accuracy: 0.9138
Test Accuracy: 0.9175
Precision: 0.9266
Recall: 0.9175
Specificity: 0.9591
F1-score: 0.9172
AUC: 0.9804987714755086
```

PHASE III - Post-Pruned Decision Tree Classifier

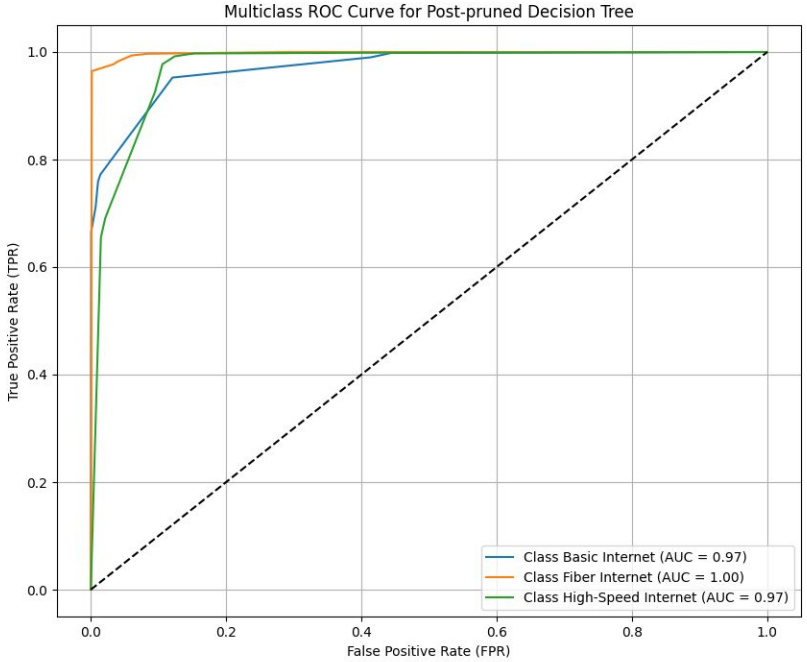
Confusion matrix shows the classifier's prediction for each class.



PHASE III - Post-Pruned Decision Tree Classifier

AUC Scores:

- Basic Internet: 0.97
- Fiber Internet: 1.00
- High-Speed Internet: 0.97



PHASE III - Logistic Regression Classifier

Implemented Logistic regression classifier to predict the target variable.

Using grid search to optimize the hyperparameters, selecting the best values to improve model performance.

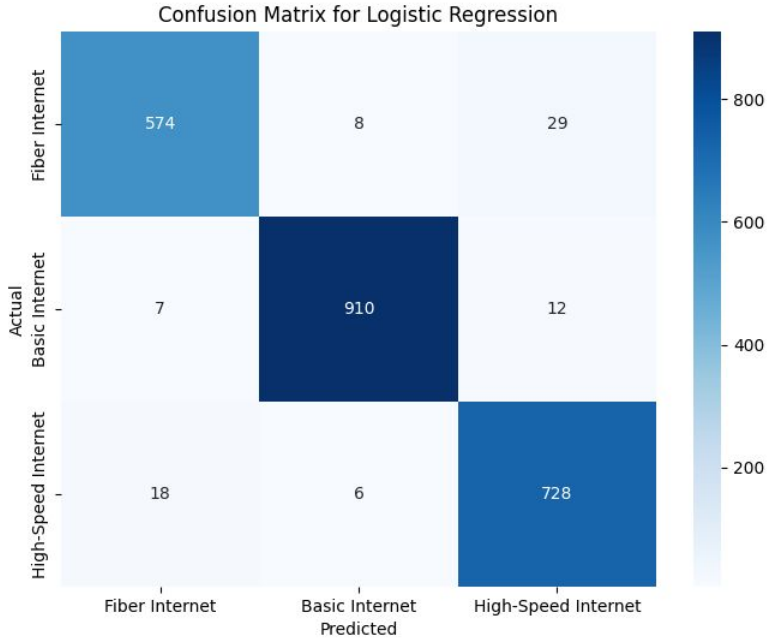
The test accuracy of 96.51% indicates the model's excellent generalization to new data.

The AUC score of 0.9967 shows exceptional performance.

```
Starting Grid Search for Logistic Regression...

Best Parameters: {'C': 10, 'penalty': 'l2'}

Confusion Matrix:
[[574   8  29]
 [  7 910  12]
 [ 18   6 728]]
Train Accuracy: 0.9596
Test Accuracy: 0.9651
Precision: 0.9652
Recall: 0.9651
Specificity: 0.9825
F1-score: 0.9651
AUC: 0.9967063042757225
```

PHASE III - Logistic Regression Classifier

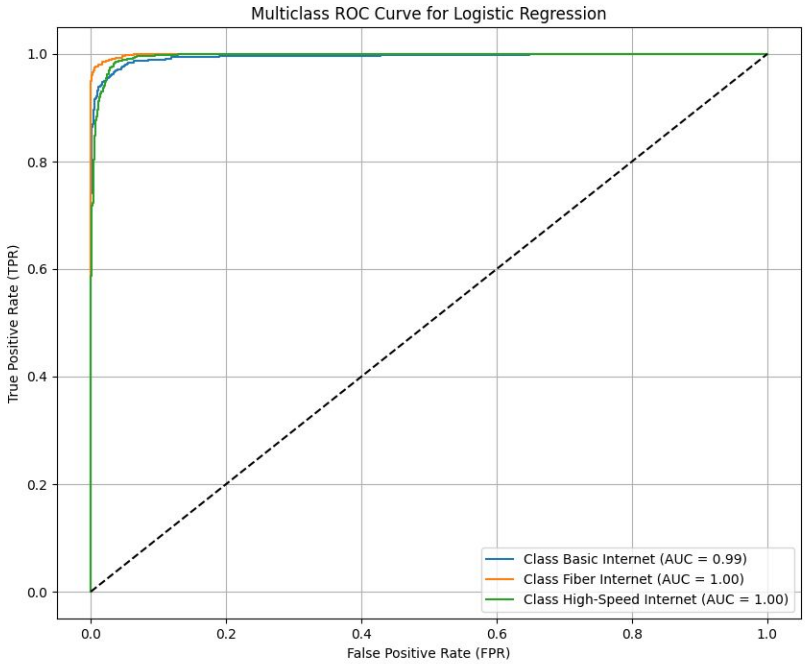
Confusion matrix shows the classifier's predictions for each class.



PHASE III - Logistic Regression Classifier

AUC Scores:

- Basic Internet: 0.99
- Fiber Internet: 1.00
- High-Speed Internet: 1.00



PHASE III - K-Nearest Neighbors Classifier

Applied the KNN Classifier to predict the target variable.

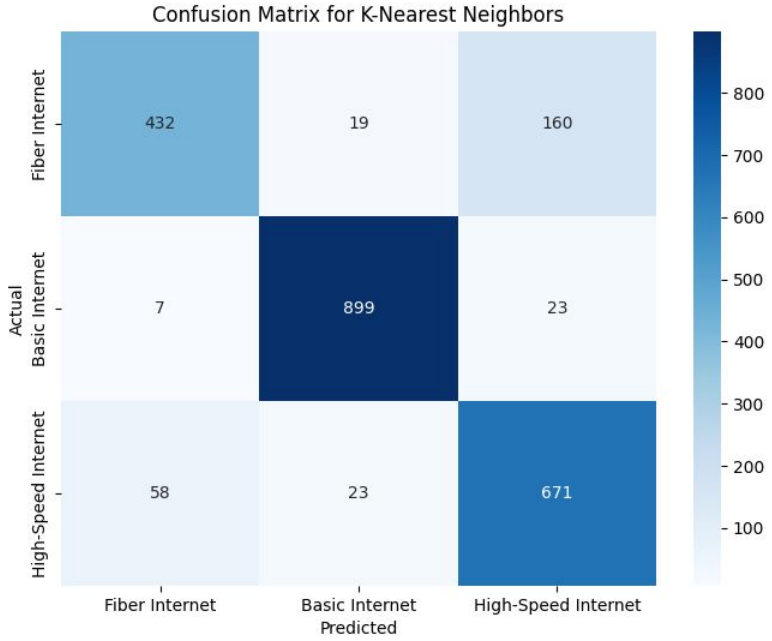
Using the grid search to find the best hyperparameters for the classifier.

The model achieves perfect accuracy on training data (1.00), but shows a test accuracy of 87.35% indicating some overfitting.

```
Starting Grid Search for K-Nearest Neighbors...

Best Parameters: {'algorithm': 'auto', 'n_neighbors': 11,
 'weights': 'distance'}

Confusion Matrix:
[[432  19 160]
 [  7 899  23]
 [ 58  23 671]]
Train Accuracy: 1.0000
Test Accuracy: 0.8735
Precision: 0.8767
Recall: 0.8735
Specificity: 0.9371
F1-score: 0.8718
AUC: 0.9676117729536106
```

PHASE III - K-Nearest Neighbors Classifier

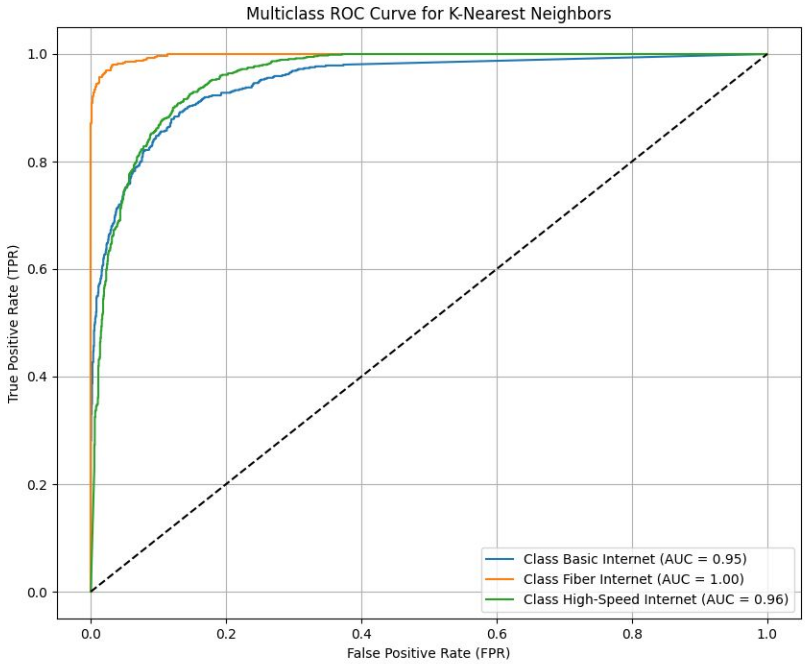
Confusion matrix displays the classifier's predictions for each class.



PHASE III - K-Nearest Neighbors Classifier

AUC Scores:

- Basic Internet: 0.95
- Fiber Internet: 1.00
- High-Speed Internet: 0.96

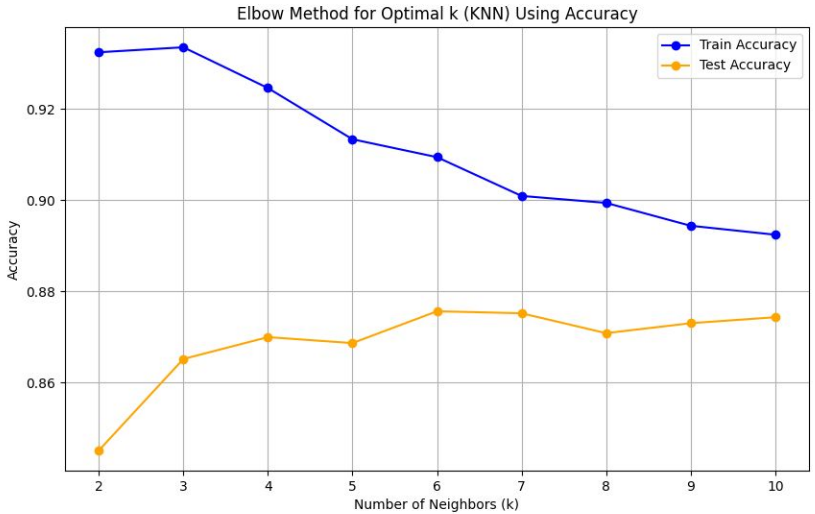


PHASE III - K-Nearest Neighbors Classifier

Used the Elbow Method to find the optimal number of neighbors (K) for the KNN classifier.

After plotting the WCSS (Within-Cluster Sum of Squares), the optimal value of K = 6 , as it provides the best balance between model complexity and performance.

This value was found where the elbow in the curve is observed, indicating minimal improvement with higher values of K.




```
Optimal k for KNN (using Elbow Method with Accuracy): 6
```

PHASE III - Support Vector Machine

Implemented the Support Vector Machine (SVM) Classifier to predict the target variable.

Using grid search to select the best kernel (linear, radial basis function, polynomial) for SVM.

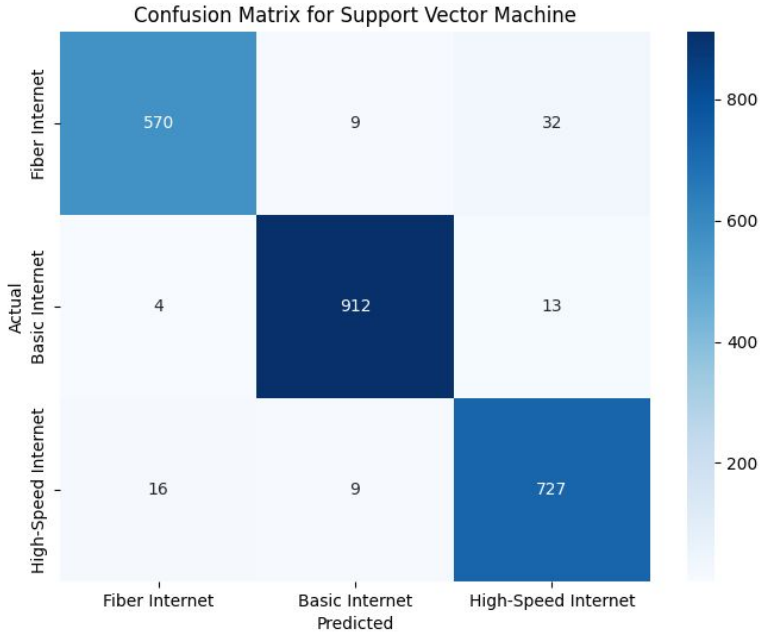
After Grid search, linear kernel was selected for its simplicity and effectiveness in separating the classes in a hyperplane.

Test accuracy of 96.38% indicates excellent model performance.

```
Starting Grid Search for Support Vector Machine...

Best Parameters: {'kernel': 'linear'}

Confusion Matrix:
[[570   9  32]
 [  4 912  13]
 [ 16   9 727]]
Train Accuracy: 0.9568
Test Accuracy: 0.9638
Precision: 0.9640
Recall: 0.9638
Specificity: 0.9821
F1-score: 0.9638
AUC: 0.9968979470532524
```

PHASE III - Support Vector Machine

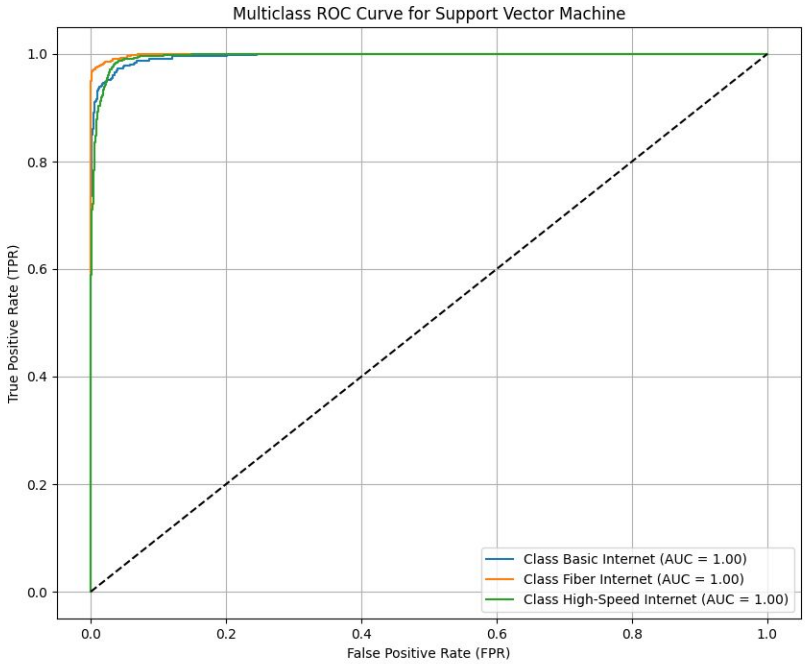
Confusion matrix shows the classifier's predictions for each class.



PHASE III - Support Vector Machine

AUC Scores:

- Basic Internet: 1.00
- Fiber Internet: 1.00
- High-Speed Internet: 1.00



PHASE III - Naives Bayes Classifier

Applied the Naives Bayes Classifier to predict the target.

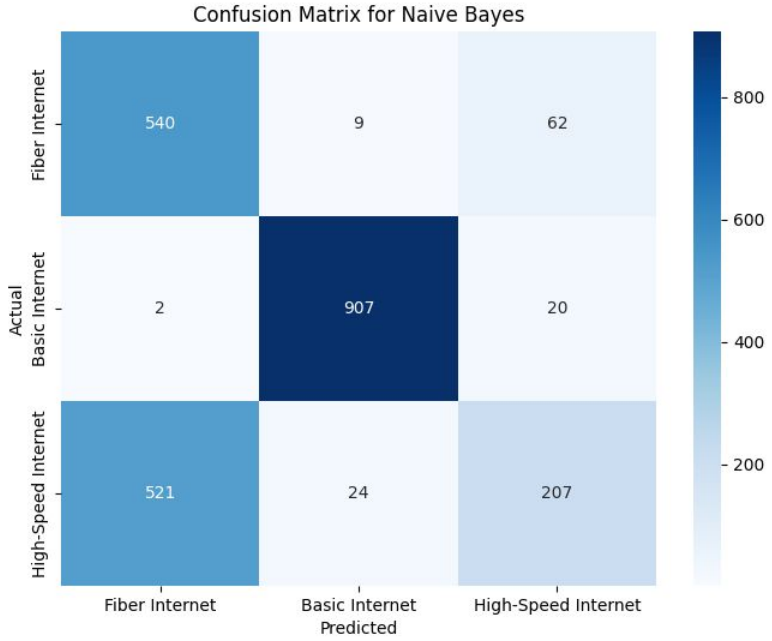
Searching the grid to select the optimal smoothing parameter, which helps improve the classifier's performance, especially in cases of small probabilities.

The model shows relatively low train and test accuracy that is around 72%, indicating potential underfitting.

```
Starting Grid Search for Naive Bayes...

Best Parameters: {'var_smoothing': 1e-07}

Confusion Matrix:
[[540   9  62]
 [  2 907  20]
 [521  24 207]]
Train Accuracy: 0.7232
Test Accuracy: 0.7216
Precision: 0.7615
Recall: 0.7216
Specificity: 0.8500
F1-score: 0.6959
AUC: 0.93928264565621 88
```

PHASE III - Naives Bayes Classifier

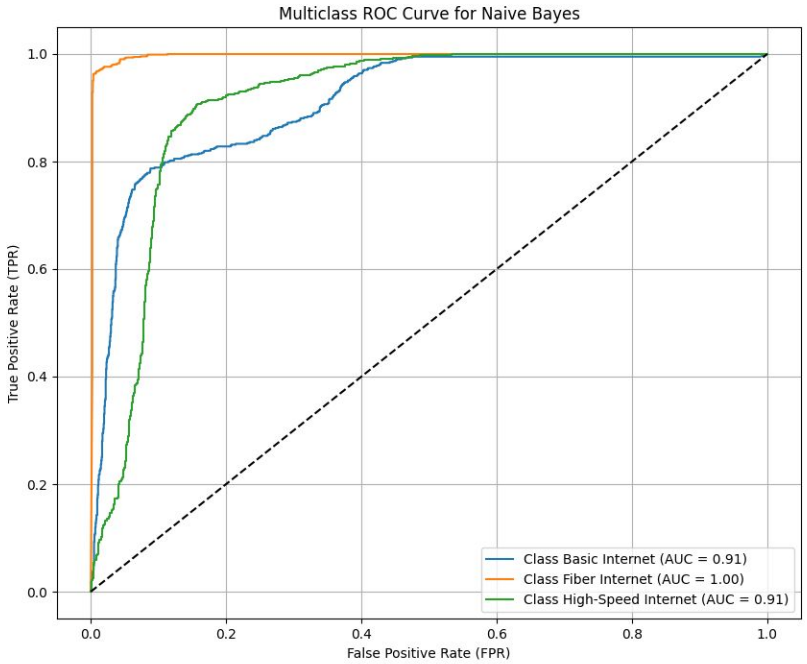
Confusion matrix displays the classifier's prediction for each class.



PHASE III - Naives Bayes Classifier

AUC Scores:

- Basic Internet: 0.91
- Fiber Internet: 1.00
- High-Speed Internet: 0.91



PHASE III - Neural Networks Classifier

Implemented the Neural Networks Classifier to predict the target variable.

Using grid search to find the optimal hyperparameters.

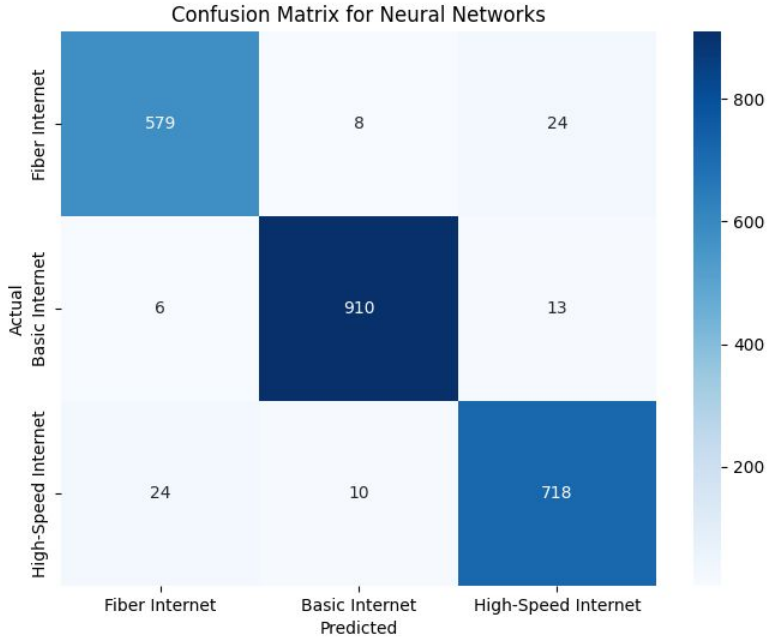
Test accuracy of 96.29% shows excellent model generalization.

The AUC score of 0.9967 indicates outstanding performance.

```
Best Parameters: {'activation': 'relu', 'alpha': 0.0001,
 'hidden_layer_sizes': (100,), 'learning_rate': 'constant'}

Confusion Matrix:
[[579   8  24]
 [  6 910  13]
 [ 24  10 718]]
Train Accuracy: 0.9691
Test Accuracy: 0.9629
Precision: 0.9629
Recall: 0.9629
Specificity: 0.9814
F1-score: 0.9629
AUC: 0.9967151746982684
                          Train Accuracy  Test Accuracy
```

PHASE III - Neural Networks Classifier

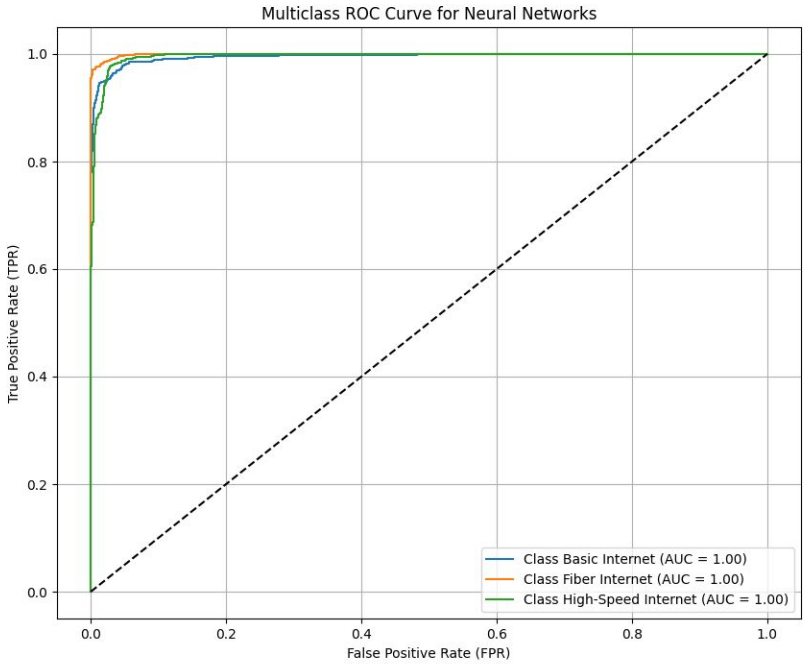
Confusion matrix shows the classifier's predictions for each class.



PHASE III - Neural Networks Classifier

AUC Scores:

- Basic Internet: 1.00
- Fiber Internet: 1.00
- High-Speed Internet: 1.00



PHASE III - Comparison of all the classifier models

Pre-pruned DT: Has good accuracy and strong AUC, but prone to overfit, especially for the Basic Internet class.

Post-pruned DT: The test accuracy improved but prone to underfit if pruning is too aggressive.

Logistic Regression: High test accuracy and AUC with strong generalization.

KNN: Perfect train accuracy, but lower test accuracy, indicating overfitting.

SVM: Strong test accuracy and AUC across all classes, but computationally expensive.

Naives Bayes: Struggled with lower accuracy and had lower AUC for certain classes.

Neural Networks: Showed perfect AUC for all classes and high test accuracy, but computationally expensive.

Model	Train Accuracy	Test Accuracy	Precision	Recall
Pre-pruned Decision Tree	0.914	0.914	0.920	0.914
Post-pruned Decision Tree	0.914	0.918	0.927	0.918
Logistic Regression	0.960	0.965	0.965	0.965
K-Nearest Neighbors	1.000	0.873	0.877	0.873
Support Vector Machine	0.957	0.964	0.964	0.964
Naive Bayes	0.723	0.722	0.762	0.722
Neural Networks	0.969	0.963	0.963	0.963

Model	Specificity (Weighted)	F1-score	AUC
Pre-pruned Decision Tree	0.958	0.913	0.982
Post-pruned Decision Tree	0.959	0.917	0.980
Logistic Regression	0.982	0.965	0.997
K-Nearest Neighbors	0.937	0.872	0.968
Support Vector Machine	0.982	0.964	0.997
Naive Bayes	0.850	0.696	0.939
Neural Networks	0.981	0.963	0.997

The **best classifier** in my opinion is **Logistic Regression** due to its strong performance with high test accuracy and AUC across all classes, as it has the ability to generalize well to new data.

PHASE IV: CLUSTERING & ASSOCIATION

Dataset for PHASE IV

Feature selection – Dropped categorical features and irrelevant numeric features for clustering based on VIF.

Data cleaning – Dropped unnecessary columns and handled missing values by filling with median or mean.

Feature engineering – Created a new feature, 'speed_category', by classifying 'speed_down' as 'Slow' or 'Fast'.

Downsampling – Reduced the dataset size to 30% for faster computation in clustering.

Target variable – No target variable used for clustering, focusing on unsupervised learning.

PHASE IV - K-Means Clustering

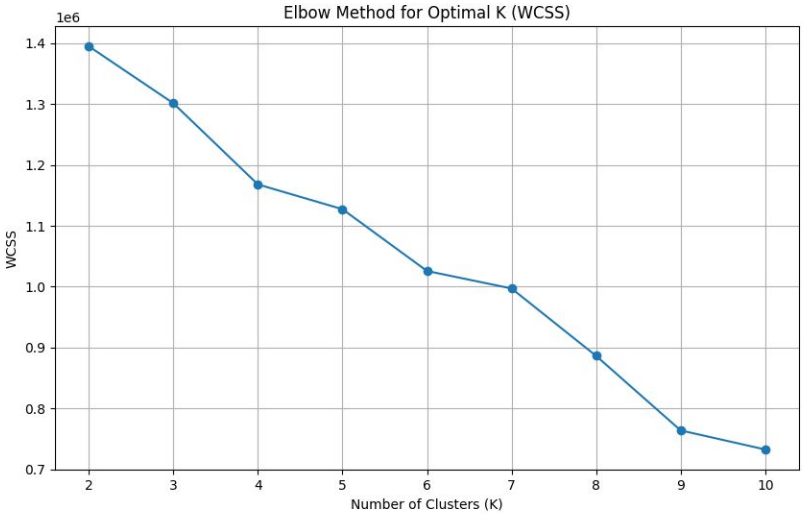
To find the optimal K there are two methods –

Elbow Method –

To determine the ideal number of clusters, we analyze how the Within-Cluster Sum of Squares (WCSS) changes with increasing K.

As K increases, WCSS decreases, but it slows down after a certain point and that point is known as elbow representing the Optimal K.

The elbow graph shows a distinct elbow at K=6 suggesting it as the optimal number of clusters.




```
Optimal K (using WCSS - Elbow Method): 6
```

PHASE IV - K-Means Clustering

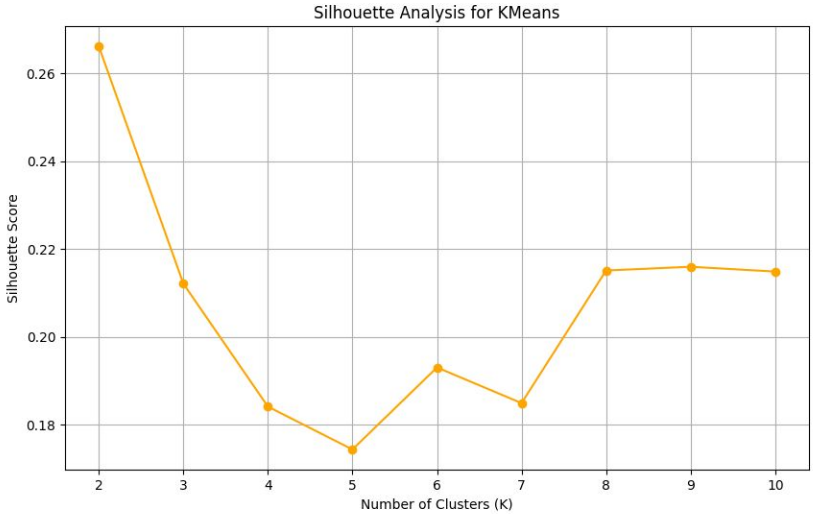
Silhouette Scores -

To determine the number of clusters, we measure the silhouette scores.

Higher scores indicate better-defined clusters, with a value close to 1.0 being ideal.

By analyzing scores for K 2 to 10, we identify K that yields the highest Silhouette Score as the optimal number of clusters.

The Silhouette graph indicates the highest score at K=2 suggesting it as the optimal number of clusters.



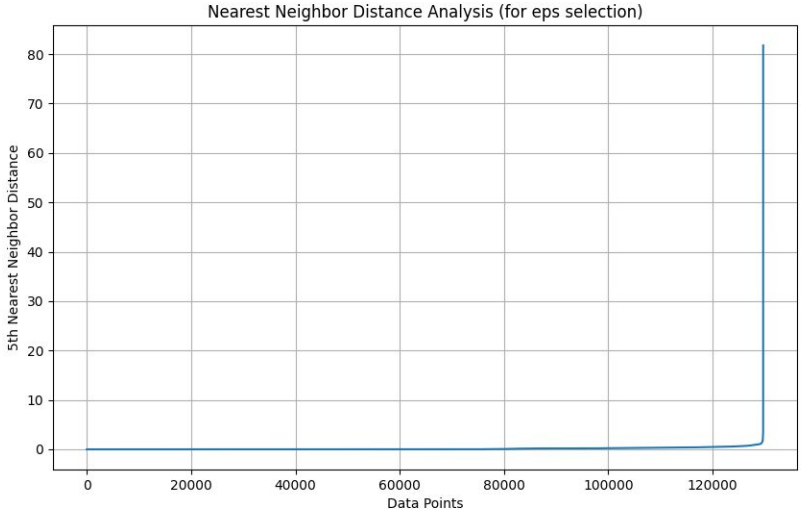

```
Optimal K (using Silhouette Score): 2
```

PHASE IV - DBSCAN Clustering

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is an unsupervised learning algorithm that groups data points into clusters based on density.

It relies on two parameters, eps (epsilon) and min_samples (the minimum number of points required to form a dense region).

After applying DBSCAN, it identified 95 clusters in the dataset and 398 noise points detected.




```
DBSCAN Number of clusters: 95
DBSCAN Number of noise points: 398
```

PHASE IV - Association Rule Mining

Association Rule Mining is an unsupervised learning technique used to discover relationships and co-occurrences between items in a dataset. It identifies frequent itemsets and generates rules that explain the likelihood of items appearing together.

- Fiber technology is highly predictive of Fiber Internet packages.
- Fast speed categories are frequently linked with Fiber technology and premium plans like Fiber Internet.

```
********** APRIORI ALGORITHM **********

Frequent Itemsets:
    support                                        itemsets
0  0.352970                    (package_Fiber Internet)
1  0.353602                         (technology_Fiber)
2  0.453725                     (technology_Not Fiber)
3  0.514754                       (speed_category_Slow)
4  0.485246                       (speed_category_Fast)
5  0.352954      (technology_Fiber, package_Fiber Internet)
6  0.352970   (package_Fiber Internet, speed_category_Fast)
7  0.353147        (technology_Fiber, speed_category_Fast)
8  0.321626     (speed_category_Slow, technology_Not Fiber)
9  0.352954  (technology_Fiber, package_Fiber Internet, spe...

Association Rules:
                                    antecedents  \
0                          (technology_Fiber)
1                     (package_Fiber Internet)
2                     (package_Fiber Internet)
3                        (speed_category_Fast)
4                          (technology_Fiber)
5                        (speed_category_Fast)
6                      (technology_Not Fiber)
7     (package_Fiber Internet, technology_Fiber)
8        (technology_Fiber, speed_category_Fast)
9  (package_Fiber Internet, speed_category_Fast)
10                         (technology_Fiber)
11                    (package_Fiber Internet)
12                       (speed_category_Fast)

                                    consequents   support  confidence  \
0                     (package_Fiber Internet)  0.352954    0.998168
1                          (technology_Fiber)  0.352954    0.999956
2                        (speed_category_Fast)  0.352970    1.000000
3                     (package_Fiber Internet)  0.352970    0.727404
4                        (speed_category_Fast)  0.353147    0.998713
5                          (technology_Fiber)  0.353147    0.727770
6                        (speed_category_Slow)  0.321626    0.708857
7                        (speed_category_Fast)  0.352954    1.000000
8                     (package_Fiber Internet)  0.352954    0.999454
9                          (technology_Fiber)  0.352954    0.999956
10  (package_Fiber Internet, speed_category_Fast)  0.352954    0.998168
11       (technology_Fiber, speed_category_Fast)  0.352954    0.999956
12    (package_Fiber Internet, technology_Fiber)  0.352954    0.727372

        lift
0   2.827915
1   2.827915
2   2.060812
3   2.060812
4   2.058160
5   2.058160
6   1.377079
7   2.060812
8   2.831558
9   2.827915
10  2.827915
11  2.831558
12  2.060812
```

Conclusion

Phase I: Conducted feature engineering and EDA, selecting optimal features using VIF to eliminate multicollinearity and prepare the dataset for robust modeling.

Phase II: Implemented regression analysis on upload speeds (speed_up) to predict and understand key factors influencing performance, achieving excellent results with Linear Regression.

Phase III: Built and analyzed seven classifiers for predicting internet packages, with Logistic Regression excelling in performance and efficiency, while Neural Networks captured complex patterns with perfect AUC scores.

Phase IV: Performed unsupervised clustering to identify density-based patterns using DBSCAN and K-Means, and uncovered actionable insights with association rule mining, revealing strong relationships among internet packages, technologies, and speeds.

THANK YOU!